

Contents

Chapter 1—Introduction	1-1
What is CircuitLogix?	1-1
Required Hardware/Software	1-3
First Time Installation	1-3
Multi-user (Project) Installations	1-3
Using On-line Help	1-5
Where to go from here	1-6
Technical Support	1-6
Required User Background	1-6
Chapter 2—Schematic Capture	2-1
Getting Started—Schematic Example	2-1
The Editing Tools	2-4
Placing Devices	2-6
Wiring the Circuit	2-9
Using Bus Wires	2-11
Labeling the Circuit	2-12
Netlist Information	2-12
Printing and Exporting Circuits	2-21
Creating User-defined Schematic Symbols	2-22
Chapter 3—Analog/Mixed-Mode Simulation	3-1
Using the Analog Simulator	3-2
Getting Started—Analog Circuit Examples	3-2
Simulation Tools	3-12
Vcc and Ground	3-15
Test Points	3-15
Run-Time Test Points	3-16
SPICE for Windows	3-17
The Graphics Windows	3-20
Analog Analyses	3-24
Operating Point Analysis	3-25
DC Analysis	3-25
AC Analysis	3-26
Transient Analysis	3-26
Multimeter	3-28
Multifunction Signal Generator	3-29
The Data Sequencer	3-35
Initial Conditions	3-35
SPICE's Elementary Devices	3-36

Contents

Chapter 4—Digital Logic Simulation	4-1
Using the Digital Logic Simulator	4-2
Getting Started—Digital Circuit Examples	4-2
Simulation Tools	4-3
Propagation Delays	4-6
The Pulser	4-7
Digital Waveforms	4-10
Digital Options	4-11
Digital Tutorial	4-12
What is Digital Electronics?	4-13
Frequently Used Terms	4-14
Experiment 1	4-17
Experiment 2	4-19
Experiment 3	4-21
Experiment 4	4-23
Experiment 5	4-26
Experiment 6	4-29
Conclusion	4-32
Answers to Experiments	4-33
Chapter 5 — File Menu	5-1
New	5-1
Open	5-1
Close	5-1
Revert	5-1
Merge	5-1
Save	5-2
Save As	5-2
Export Options	5-2
Export Circuit	5-3
Export Waveforms	5-3
Print Setup	5-3
Print Circuit	5-4
Print Waveforms	5-4
Script Functions	5-4
Create Parts list (.PRT)	5-5
Create SPICE netlist (.NET)	5-5
Create SPICE subcircuit (.SUB)	5-5
Create PCB netlist (.NET)	5-5
Simulate external SPICE netlist	5-6
Step device value and run SPICE	5-6
Preferences	5-7
Exit	5-10

Contents

Chapter 6 — Edit Menu	6-1
Undo	6-1
Cut	6-1
Copy	6-1
Paste	6-1
Move	6-1
Delete Items	6-1
Duplicate	6-2
Select All	6-2
Copy Circuit	6-2
Copy Waveforms	6-2
Refresh Screen	6-3
Rotate 90	6-3
Mirror	6-3
Straighten Selected Wire(s)	6-3
Place Selected Label(s)	6-3
Set Prop Delay(s) (for Digital Simulation Mode)	6-3
Set Auto Designation	6-4
Edit Selected Item	6-4
Group Selected Items	6-8
Select Font	6-8
Chapter 7 — Fault Simulation	7-1
Device Faults	7-1
Access Faults	7-4
Circuit Faults	7-5
“Black Box” Macros	7-9
FaultExample	7-9
Chapter 8 — Device Editor	8-1
The Macro Button	8-2
The Symbol Editor	8-3
Drawing with the Mouse	8-3
Adding an Existing Shape	8-4
Adding a DIP Package	8-5
Editing Pin Information	8-5
The Element List and the Edit Buffer	8-5
The Trace Feature	8-6
Element Definitions	8-6
When the Symbol is Complete	8-8
Creating a New Macro	8-8

Contents

Digital Simcode Devices	8-12
SPICE Models and Subcircuits	8-14
SPICE Models	8-18
SPICE Subcircuits	8-22
Parameter Passing	8-26

Chapter 9 — Device Editor Menu 9-1

Define New Macro	9-1
Edit Macro	9-1
Save Macro	9-2
Expand Macro	9-2
Macro Lock	9-3
Macro Utilities	9-3
Macro Copier	9-5
Save ASCII Library	9-6
Convert ASCII Library	9-6

Chapter 10 — Options Menu 10-1

Circuit Info	10-1
Repeat On	10-1
Auto Refresh	10-1
Auto Designation	10-1
Display Toolbar	10-2
Display Variable Names	10-2
Scope Probe	10-2
Arrow/Wire	10-2
Cursor Tools	10-2
Digital/Analog Mode	10-2
Reset Simulation	10-3
Step Simulation	10-3
Run Simulation	10-3
Trace On	10-3
Waveforms	10-3
Digital Options	10-3
Analog Options	10-3
Analog Analyses	10-3

Contents

Chapter 11 — View Menu	11-1
Select Colors	11-1
Display Scale	11-2
Normal Size/Position	11-2
Fit Circuit To Window	11-2
Find and Select	11-2
Show Pin Dots	11-3
Show Bus Labels	11-3
Show Page Breaks	11-3
Show Node Numbers	11-3
Show Prop Delays	11-3
Device Display Data	11-4
Circuit Display Data	11-4
Drawing Grid	11-4
Title Box	11-5
Cascade Windows	11-5
Tile Windows	11-5
Windows	11-5
Chapter 12 — Device Library	12-1
Appendix A: Troubleshooting SPICE Convergence	A-1
Solutions to Operating Point Analysis Failures	A-1
Solutions to DC Analysis Failures	A-2
Solutions to Transient Analysis Failures	A-2
Suggested Reading	A-3
Appendix B: Frequently Asked Questions	B-1

Contents

Chapter 1

Introduction

What is CircuitLogix?

CircuitLogix is an electronics lab in a computer, and is a powerful yet easy to use schematic capture and circuit simulation tool developed by Logic Design Inc. Its schematic capabilities will enable you to draw any electronic circuit and create a netlist for PCB design tools and autorouters. You can easily build an analog or digital circuit schematic, attach test instruments, and run the simulator to see how the circuit functions. The behaviour of a circuit is simulated realistically, and the results are quickly displayed. It can perform a live, free running simulation of the digital portion of your circuit. It can also perform analog and mixed analog/digital simulations based on Berkeley SPICE and XSPICE.

SPICE (Simulation Program with Integrated Circuit Emphasis), is the industry standard set of algorithms for analog simulation. CircuitLogix expands SPICE with its own proprietary mixed-signal extensions to support digital devices. As a result, you can combine digital and analog devices any way you want, with CircuitLogix automatically handling the signal processing. You can choose from ideal or real-world SPICE models, or create your own models. This feature gives you complete control over the values and parameters of all components in a circuit.

Circuit files created by CircuitLogix contain details such as copies of model files, settings of instruments, and analysis options. As a result, your circuit files are highly reusable and can be synchronized with later versions of libraries and for export. CircuitLogix is a valuable tool at all stages of the design process. At the conceptual stage, use CircuitLogix to explore your ideas and preliminary thoughts. Then, utilize this conceptual work by refining your CircuitLogix file to encompass your full design layout. If your circuit requires components from another software design package, you can import the standard SPICE netlists from other vendors directly into your CircuitLogix circuit files. As you move from design to production, you can use CircuitLogix to export files in the format used by PCB layout packages such as Tango.



CircuitLogix is a true 32-bit program intended for use with Windows 98, NT, XP, and Vista. It is very easy to use and even someone with a minimum amount of circuit theory can successfully use it to design circuits. In fact, by using CircuitLogix you can practice designing circuits on your own and thus greatly increase your expertise in this area. Learning from mistakes can be instructive, but when the subject matter is electronics, it can also be dangerous. With CircuitLogix you never need to be concerned about plugging components in backwards or hooking up wires incorrectly like you would if you were using real parts on a breadboard. You will never blow out your expensive components either, and best of all you will be able to construct circuits in much less time than is required on a real breadboard.

If you are mainly interested in electronics theory, or if you don't have access to an electronics laboratory, CircuitLogix can replace an entire electronics laboratory by providing accurate simulation of real-world components, circuits, and instruments. Equipping and maintaining an electronics laboratory can be very expensive. CircuitLogix simulates a lab at a fraction of the cost, and the endless supply of unbreakable components reduces the ongoing cost of materials. If you have limited access to real-world laboratory equipment, CircuitLogix allows you to make the most of your lab time by designing and testing circuits before entering the lab.

For the instructor, electronic principles and practices can be demonstrated quickly and easily using CircuitLogix. An exciting educational feature of CircuitLogix is the ability to place faulty devices in circuits. This allows you, as the instructor, to create troubleshooting exercises for students. The fault simulator lets you create multiple, complex faults to challenge student's knowledge of devices and circuits and re-inforce key concepts. Circuit files have been designed for most of the major electronics textbooks and are available from Logic Design Inc. In addition to circuit files, Logic Design Inc. also provides instructor resources such as course management software, testing and assessment, and other electronics curriculum support tools.

Required Hardware/Software

In order to use CircuitLogix you will need:

- IBM® compatible Pentium I or higher PC with a hard disk drive and a CD-ROM drive.
- 10MB of available hard disk space (15MB during installation).
- Mouse or other pointing device.
- Microsoft® Windows 98, NT, XP, or Vista.

First Time Installation

1. Start your Windows operating system.
2. Insert CircuitLogix CD-ROM into CD-ROM drive.
3. Autorun feature will automatically install CircuitLogix software.
4. When the installation is complete, a window containing a CircuitLogix icon will be displayed. Double-click on the CircuitLogix icon to launch the program.

Multi-user (Project) Installations

CircuitLogix can be configured to allow “Projects” for multiple users, each user having access to separate libraries and preferences. Project installations are possible whether on a network or on a stand-alone system. Each user must be assigned a separate directory from which the CircuitLogix preferences, libraries and circuit files can be accessed. This same method may be used by a single user who needs to access multiple projects, each with separate libraries and preferences.

To Set Up Multiple Projects:

1. Install *and run* CircuitLogix to initialize the default file paths, then exit.
2. Create a separate directory for each user or project.
3. Place a copy of the Cirlogix.dat file in each user’s project directory. This file contains the Preferences data which allows each user to specify their own circuit and library paths and other circuit and program preferences.

4. If a user will need to make modifications or additions to the Macros library (changes that must not affect other users), place a copy of the User.lib, Devicedb.dat, Symboldb.dat and Hotkeysdb.dat files into that user's project directory. The Devicedb.dat, Symboldb.dat and Hotkeysdb.dat files must always be placed in the same directory as User.lib.
5. If a user will need to make modifications or additions to the SPICE models or subcircuits, place a copy of the entire Models directory into that user's project directory.
6. If accidental modification of files which are common to multiple users is a concern, see your network administrator for details on how to protect these files from modification.
7. Double-click on the CircuitLogix icon to launch the program.
9. The Select Project dialog box will appear allowing you to find the individual project directories. Browse the directories to find one individual's Cirlogix.dat file and click on the OK button.
10. When CircuitLogix has loaded, select "Preferences..." from the File menu, then click on the Directories and Files... button.
11. Change the Circuit Directory path to that of the project directory. This is where that individual's circuit files (*.CKT) will be stored.
12. If there is a copy of the User.lib file in the project directory, change the User Library File path to that of the individual's directory. This allows the individual to make changes/additions to the Macros library.
13. If there is a copy of the Models directory in the project directory, change the Model Directory path to that of the individual's directory. This allows the individual to make changes/additions to the SPICE models and subcircuits.

Note: *A site licence is required for network installation.*

14. Click on the OK button to exit the Directories and Files dialog box, then click on the OK button in the Preferences dialog box to save these changes.
15. Exit CircuitLogix by selecting “Exit” from the File menu.
16. CircuitLogix is now completely configured for one user. Repeat steps 8-15 to configure CircuitLogix for each individual user.

To Access a Project:

1. Double-click on the CircuitLogix icon to launch the program.
2. The Select Project dialog box will appear allowing you to find your personal project directory. Browse the directories to find your Cirlogix.dat file and click on the OK button.

Using On-line Help

CircuitLogix on-line help file can be accessed in several ways.

Select “CircuitLogix Help Topics” from the Help menu:

Browse through the Contents or search for a specific subject.

Press the F1 key on the keyboard:

- If the circuit window is the active window, Contents help will be displayed.
- If an analysis window is the active window, help for that analysis type will be displayed.
- If a dialog box is open, help for that dialog box will be displayed.
- If the Help file is active, Help on Help will be displayed.
- If the cursor is over a button on the Toolbar, help for that button will be displayed.

Select the Help Tool in the Toolbar:

The cursor will be changed to the Help Tool. You may click on any item in the circuit to get information about that item.

Double-click on the Help icon:

Even when CircuitLogix is not running, you can view the Help file by double-clicking on its icon in the CircuitLogix program group.

Where to go from here

If you are familiar with the Windows™ environment and electronic circuit design and generally prefer to go right to work, please do so, referring to Chapters 2-12 as you need help. If you are mainly interested in drawing schematics, then begin by reading Chapters 2, 3, 10 and 11. If you are interested in simulation, read Chapters 4, 5 and 6. To help solve problems using the SPICE analog simulator, refer to Appendix A. Frequently Asked Questions can be found in Appendix B.

Technical Support

Logic Design Inc. is dedicated to producing only the finest quality software and supporting the customer after the initial purchase. Appendix B contains a list of commonly asked questions. Please look these over before contacting us for technical support. If you encounter other problems while using CircuitLogix or just need general help, contact us via phone, e-mail, or regular mail.

Telephone:	(877) 687-7999
Internet:	http://www.CircuitLogix.com
Email:	support@circuitlogix.com
Mail:	Logic Design Inc. 1011 Boren Ave. Seattle, WA, 98104

Future versions of CircuitLogix are planned, so please feel free to write and let us know what features or additions you would like to see. Our goal is to provide a product that will meet your needs and expectations, so feedback from you is essential if we are to achieve this goal!

Required User Background

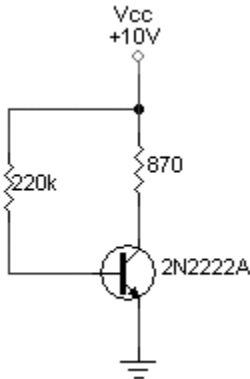
This manual assumes that you are already familiar with the Windows™ environment and that you know how to use the mouse to open windows, start applications, select a file from the file selector dialog, and select menu items from pull down menus. If this is not the case please read and study the users manual that came with Windows before trying to use CircuitLogix.

Chapter 2

Getting Started

Schematic Example

Drawing a schematic is as easy as pointing and clicking with the mouse. Let's walk through a simple example by constructing the circuit shown at the left.

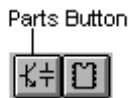


This is a very simple circuit that consists of one transistor, two resistors, a power source and a ground. Follow these steps to build the circuit:

1. Click on the “New” button in the Toolbar to clear the work area.
2. Select “Browse...” from the Hotkeys1 menu or click on the Parts button in the Toolbar. The Device Selection dialog box will appear. We will indicate the location of devices in the library by their major and minor class and, where applicable, their default hotkey using the format:

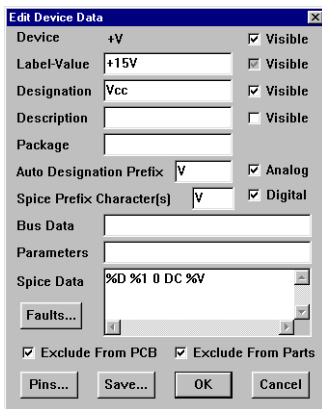
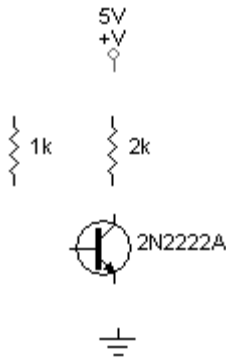
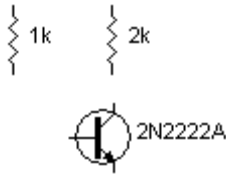
[major device class/minor device class] (default hotkey)

For example a battery could be found at [Analog/Power] (b). Devices which have hotkeys assigned can also be found in the Hotkeys menus and can be selected by simply pressing the desired hotkey on the keyboard.



Let's begin with the “2N2222A” transistor [Active Components/BJTs]. Select “Active Components” in the Major Device Class list. Select “BJTs” in the Minor Device Class list. Select “NPN Trans:C” in the Device Symbol list. Select “2N2222A” in the Model/Subcircuit list. Now click on the Select button to select this device from the library. Position the transistor at about mid-screen and then click the left mouse button once. The transistor will be placed on the work area and will no longer follow the mouse.

3. Enable the “Repeat On” feature in the Options menu (so it has a check mark by it).
4. Select a “Resistor” [Passive Components/Resistors] (r) by



pressing the “r” key on the keyboard. Notice that the resistor is oriented horizontally. Press the “r” key on the keyboard again (or click on the RIGHT mouse button) to rotate the device 90°. Position the resistor above and to the left of the transistor. Now click the LEFT mouse button once to place the device. This will be RB. Don’t worry about the value yet.

5. Since the Repeat On feature has been enabled, another resistor will appear with the same orientation as the previous one. Place this resistor directly above the transistor. This will be RC.

6. Another resistor will now appear. Press any key on the keyboard (except R or M) to delete it.

7. Press Ctrl+R to disable the Repeat On feature.

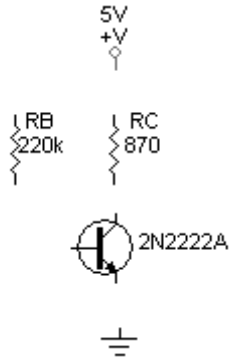
8. Select a “+V” [Analog/Power] (1) by pressing the “1” key on the keyboard. Place it above resistor RC.

9. Select a “Ground” [Analog/Power] (0 (zero)) by pressing the “0” key on the keyboard. Place it below the transistor.

10. Double-click on the +V device using the LEFT mouse button. This will open the Edit Device Data dialog box. Change the Label-Value field to read “+15V”. Click on the Visible check box next to the Label-Value field to change the black check mark to a gray check mark. This will cause the “+15V” to *replace* the “+V” on the schematic. Enter “Vcc” in the Designation field and click once in the Visible check box next to it so that there is a black check mark in it. Click on the OK button to exit dialog box.

11. Double-click on resistor RB. Change the Label-Value field to read “220k”. Enter “RB” into the Designation field and make it visible. Click on the OK button to exit.

12. Double-click on resistor RC. Change the Label-Value field to read “870”. Enter “RC” into the Designation field and make it visible. Click on the OK button to exit.

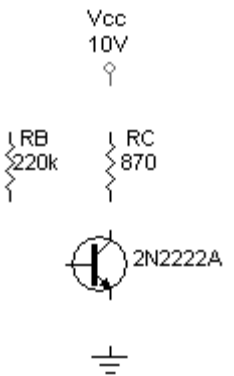


13. Now, double click on the transistor. The Model Selection dialog box will be displayed. Since we have already selected the model that we want displayed (2N2222A), just click on the Netlist button to open the Edit Device Data dialog box. Enter Q1 into the Designation field and make it visible. Click on the OK button to exit, then click on the Exit button to return to the schematic.

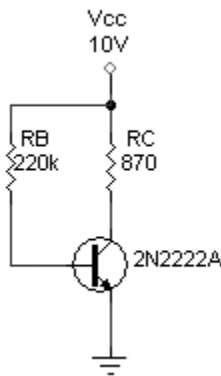
14. If needed, drag the devices and labels around with the mouse to place them in convenient locations. The position of labels can be cleaned up, if necessary, by using the "Place Selected Label(s)" command in the Edit menu.



15. You'll now wire the circuit together. Select the Wire Tool from the Toolbar. Place the cursor on the emitter pin (the pin with the arrow) of the transistor. When the cursor gets close to the pin, a small rectangle will appear. *Click and hold* the left mouse button, then drag the wire to the pin of the Ground symbol. Release the mouse button to make the connection. If "Show Pin Dots" is enabled (has a check mark by it) in the View menu, a small dot will be placed at each connection point to verify the connection (see circuit example at left).



16. Now, place the cursor on the bottom pin of RC, then *click and hold* the mouse button to start a new wire. Drag the end of the wire to the collector pin of the transistor and release the mouse button. Next, connect a wire from the top pin of RC to Vcc. Connect another wire from the bottom pin of RB to the base of the transistor. Finally, connect a wire from the top pin of RB to the middle of the wire which connects Vcc to RC. Device and wire positions can be moved by dragging them with the mouse.



The Editing Tools

The Toolbar includes several buttons which deal specifically with circuit layout. These buttons are described here.



The Editing Tools

The Arrow Tool:

The Arrow Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+A. Use the Arrow Tool to select items, move items, flip switches, select tools from the Toolbar, etc. You can double-click (click twice with the mouse in rapid succession) with the Arrow Tool to perform many functions, such as editing specific devices, etc.

If the “Arrow/Wire” option is enabled in the Options menu, the Arrow Tool may be used to initiate a wire when you click on a device pin.

The Wire Tool:

The Wire Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+W. Use the Wire Tool to place wires in the work area. Bus wires are drawn by holding down the SHIFT key when starting to draw the wire. Refer to the sections *Wiring the Circuit* and *Using Bus Wires* later in this chapter.

The Text Tool:

The Text Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+T. Use the Text Tool to place text in the circuit. Just select the tool, click in the work area and type the text. Use the “Select Font” option in the Edit menu to stylize the text. You can alter the way multi-line text wraps by clicking on it with the Text Tool and resizing its enclosing rectangle.

The Delete Tool:

The Delete Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+D. Use the Delete Tool to selectively delete items. Simply select the Delete Tool and click on the item to be deleted. The item will be immediately deleted except in the case of a wire. For wires, when the mouse button is held down, the wire will be highlighted but not deleted until the button is released. If you hold

down the mouse button and move the Delete Tool away from the wire, the delete will be cancelled.

Any selected item can also be deleted by pressing the Delete key on the keyboard.

A wire may be cut (divided) in two by holding down the SHIFT key and clicking on the wire with the Delete Tool.

The Naming Tool: 

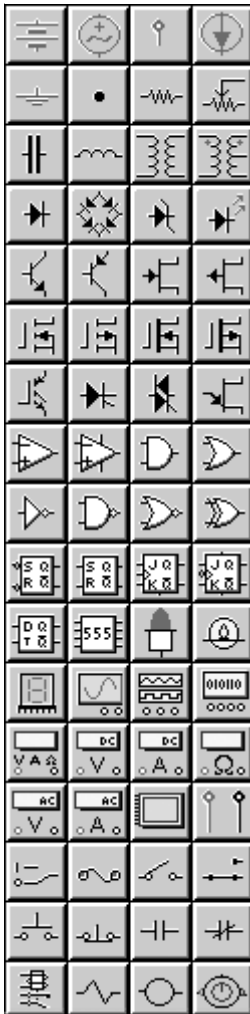
The Naming Tool can also be selected from the “Cursor Tools” option in the Options menu (Name Item) or by pressing Alt+N. Use the Naming Tool to edit the names of SCOPES, Inputs, Outputs, bus wire numbers and bus connection wire numbers. Just click the pointer on the item to be edited and a dialog box will appear. Macros must be expanded to be edited. A macro that is saved with a new name will not delete the original macro.

Tip: Many of the items which can be edited by the Naming Tool can also be edited by double-clicking on them with the Arrow Tool.

The Zoom Tool: 

The Zoom Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+Z. This tool allows you to magnify (zoom in) and reduce (zoom out) the circuit being displayed. To zoom in, select the Zoom Tool and position it over the area that is to be enlarged. Click the left mouse button to magnify the circuit by the selected Scale Step. To zoom out, select the Zoom Tool and position it over the area that is to be reduced. Hold down the SHIFT key and click the left mouse button to reduce the circuit by the selected Scale Step Size.

Alternately, you can press the Page Up key on the keyboard to zoom in or the Page Down key to zoom out at any time, without using the Zoom tool. The zoom will be centered on the position of the mouse cursor. See also *Display Scale, Normal Size/Position* and *Fit Circuit To Window* in the View Menu chapter.



Parts Button



The Rotate 90 Button: 

The Rotate 90 button can also be selected from the Edit menu or by pressing Alt+R. Use the Rotate 90 button to rotate the selected device in 90° increments.

A device may also be rotated when it is selected from the library by pressing the R key on the keyboard or by clicking the RIGHT mouse button before placing it in the circuit.

The Mirror Button: 

The Mirror button can also be selected from the Edit menu or by pressing Alt+M. Use the Mirror button to flip the device horizontally.

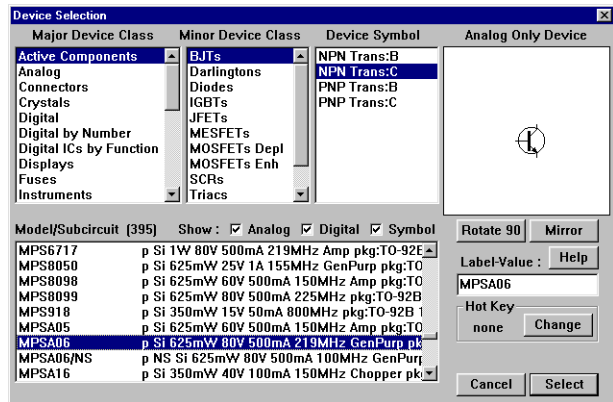
A device may also be mirrored when it is selected from the library by pressing the M key on the keyboard before placing it in the circuit.

Placing Devices

The main device toolbar shown on the left side of the screen provides access to 68 of the most commonly used devices and instruments. Components are selected from this toolbar by left-clicking on the desired component with your mouse, and then moving the mouse cursor onto the work area and left-clicking again to place the component.

A much more extensive listing of the available devices is contained in the device library. Parts can be selected from the library using the Device Selection dialog box. Commonly-used devices can be added to the Hotkeys menus, allowing you to select the device by pressing one of the user-definable hotkeys on the keyboard.

To invoke the Device Selection dialog box, click on the Parts button in the Toolbar, select “Browse...” from the Hotkeys1 menu, or press the “x” key on the keyboard. The following dialog box will be displayed.



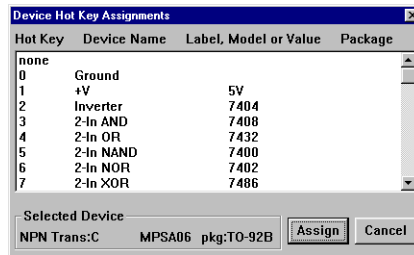
Locate the device you wish to use by first selecting a Major Device Class, a Minor Device Class, a Device Symbol, and if required, a SPICE model or subcircuit. In this manual, device location is indicated as [major class/minor class]. For example, a 2N3904 could be found at [Active Components/BJTs]. Click on the Select button to select the device for placement in your circuit (or just double-click on the item to be selected in the Device Symbol list or the Model/Subcircuit list).

The Show Analog, Digital and Symbol check boxes allow you to filter the devices to reduce the number of items to be searched. If only Analog is checked, only the devices which function in analog simulation mode will be displayed. If only Digital is checked, only the devices which function in digital simulation mode will be displayed. If only Symbol is checked, only the nonfunctional symbols will be displayed. For example, if you are looking for digital devices to be used in analog simulation mode, you could uncheck both Digital and Symbol to hide all the devices which will not run in analog simulation mode. At least one of these boxes must be checked at all times.

The symbol that corresponds to the selected item in the Device Symbol list is also displayed. Rotate 90 and Mirror buttons are provided in the dialog box to allow you to view each device in other orientations.

60 of your most commonly used devices can be assigned hotkeys which allows you to quickly select these devices, either by pressing the hotkey on the keyboard or by selecting the item from the Hotkeys menus.

To assign a hotkey to a specific device, select the device by clicking on it once so that it is highlighted in the list. Then, click on the Change button to access the following dialog box.



The hotkeys are listed alphabetically along with the devices that are currently assigned to each hotkey. Scroll through the list to find the hotkey that you want assigned to your selected device, then click on the Assign button. That device will be included in the Hotkeys menus and can now be selected from the Hotkeys menu or by pressing the assigned key on the keyboard. If you want to remove a device from the hotkey list, you can either assign a new device in its place or assign “none” as its hotkey.

Once you have selected a device to place in your circuit, it will follow the mouse around the screen until you click the LEFT mouse button. If you press the R key on the keyboard or click the RIGHT mouse button before clicking to place the device, the device will be rotated. If you press the M key on the keyboard before clicking to place the device, the device will be mirrored. If the “Repeat On” option is enabled in the Options menu, as soon as a device is selected from the library and placed in the work area, another identical device is automatically selected and made ready for placement. This repeat placement process continues until you cancel it by pressing any key on the keyboard or by double-clicking the mouse.

Use the Arrow Tool to select and move devices around the work area. If parts are to be rotated or mirrored, this should be done prior to connecting wires to them.

Refer to the Device Library chapter for a description of the basic devices and instruments.

There are four distinct ways of selecting items in the circuit window:

- To select a single item click on it with the Arrow Tool. Click anywhere else in the work area to deselect the item.
- To select multiple items hold down the SHIFT key and click on one or more items with the Arrow Tool. To deselect any single item click on it a second time while still holding down the SHIFT key. To deselect all items, release the SHIFT key and click somewhere in the work area away from all items.
- To select switches or multiple items hold down the mouse button and drag the selection rectangle around the desired items. To deselect any single item, SHIFT-click on it. To deselect all items click somewhere in the work area away from all items.
- To quickly select all items use the “Select All” command from the Edit menu.

Wiring the Circuit

Your circuit must be correctly wired together in order to function. **The circuit will not function if you simply place devices so that device pins are touching the pins of other devices where you want connections to occur!** Instead, you must connect a wire between the pins you want connected together. All wires must be vertical, horizontal, or combinations of these two. It is not possible to wire at odd angles such as 45° or 60°.

There are two basic methods of wiring: Auto Routing and Manual Routing. These two methods are fully integrated; you don't have to switch between wiring modes.

Auto routing has the advantage of being quick and easy to use. Simply place the Wire Tool (or the Arrow Tool if the “Arrow/Wire” option is enabled) on a valid connection point (a valid connection point is any device pin or wire), then **click and hold** the left mouse button. Drag the mouse to another valid connection point and release the mouse button. The wire is automatically routed between the two points.

Tip: You can initiate a wire by clicking on a device pin with the Arrow Tool when the “Arrow/Wire” option is enabled in the Options menu.

Auto routing requires two valid connection points. You cannot draw a wire with auto routing that does not connect to something on both ends. Also, you cannot draw bus wires with the auto routing method. The auto routing can be either Simple or Intelligent, depending on what you have selected in the Preferences dialog box. Simple routing draws only one or two wire segments, horizontal and/or vertical, making the shortest path without regard for devices that might be in the way. Intelligent routing tries to find a path which does not cross directly over any devices. If no reasonable path can be found, the simple method is used.

Manual routing has the advantage of routing wires exactly where you want them without need for readjustment after the wire is drawn. It also allows you to place wires in your circuit which are not connected on both ends. Manual routing must be used to draw bus wires.

Move the Wire Tool (or the Arrow Tool if the “Arrow/Wire” option is enabled) to the position where you wish to start the wire, then **click and release** the left mouse button. The Wire Tool cursor will disappear until you begin to drag the mouse. It will then be replaced by an extended wiring cursor. The extended cursor simplifies the task of precisely aligning wires with other objects. Click once with the left mouse button to turn 90° or double-click to end the wire. If the “Single Click Connect” option is enabled in the Preferences dialog box, a single click of the mouse will terminate the wire when it is at a valid connection point. A wire can be cancelled at any time while it is being drawn by pressing any key.

The SmartWires™ feature allows you to connect a wire to a device pin or another wire without being in exactly the right place. A user definable connection area exists around each valid connection point (a valid connection point is any device pin or wire). When the Wire Tool enters a connection area, a rectangle is displayed, highlighting the connection point. The Preferences dialog box may be used to set the size of the connection area and whether or not the rectangle is displayed.

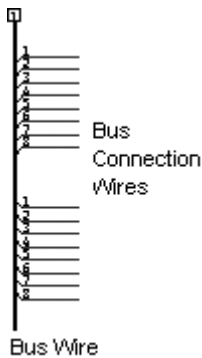
When a device that has wires connected to it is moved (dragged with the Arrow Tool), the wires will “rubberband” (they will stretch and remain connected to the device).

Wires may be extended or joined. To extend a wire, select the Wire Tool and place it over the END of the wire and draw a wire. To join two wires together, draw a wire from the end of the first wire to the end of the second wire. They then become the same wire.

A single wire can also be cut (divided) into two separate wires. To cut a wire, select the Delete Tool, place it over the wire that is to be cut, hold down the SHIFT key and click the left mouse button. The wire will be separated into two individual wires.

Using Bus Wires

Bus wires are a special type of wire that contain multiple individual wires. Each bus wire is identified by a number and each individual wire within a bus is also identified by a number. Bus wires can be easily identified in a circuit because they are drawn thicker than regular wires.



To draw a bus wire, hold down the SHIFT key and draw a regular wire using the manual routing method (see *Wiring the Circuit* earlier in this chapter). The SHIFT key must be down before a wire is started but it can be released before the wire is finished. After the bus wire has been drawn, a dialog box will prompt for a bus number. Each unique bus must have a unique bus number. If two separate buses are given the same number, they are considered to be the same bus, even though they are not physically connected on the screen. Bus wires can also be extended, joined, or cut, just like regular wires. You don't need to hold down the SHIFT key when extending bus wires.

Regular wires that are connected to a bus wire are called bus connection wires. These wires connect to the individual wires within the bus. When the connection is made a dialog box will prompt for a wire number and the connection angle. Each bus connection wire is assigned a number. Each bus connection wire that is assigned the same number and is connected to the same bus wire (or another bus wire with the same bus wire number) will function as though they were connected together.

Bus wire numbers and bus connection wire numbers may be edited by clicking on them with the Naming Tool or by double-clicking on them with the Arrow Tool.

Labeling the Circuit

Various methods of labeling the circuit are available. One method is to select the Text Tool, click in the work area where you want the text to appear, and type in the text. Text entered by this method can be multi-lined and fully stylized. It may also be repositioned at any time and is always visible.

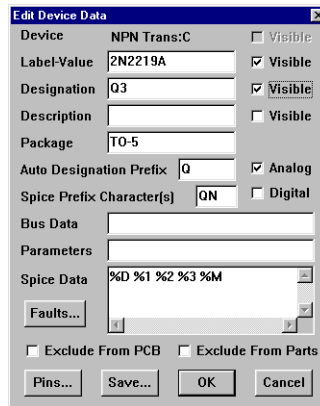
Each of the devices may also be labeled using the dialog box that appears when you double-click on the device. Labels entered by this method may be positioned anywhere around the device by dragging them with the Arrow Tool and remain attached to the device when it is moved around the work area. For a detailed description of these labels, refer to the following section, *Netlist Information*.

Bus labels and bus connection labels are enabled by default, but can be disabled by unchecking the “Show Bus Labels” option in the View menu.

Netlist Information

CircuitLogix has the ability to produce a SPICE netlist which is fully compatible with the Berkeley SPICE3 simulation software. CircuitLogix analog simulation is based on the Berkeley SPICE3, so to make the analog simulation work you must supply specific information about each component in the circuit. We discuss the information here because it can also be useful for simple schematic layout, even if you are not simulating the circuit.

The user double-clicks on a device to bring up the following dialog box. (On some devices, a different dialog box will be displayed. Look for a “Netlist...” button in those dialog boxes to get here.) This dialog box is used to enter data about the device. Following is a description of the various items.



- **Device:** This is a non-editable field which shows the name of the device as it appears in the library menus. The Visible check box can be used to hide the name on the schematic drawing. If visible, it will retain the same orientation as the device when rotated. Some device names, such as “Resistor”, cannot be made visible.
- **Label-Value:** This field may be used to enter information about the device such as its label (1N914, 2N3904, etc.) or its value (47K, 100U, etc.), or to replace the existing Device name (make the Device not visible and the Label visible). The Label-Value can be dragged around the device on the schematic with the mouse and will remain attached to the device when the device is moved. If the Visible check box is set to gray, the Label will directly replace the Device name and retain the same orientation as the device when rotated.
- **Designation:** This field is used to identify the device in the circuit such as U3, CR7, RLOAD, etc. This field may also be made visible with the Visible check box. This field must contain the part designation in order for netlisting to work properly.

This field will be filled in automatically when you select the “Set Auto Designation...” command from the Edit menu. Also, if the “Auto Designation” option is enabled in the Options menu, it will be filled automatically each time analog simulation is run. When the Auto Designation is run, the previous contents of this field are lost. See *Auto Designation Prefix* below.

For multipart packages, the individual parts must be grouped by using the “Group Selected Items” command in the Edit menu. Each individual part must be identified as PART A, PART B, etc., from the Edit Device Pin Data dialog box. See *Pins* later in this chapter.

- **Description:** This field is used for schematic reference only. You may use it to display additional information such as custom part numbers, tolerances, etc. This field does not affect simulation. This field may also be made visible with the Visible check box. The Description label may also be dragged around on the schematic with the mouse.
- **Package:** This field is used to identify the type of package the device is in (DIP14, TO-92B, etc.). This field is used when creating netlist files and the Package entered here should match the name of the corresponding component in the library.
- **Auto Designation Prefix:** This field is used to specify the prefix used in the device’s designation when “Set Auto Designation...” is selected from the Edit menu. The prefix may be up to 4 characters in length.
- **Spice Prefix Character(s):** This field is used to specify the SPICE prefix used in conjunction with the %D and %M flags described below under *Spice Data*. By specifying this prefix, user defined macro symbols can be linked to the proper model selections. Valid prefixes are:

A	XSpice Model
BV	Nonlinear Dependent Voltage Sources
BI	Nonlinear Dependent Current Sources
C	Capacitors
D	Junction Diodes
DZ	Zener Diodes
E	Linear Voltage-Controlled Voltage Sources
F	Linear Current-Controlled Current Sources
G	Linear Voltage-Controlled Current Sources
H	Linear Current-Controlled Voltage Sources
I	Independent Current Sources
JN	JFETs (N-channel)

JP	JFETs (P-channel)
K	Coupled (Mutual) Inductors
L	Inductors
MN	MOSFETs (N-channel)
MP	MOSFETs (P-channel)
O	Lossy Transmission Lines
QN	Bipolar Junction Transistors (NPN)
QP	Bipolar Junction Transistors (PNP)
R	Resistors
S	Voltage Controlled Switches
T	Lossless Transmission Lines
U	Uniform Distributed RC Lines (Lossy)
V	Independent Voltage Sources
W	Current Controlled Switches
X	Subcircuits
ZN	MESFETs (N-channel) (GaAs FETs)
ZP	MESFETs (P-channel) (GaAs FETs)

- **Analog:** This check box identifies this as an analog device. The analog simulator can only simulate a device if there is SPICE simulation data for that device. If you attempt to run an analog simulation using a device that does not have the Analog check box checked, CircuitLogix will display a warning message and that device will be ignored in the simulation. When creating your own device, this box should be checked only if you have supplied SPICE data for the device or if the device is a macro circuit with other analog devices.
- **Digital:** This check box identifies this as a digital device. The digital simulator can only simulate a device if there is digital simulation code for that device. If you attempt to run a digital simulation using a device that does not have the Digital check box checked, CircuitLogix will display a warning message and that device will be ignored in the simulation. When creating your own device, this box should be checked only if the device is a macro circuit containing other digital devices.
- **Bus Data:** This field is used to specify which pins on the device are connected to the power or ground buses, since these pins are not shown on the predefined device packages. The general format for this data is:

busnam[=pinnum[,pinnum,...]];[busnam=pinnum[,pinnum,...]:...]

For example, on a 74LS83 the Bus Data is “DVCC=5;DGND=12;”. This means that the Vcc bus is connected to pin 5 and the Ground bus is connected to pin 12. The bus data for a 74AC11190 would look like “DVCC=15,16; DGND=4,5,6,7;”. The order in which the buses are listed is not important.

In order to connect these bus pins for analog simulation or for creating a PCB netlist to export into TraxMaker, each bus must be defined. There are three devices that can be used for this purpose: +V, Ground and Terminal. The Bus Data format for these devices is simply:

```
busnam;
```

where busnam identifies the specific bus. For example, each Ground device, by default, contains the Bus Data “GND;” which causes all of the Ground symbols to be tied together in a single node or net. *Note: For simulation purposes, the Ground device ALWAYS equates to Spice node 0, even if the Bus Data changes.*

A +V device does not contain any default Bus Data. This is because each +V device may represent a different supply voltage. However, the Bus Data field can also be used on these devices to identify it as the voltage source for the digital devices. For this example, set the Bus Data to “DVCC;” or “DVDD;”.

In some cases it may be necessary to connect a different voltage source (such as the output of a voltage regulator) to the digital devices. In this case, connect a Terminal device to the power source and enter “DVCC;” in its Bus Data field.

See also *Vcc and Ground* in the Analog/Mixed-Mode Simulation chapter.

- **Parameters:** This field is used to store information that affects the simulation of certain devices. For digital simcode devices, this field would contain “type:digital”. It could then be followed by a list of parameters which are generally set in the Digital Model Parameters dialog box.

Some generic device models can be redefined by passing parameters as an alias to describe a specific device. For example, the parameter field of a crystal would contain “alias:XCRYSTAL” and could be followed by a list of databook parameters that define the specific crystal. These parameters are generally entered in the Subcircuit Parameters dialog box.

- **Spice Data:** This field is used to specify the SPICE simulation data for the device. The analog devices provided with CircuitLogix already have default SPICE data included. If you create your own devices to use with the analog simulation, you will need to fill in this field yourself. The SPICE data may be entered into this field directly, or it can reference the other fields in the dialog box. The percent sign (%) is used as a flag to tell CircuitLogix to reference the already defined fields. Their meanings are:

%N **Name**—Insert the Device Name into the SPICE data string. This is the name found in the library menus.

%L **Label**—Insert the Label-Value into the SPICE data string. This requires the first character in the Label-Value string to be an alpha character. The label may not exceed 8 characters.

%V **Value**—Insert the Label-Value into the SPICE data string. This requires the first character in the Label-Value string to be numeric. The value may be an integer (12, -44), a floating point number (3.14159), either an integer or floating point number followed by an integer exponent (1e-14, 2.65e3), or an integer or floating point number followed by one of the following multipliers:

T	= 10 ¹²	u	= 10 ⁻⁶
G	= 10 ⁹	n	= 10 ⁻⁹
Meg	= 10 ⁶	p	= 10 ⁻¹²
K	= 10 ³	f	= 10 ⁻¹⁵

$$m = 10^{-3}$$

If multipliers are used, they must immediately follow the number with no spaces. Letters immediately following a number that are not multipliers are ignored, and letters immediately following a multiplier are ignored. Hence, 10, 10V, 10Volts, and 10Hz all represent the same number and M, MA, Msec, and MMhos all represent the same multiplier. Note that 1000, 1000.0, 1000Hz, 1e3, 1.0e3, 1KHz, and 1K all represent the same number.

- %M Model**—Insert the Label-Value into the SPICE data string. If the first character of the Spice Prefix Character(s) does not match the first character in the label, then insert the first character of the prefix at the beginning of the string. The %M is also required in order to guarantee that the .MODEL data for this device will be included in the SPICE netlist file for simulation. The Label (including prefix) may not exceed 8 characters.
- %S Subcircuit**—Insert the Label-Value into the SPICE data string. If the first character in the label is not an “X”, then insert an “X” at the beginning of the string. The %S is also required to ensure that the .SUBCKT data for this device will be included in the SPICE netlist file for simulation. The Label (including the “X”) may not exceed 8 characters.
- %D Designation**—Insert the Designation label into the SPICE data string. If the first character of the Spice Prefix Character(s) does not match the first character in the designation, then insert the first character of the prefix at the beginning of the string.
- %I Description**—Insert the Description field into the SPICE data string.
- %P Package**—Insert the Package label into the SPICE data string.
- %B Bus Data**—Insert the entire Bus Data field into the SPICE data string.

%X Named subcircuit—Insert the specified subcircuit into the SPICE data string. If the subcircuit resides in a .SUB file other than the one associated with the symbol name of the current device, the .SUB file name where the subcircuit is located must also be specified. For example: “%XUA741” or “%XUA741:OPAMP5”.

%=“*path\filename.ext*” Include File—Insert the ASCII text file *filename* into the SPICE data string by using SPICE’s .INCLUDE command. Path is the same as the current circuit unless enclosed in quote marks (“”).

%*pinorder* Node—Insert the node number for the specified pin into the SPICE data string. Pinorder refers to the position of the pin in the Edit Device Pin Data dialog box. For example, the pin at the top of the list is represented by %1, the next pin down is %2, etc. See *Pins* later in this section.

The following examples show how the Spice Data field is used when creating custom devices:

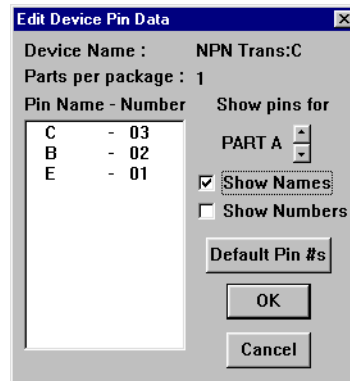
If resistor R3 has a value of 27 ohms and is connected between node 5 and ground, the SPICE data for R3 could be written as “R3 5 0 27ohms”. This data may be entered directly into the Spice Data field, but it would be difficult to determine the node numbers to use. A more universal method would be to enter the string “%D %1 %2 %V”.

If transistor Q2 is a 2N3904 with its collector connected to node 7, its base to node 4 and its emitter to node 12, the SPICE data may be written as “Q2 7 4 12 Q2N3904”, but the SPICE model data may need to be entered into the netlist manually. A more universal method would be to enter the string “%D %1 %2 %3 %M”.

If op amp U4 is an LM741 where the +input is connected to node 3, the -input is connected to node 1, the +V pin to node 5, the -V pin to node 12 and the output to node 9, the SPICE data may be written as “XU4 3 1 5 12 9 XLM741”, but

again, a more universal method would be to enter the string “%D %1 %2 %3 %4 %5 %S”.

- **Faults...:** Clicking on this button brings up the Device Faults dialog box for adding fault data to the device. If the fault data has been password protected the Access Faults dialog box will be displayed. See the Fault Simulation chapter for details.
- **Exclude From PCB:** This checkbox is used to exclude a device from the PCB netlist (see *Script Functions* in the File Menu chapter).
- **Exclude From Parts:** This checkbox is used to exclude a device from the parts list.
- **Pins...:** Clicking on this button brings up the following dialog box which allows you to edit the pin numbers of the package for the selected device and to determine whether the pin numbers will be shown on the schematic.



Some device packages actually contain more than one of the same device. For example, a 7400 Quad 2-Input NAND gate actually has 4 gates in the same package. The pin numbers are different for each gate. The individual gates are grouped together to indicate which gates go in which package. As each gate is placed in the circuit, the next available gate is used from the previous package. If no gates are available in the previous package, a new package is used. The gates can later be regrouped as needed using the “Group Selected Items” command in the Edit menu. Each gate in the package is assigned a letter (A, B, C, etc.) and the pin numbers for

that gate correspond to that assignment. The pins used for a particular gate can be manually reassigned by selecting the appropriate letter (PART A, PART B, etc.) for that gate with the up and down arrows in the dialog box.

Default pin data has already been entered for the predefined devices. Clicking on the Default Pin #s button will restore the pin numbers to their original default values. Default pin numbers may be added to a macro device by editing the pin numbers while the macro is expanded and saving the macro. Macro pin names and numbers may be edited by double-clicking on them in the Symbol Editor. The order in which the pins appear in the list is determined by the order in which the pins were placed on the device. To edit pin numbers for a specific part, double-click on them in this dialog box. This does not change the pin numbers in the library.

- **Save...:** Clicking on this button will temporarily set the default settings for this device type. Each time a new device of this same type is pulled from the library menus, it will have these settings. For example, if you set the resistor value to 1k ohms and click on the Save... button, each subsequent resistor pulled from the menu will have a value of 1k ohms. These settings will remain the default until replaced by another save, or until you quit the program altogether.

Printing and Exporting Circuits

When you're done designing and testing your circuit, it can be printed out on any Windows-selectable printer. If your design is larger than a single sheet of paper it will automatically be printed on multiple sheets of paper. Page breaks can be shown by enabling the "Show Page Breaks" option in the View menu. Other print options are also available, including print scaling and color printing. See *Print Setup* in the File Menu chapter. Digital timing diagrams and analog waveforms can also be printed.

CircuitLogix also provides easy-to-use mechanisms which enable you to export both the circuit and waveforms for use in

word processing, publishing, drawing, or other programs. Export of the circuit and waveforms is done using the Clipboard or by saving the circuit or waveforms in a bitmap file or metafile. See *Export Options* in the File Menu chapter.

Various script functions are provided to allow you to create PCB netlists, SPICE3 netlists and Parts lists. See *Script Functions* in the File Menu chapter.

Creating User-defined Schematic Symbols

CircuitLogix is fully expandable. By creating your own schematic symbols and adding them to the user library, you can adapt CircuitLogix to your specific needs. Refer to the next chapter, Macro Devices, for more information on creating both functional and nonfunctional devices.

Chapter 3

Analog/Mixed-Mode Simulation

SPICE:
Simulation
Program with
Integrated
Circuit
Emphasis.

One of the most important features of CircuitLogix is its ability to simulate your design, enabling you to detect and correct design errors before investing time and money in the construction of actual hardware prototypes.

Analog is the classic world of electronics. There are no logic state restrictions as in digital electronics; the voltage level of any given circuit node is not limited to a high or low. Analog simulation, therefore, is much more complex. CircuitLogix analog/mixed-mode simulation is based on an enhanced version of Berkeley SPICE3f5/XSpice. This combination provides the ability to simulate a wide variety of both analog and digital devices. The analog simulator generates a data set based on the analysis parameters selected by the user. This data can then be analyzed in the graphics windows.

Mixed-mode refers to the ability to mix both analog and digital devices in a single simulation. This is actually an extension of the analog simulation mode. Logic Design Inc. has created accurate, event-driven behavioral models for a wide variety of TTL and CMOS digital devices so that they may be simulated in analog mode. These digital models provide accurate characteristics for each device within their respective families.

All *Analog Only* and *Mixed-Mode* devices and instruments—capacitors, transistors, Signal Generators, etc., as well as the 4xxx and 74xxx devices—can be used with the analog simulator. In the Device Selection dialog box, above the picture of each device symbol, appear the words “Digital Only Device”, “Analog Only Device”, “Mixed-Mode Device”, or “Schematic Only Device”. This indicates the simulation mode for which the currently displayed device is intended. Refer to the Device Library chapter for a description of each device and the simulation mode for which it is intended.

Using the Analog Simulator

To do analog simulation there must be SPICE information for each device in the circuit. Only those devices listed as analog compatible in the Device Library chapter have SPICE data associated with them. Other devices may be used in the circuit, but the SPICE information for those devices must be provided by the user. SPICE information can be added in the Spice Data field of the Edit Device Data dialog box (double-click on the device to open the dialog box). The Analog check box in the Edit Device Data dialog box indicates whether or not there is SPICE simulation data available for the particular device or if it can be used in analog simulations. If the Analog check box is not checked and the device is used in analog simulation, a warning will be displayed and that device will be ignored, leaving an open circuit where that device is located.



Analog Simulation Tool

Analog simulation mode is selected when the transistor icon is displayed on the Digital/Analog button in the Toolbar.

The analog analyses are set up in the Analog Analyses dialog box described later in this chapter. By default, whenever you create a new circuit, the “Always Set Defaults” option is enabled for the analog analyses. This means that Operating Point Analysis (the digital meter) will be enabled for simple DC circuits. For more complex circuits, Transient Analysis (the oscilloscope) will also be enabled and set to its default conditions. Under normal conditions, these settings will rarely need to be changed.

Once the circuit has been created, run the simulation by clicking on the Run button in the Toolbar. An interactive SPICE simulation window will be displayed during the SPICE data collection process showing the progress of the simulation. When the SPICE data collection process has completed, a graph window will be displayed for each of the selected analyses.

Getting Started—Analog Circuit Examples

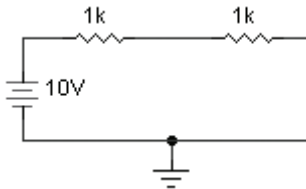
The best way to get acquainted with CircuitLogix analog simulation is to build a few simple circuits, set up the analyses, and run the simulations.



The New Button



The Digital/Analog Button



OFF



ON



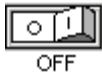
The Probe Tool

Simple Circuit Analysis

Let's begin with a simple DC circuit:

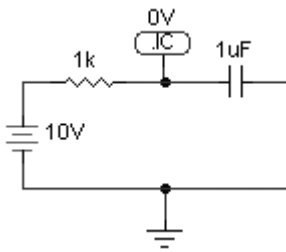
1. Click on the New button in the Toolbar. An "Untitled" circuit window will be opened.
2. Select Analog simulation mode. The transistor icon should be visible in the Toolbar, not the AND gate icon. If the AND gate icon is displayed on the button, click on the button.
3. Make sure the "Auto Designation" option in the Options menu is enabled (so there is a check mark by it).
4. Draw the circuit as shown, using the following devices. Click on the Parts button and select 1 Battery [Analog/Power] (b), 1 Ground [Analog/Power] (0 (zero)) and 2 Resistors [Passive Components/Resistors] (r). **Note: every analog circuit must have a Ground and every node in the circuit must have a DC path to ground.** Use the Wire Tool to wire the circuit together.
5. Select "Analog Options..." from the Options menu. Under the section "Analysis data saved in RAW file", select the third option, "Node V, Supply I, Device I, Device P". This option will allow you to make current and power measurements with the Probe Tool. Click on the OK button to exit.
6. Click on the ON/OFF switch located in the top right corner of the screen to switch on power to the circuit and start the simulation. An interactive SPICE simulation window will be displayed during the SPICE data collection process showing the progress of the simulation. When the SPICE data collection process has completed, the Value Window will be displayed.
7. Click on the wire connected to the + terminal of the battery with the tip of the Probe Tool. The DC voltage at that node (+10V) will be displayed in the Value Window. Click on the wire connected between the two resistors. The DC voltage at that node (+5V) will be displayed in the Value Window. SPICE data is not collected for the Ground node in the circuit; it is always at zero volts. Click on the + pin of the battery or on one of the resistor pins. The current through that device (5mA) will be displayed in the Value

Window. Click directly on one of the resistors. The power dissipated by that resistor (25mW) will be displayed in the Value Window.



8. Click on the ON/OFF toggle switch again to stop the simulation and return to editing mode.

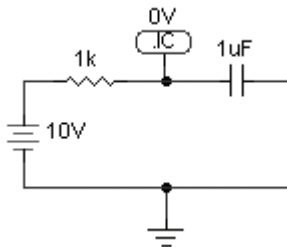
Now, let's replace one of the resistors with a capacitor to create a simple RC circuit where we can see the charging of the capacitor. Transient Analysis begins its simulation in a stable DC condition where the capacitors are already charged. Since we want to see the capacitor charging from time zero, we must set the initial condition of the capacitor to 0V:



9. Delete the second resistor (the one connected to ground) and replace it with a Capacitor [Passive Components/Capacitors] (c).

10. Select .IC device [Analog/SPICE Controls] (I) and connect it between the resistor and capacitor. This will set an initial condition of 0V on the capacitor for the analysis.

11. Run the simulation again. This time the Transient Analysis window (displays data like an oscilloscope) will be displayed. Click in the Transient Analysis window to select it, then click with the tip of the Probe Tool between the resistor and capacitor. You will see a diagonal line across the scope. This is actually the beginning of the charge curve for the capacitor. Our view of the curve is limited by start and stop times of the Transient Analysis that were selected by default. We now have the option of changing the Transient Analysis settings to increase the size of the time segment that we can view with the scope, or we can reduce the component values so the capacitor will charge quicker. For this example, we will change the component values.



12. Stop the simulation. Double-click on the Resistor to bring up the Edit Device Data dialog box. Change the Label-Value from 1k to 100. Click on the OK button to exit.

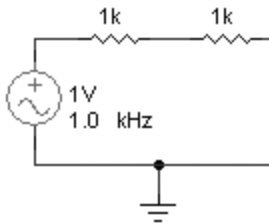
13. Double-click on the Capacitor. Change the Label-Value from 1uF to .001uF. Click on the OK button to exit.

14. Run the simulation again. This time you will see the charge curve of the capacitor.

Now let's create a simple AC circuit using a Signal Generator and 2 Resistors:



The New Button



15. Click on the New button in the Toolbar. An “Untitled” circuit window will be opened.
16. Draw the circuit as shown, using the following devices. Select 1 Signal Gen [Analog/Instruments] (g), 1 Ground [Analog/Power] (0 (zero)) and 2 Resistors [Passive Components/Resistors] (r). Use the Wire Tool to wire the circuit together.
17. Run the simulation. Click in the Transient Analysis window to select it, then click on the wire connected to the output of the Signal Generator. The sine wave will be displayed on the scope. Hold down the SHIFT key and click on the wire connected between the two resistors. A second waveform will be displayed on the scope.
18. Stop the simulation.

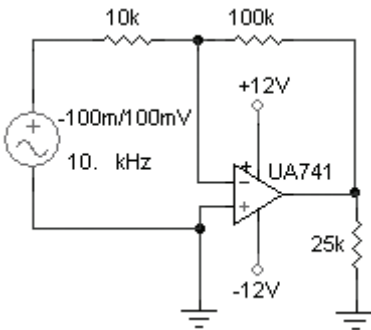
More Circuit Simulation

The next example will demonstrate how to use all of the analyses and how to make simple measurements using the cursors in the graphics windows. Let's create a simple 10X amplifier circuit using a $\mu\text{A}741$ Op Amp. In this configuration:

$$\text{voltage gain} = R_F/R_I$$

1. Select “New” from the File menu. An “Untitled” circuit window will be opened.
2. Make sure that Analog simulation mode is selected.
3. Disable the “Auto Designation” option in the Options menu (so there is no check mark by it). This allows us to specify our own designation for each device (e.g., Vcc, U1, RF, etc.)
4. Draw the circuit as shown (don't worry about the values), using the following devices. Select 1 Signal Gen [Analog/Instruments] (g) for Vin on the schematic. Select 2 +V devices [Analog/Power] (1) for Vcc and Vee. Select 2

Grounds [Analog/Power] (0 (zero)). Select 3 Resistors [Passive Components/Resistors] (r) for RI, RF and RL. Select 1 Op-Amp5 [Linear ICs/OPAMPS] for U1. Devices can be rotated in 90° increments by selecting the device with the mouse and clicking on the Rotate 90 button in the Toolbar. Use this method to rotate RL and the -12V supply. Use the Wire Tool to wire the circuit together. Use the Arrow Tool to drag the devices, wires and labels to make the circuit look nice.



5. Select the Arrow Tool from the Toolbar and double-click on the Op Amp. Select UA741 from the list of available subcircuits (it's near the bottom of the list) and click on the Select button. Now click on the Netlist button. Set the Designation field to "U1" and visible. Click on the OK button, then the Exit button.
6. Double-click on the TOP +V device. Set the Label-Value field to "+12V" and visible. Set the Designation field to "Vcc" and visible. Set the Device field to NOT visible. Click on the OK button.
7. Double-click on the BOTTOM +V device. Set the Label-Value field to "-12V" and visible. Set the Designation field to "Vee" and visible. Set the Device field to NOT visible. Click on the OK button. Click and drag the labels so they are positioned as shown on the schematic.
8. Double-click on each resistor to change both its Label-Value and its Designation and make them visible. Set them up as follows (refer to the circuit diagram):

<u>Resistor</u>	<u>Label-Value</u>	<u>Designation</u>
Input	10k	RI
Feedback	100k	RF
Load	25k	RL

9. Double-click on the Signal Generator. Set the Peak Amplitude to 0.1V and the frequency to 10kHz. Click on the Wave button. Enable the Source check box for AC Analysis. Set Magnitude to -0.1V and Phase to 0. Click on the OK button. The Signal Generator can now be used as a

reference for the AC analysis. Click on the Netlist button. Set the Designation field to “Vin” and visible. Note that the Label-Value field contains “-1/1V” which represents the minimum and maximum programmed voltage swings, respectively before you double-clicked on the Signal Generator. Click on the OK button. Again, click on the OK button to exit.

Setting up the Analyses

Once the circuit has been created, we will set up the analyses. When we run the simulation, the results will be based on the setup conditions provided here.

1. Select “Analog Analyses...” from the Options menu.
2. Click on the Always Set Defaults check box so it is cleared. This gives you access to the Transient and Operating Point analysis setups. When this box is checked, defaults are used every time you run a simulation.
3. Click on the Set Defaults button for default transient analysis setups. This will provide simulation for 5 cycles of the input signal with 200 data points. For best reliability, Max Step should be the same size as Step Time. More data points requires longer simulation time.
4. Select “DC” in the Operating Point section. This sets the initial display mode of the Value Window to DC. Note: Transient Analysis must be enabled in order to obtain DC AVG or AC RMS values. Operating Point must be enabled in order to use the Value Window.
5. Enable the DC Analysis. Set it up as follows:

	<u>Source Name</u>	<u>Start</u>	<u>Stop</u>	<u>Step</u>
Primary	Vin	-1.5V	-7V	0.01V
Secondary	Vcc	10V	14V	1V

This setup will allow us to sweep the voltage of Vin over the specified range at each of 5 different Vcc levels.

6. Enable the AC Analysis. Set it up as follows:

<u>Start Frequency</u>	<u>Stop Frequency</u>	<u>Test Points</u>	<u>Sweep</u>
1 Hz	1MegHz	100	Linear

7. Click on the OK button to save the settings.
8. Select “Save As...” from the File menu. Save the circuit as “MYAMP.CKT”.

Running the Simulation

When you run the simulation, an interactive SPICE simulation window will be displayed showing the progress of the simulation. If Run-Time Test Points have been placed in your circuit, you will be able to monitor the results as the data is collected. Otherwise, you will see only a bar graph showing the progress of the simulation. The amount of time it takes to finish is based on the analyses that are enabled and their setup values, the complexity of the circuit, and the speed of your computer.

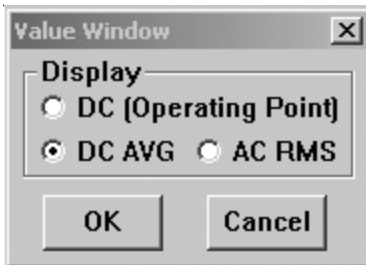
1. Select the Probe Tool in the Toolbar. Use the left mouse button, click on the wire connected to the output of the Op-Amp with the tip of the Probe Tool. A Run-Time Test Point will be placed on that node and a dialog box displayed. Enable the AC, DC and TRAN check boxes and change the Max. Scale of the DC graph to 15. Click on OK.
2. Click on the ON/OFF button in the Toolbar to start the simulation. The interactive SPICE simulation window will be displayed showing the waveforms as the data is collected. When the SPICE data collection process has completed, the analysis windows will be displayed.
3. Select the “Fit Circuit to Window” command (press F4) to make the entire circuit visible.
4. Click on the Value window to select it (it’s small black box in the upper right hand corner of the screen and should say “DC” in the title bar). Click on any wire in the circuit (except a wire connected to ground) with the tip of the Probe Tool. The DC voltage at that node will be displayed in the Value window. SPICE data is not collected for the Ground node in the circuit; it is always at zero volts.



5. Click on the “pin” of the +12V power supply (click above the pin dot, very close to the circle, otherwise you may get the wire instead of the pin). The DC current through that supply will be displayed in the Value window (if you clicked too close to the wire, the voltage for this node will be shown). Current and power can also be measured on other devices, but only if corresponding Test Points are enabled (see the Test Points section later in this chapter). Note: SPICE sees the current flowing *into* the positive terminal of a power supply, Multimeter or Signal Generator as positive current.

6. Double-click inside the Value window and change the setting to AC RMS. Click on the OK button. Now when you click on the wires in the circuit the AC voltage or current will be displayed.

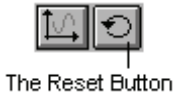
7. Click in the Transient Analysis window to select it, then click on the wire connected to the output of the Signal Generator with the tip of the Probe Tool. A green waveform will be displayed in the Transient Analysis window, similar to what would be seen on an oscilloscope.



8. Hold down the SHIFT key and click on wire connected to the output of the Op Amp. A second (yellow) waveform will be displayed in the Transient Analysis window. A quick comparison of the two waveforms will confirm that the amplitude at the output of the amplifier is much greater than the amplitude at the input.

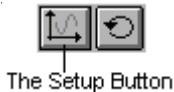
9. Click on the “c” cursor at the far right of the Transient Analysis window and drag it to the top peak of the output waveform (the yellow one). Click on the “d” cursor and drag it to the top peak of the input waveform (the green one). The actual peak voltages are displayed at top of the graph as “Yc” and “Yd”. As you can see from the “Yc” and “Yd” values, the peak voltage at the output of the amplifier is 10 times the peak voltage at the input of the amplifier. The difference between the two Y cursors is shown as “c-d”.

10. Click on the “b” cursor at the top of the Transient Analysis graph and drag it to the top peak of the first cycle of the output waveform. Click on the “a” cursor and drag it to the top peak of the second cycle of the output waveform. The period (period = 1/frequency) of the signal is shown as the difference between the two X cursors as “a-b”. The frequency is also displayed directly.
11. Click and drag a selection rectangle around a portion of the waveforms in the Transient Analysis window. The view will zoom in on the portion of the waveform selected. To restore the original view, click on the Reset button in the graph window.



12. Click on the DC Analysis window to select it, then click on any wire in the circuit. A DC analysis waveform will be displayed in the window, similar to what would be seen on a curve tracer. The cursors can be used to get measurements from the waveforms.

13. Click on the AC Analysis window to select it, then click on the wire at the output of the Op Amp. An AC analysis waveform will be displayed in the window. Click on the Setup button (the left button) in the upper left hand corner of the AC Analysis window. Select “Log” scale for the X Grid, select “Decibels” for the Y Axis and enable the “Show Wave Grid” check box. Click on the OK button. The waveform will now show the response of the circuit over the specified frequency. The cursors can be used to get measurements from the waveforms.

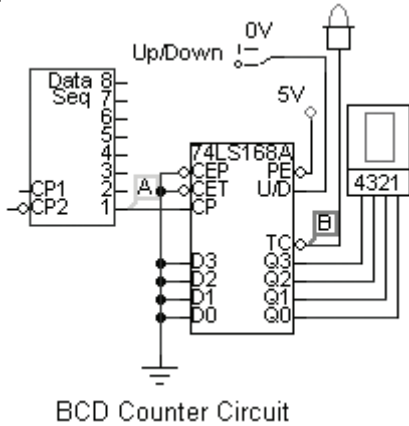


14. Switch the power to the circuit off by changing the toggle switch to the OFF position, which will stop the simulation and return to editing mode.

Mixed-Mode Simulation Example

The following BCD counter circuit demonstrates how digital simcode devices can be used in analog simulation mode.

1. Select “New” from the File menu.
2. Make sure that Analog simulation mode is selected.



3. Draw the circuit as shown using the following devices:

Data Sequencer	[Analog/Instruments]
74LS168A Counter	[Digital by Number/74xxx]
+V	[Analog/Power] (1)
Ground	[Analog/Power] (0 (zero))
Logic Switch	[Switches/Digital] (s)
Logic Display	[Displays/Digital] (9)
Hex Display	[Displays/Digital] (h).

4. Double-click on the +V and enter “DVCC;” in the Bus Data field to connect this device to the Vcc pin of the 74LS168A.

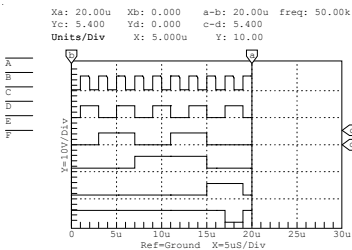
5. Double-click on the Data Sequencer. Click on the Pattern button. Select “Count Up” and click on OK. Enter “20” in the Stop Address field. Click on the OK button.

6. Select Analog Analyses from the Options menu. Make sure the “Always Set Defaults...” checkbox at the bottom of the dialog box is checked.

7. Select the Probe Tool from the Toolbar. Using the Left mouse button, click on the wire connected to the output of the Data Sequencer with the tip of the Probe Tool. A Run-Time Test Point will be placed on that node and a dialog box will be displayed. Enable the TRAN checkbox and the Combine Plots checkbox. Set the Max. Scale of the TRAN graph to 20. Click on the OK button.

8. Hold down the SHIFT key on the keyboard and click on the wire connected to the TC (terminal count) output of the 74LS168A. A second Run-Time Test Point will be placed on that node and a dialog box will be displayed. Again, enable the TRAN checkbox, and the Combine Plots checkbox. The Max. Scale should already be set to 20. Set the Vert. Offset to 6. Click on the OK button.

9. Click on the toggle switch to switch on power and start the simulation. Ignore any “pins not connected” warning. The interactive SPICE window will be displayed showing the waveforms as the data is being collected. When the SPICE data collection process has completed, the Transient Analysis window will be displayed.
10. Click on the output of the Data Sequencer to view the clock signal in the Transient Analysis window. Click on the Man button in the corner of the Transient Analysis window to switch to manual scaling mode. Click on the Up scale button to change the vertical scale to 10V/Div. Click on the green waveform label along the lefthand edge of the Transient Analysis window to select the waveform. Press the Up Arrow key on the keyboard briefly to change the vertical position of the waveform, moving the waveform near the top of the graph.
11. Hold down the SHIFT key and click on the Q0 output of the 74LS168A. A yellow waveform will appear near the center of the graph. Click on the corresponding yellow label along the lefthand edge of the Transient Analysis window to select this waveform. Press the Up Arrow key on the keyboard to change the vertical position of the waveform, moving it near the green waveform.
12. Repeat this procedure for each output. The resulting graph should appear similar to the one shown at the left.



This same circuit can be run in the Digital Logic simulation mode. To try it, stop the simulation, switch to digital and run simulation. In Digital Logic simulation mode, the displays will be animated to show the correct output.

Simulation Tools

Several buttons in the Toolbar are used specifically for simulation. These buttons are described below. It should be noted that the functionality of these tools is somewhat different for digital logic simulation.



The Simulation Tools

The Digital/Analog Button: 

Click on the Digital/Analog button to choose the type of simulation you want to run. When the AND gate icon is displayed, digital simulation is enabled; when the transistor icon is displayed, analog simulation is enabled.

The Reset Button: 

In Analog mode, clicking on the Reset button will generate the node numbers for the circuit without running the simulation. This would be important if you want to save a SPICE list to a file or view the node numbers on the schematic, but not run the simulation. Reset Simulation can also be selected from the Options menu or by pressing Ctrl+Q.

The ON/OFF Button: 

Click on the ON/OFF button in the upper right corner of the screen to start the simulation. An interactive SPICE simulation window will be displayed showing the progress of the simulation (see *SPICE For Windows* later in this chapter). The amount of time it takes to complete the simulation is based on the analyses that are enabled, their sweep ranges, the complexity of the circuit, and the speed of your computer. Finally, if no errors occurred during the simulation a graph window will be displayed for each of the selected analyses. Run Simulation can also be selected from the Options menu or by pressing F10.

If no changes have been made since the last simulation was run, clicking on the power toggle switch will not rerun the simulation, but will immediately load the previous simulation data.

Once the data collection sequence has ended and the analysis windows are displayed, pressing the OFF button will stop the simulation, closing all analysis windows, and returning to editing mode.

The Probe Tool: 

The Probe Tool can also be selected from the “Cursor Tools” option in the Options menu or by pressing Alt+P. It works differently in Analog mode than in Digital mode. If the simulation is not running, clicking on the circuit using the right mouse button will add or remove user defined Test Points. Clicking on the circuit using the left mouse button will add or remove Run-Time Test Points. See the Test Points and Run-Time Test Points sections below. If the simulation *is* running, clicking with the Probe Tool on a wire or device pin will display a value or waveform only in the active analysis window. SHIFT-click will allow you to select or deselect multiple items. CTRL-click will set the Probe Tool’s voltage reference node (the default voltage reference is ground) only for the active analysis window. The variable names shown in the selected analysis window are also shown on the circuit. Only the DC, DC AVG or AC RMS voltage or current of the last selected variable will be displayed in the Value window.

The Trace Button: 

The Trace button is used for digital simulation only. Refer to the *Simulation Tools* section of the *Digital Logic Simulation* chapter.

The Waveforms Button: 

Click the Waveforms button to open all of the analysis windows for which there is data in the .RAW file for this circuit. This allows you to view the graphs for this circuit without rerunning the simulation (this is only available if the circuit has not been modified since the .RAW file was created). If you click the button again, all the waveform analysis windows will close. Closing all the windows also stops the simulation (just like clicking on the Stop button) and allows you to edit the circuit. These graph windows can also be opened individually from the View menu. This command can also be selected in the Options menu or by pressing F12.

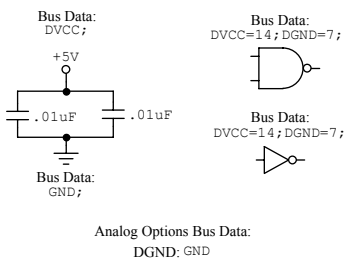
Vcc and Ground

CircuitLogix digital devices do not include Vcc and ground pins. However, to properly simulate these devices in analog simulation mode, Vcc and ground connections are required. This is done by placing entries in the Bus Data field of each digital device which reference specific power and ground buses.

A power supply bus can be created by placing a +V device in the circuit with a bus identifier (such as “DVCC;” or “DVDD;”) in its Bus Data field. A ground bus is already defined by the Ground symbol which contains “GND;” in its Bus Data field. If no bus is placed directly in the circuit, the default values for DVCC, DVDD and DGND are used. These default values are specified in the Analog Options dialog box. See *Analog Options* later in this chapter.

The Bus Data fields are predefined in CircuitLogix digital devices with values such as “DVCC=16;DGND=8;” which also identifies the specific pin numbers on the package that are connected to the buses.

In the example at the left, both the NAND gate and the inverter are connected to +5V and ground, along with two bypass capacitors. The DGND on the devices is connected to the GND bus through the Analog Options dialog box.



This is the same procedure used to connect the Vcc and ground pins in a PCB netlist for export. See *Bus Data* in the Schematic Capture chapter for more information.

Test Points

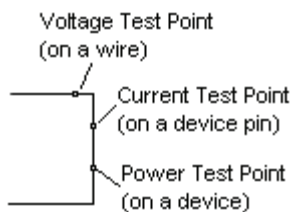
Test Points are locations where you can probe the circuit with the Probe Tool and expect to see data. Test Points determine how much data is actually stored in the .RAW file, and they determine which variables can be displayed in the analysis windows when the simulation is run. As a result, more test

points require a longer simulation time.

There are two methods of placing Test Points in a circuit:

First (and most common), CircuitLogix will automatically place test points at specific locations based on a user selected default. These default Test Points are selected using the radio buttons in the Analog Options dialog box. See the Analog Options section later in this chapter for a description of these radio buttons. The default Test Points are not displayed on the circuit.

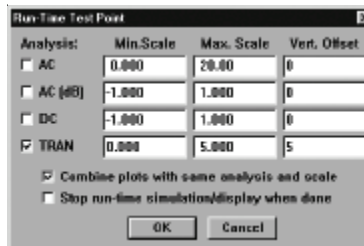
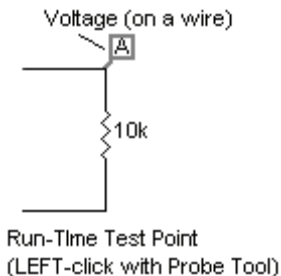
Second, user defined Test Points may be placed in the circuit. When user defined Test Points are in the circuit, the default Test Points described above become disabled and data will be collected only for each user defined Test Point. To place user defined Test Points in the circuit: 1) make sure the simulation is stopped; 2) select the Probe Tool from the Toolbar; and 3) click on the appropriate locations in the circuit using the right mouse button. To add multiple Test Points, hold to the SHIFT key while clicking. You can place Test Points on wires to measure node voltages, on device pins to measure current or on devices themselves to measure power dissipation. Data will not be collected for some devices such as those containing subcircuits. To remove all user defined Test Points from the circuit, right-click in any blank area of the circuit window with the Probe Tool.



Test Points (RIGHT-click with Probe Tool)

Run-Time Test Points

Run-Time Test Points are locations in the circuit where the data will be graphically displayed while it is being collected, that is, while the SPICE simulation program is running. Like regular Test Points, they can be used to measure voltage, current or power dissipation. To place Run-Time Test Points in the circuit: 1) make sure the simulation is stopped; 2) select the Probe Tool from the Toolbar; and 3) click on the appropriate locations in the circuit using the left mouse button. The following dialog box will be displayed:



You must specify which type of analysis data you wish to see: AC, AC in decibels, DC or Transient. You can optionally specify the minimum and maximum y-scale values that will be used on each graph. You can choose to have multiple waveforms displayed on the same graph during simulation. To do so, you must enable the “Combine plots with same analysis and scale” check box in each Run-Time Test Point that is to be combined. When combining Run-Time Test Points on a single plot, it is sometimes useful to change the DC offset of each waveform so they do not overlap. To do so, just enter a different offset value in the Vert. Offset field for each Run-Time Test Point. You can also choose to pause when the simulation is complete, before returning to CircuitLogix.

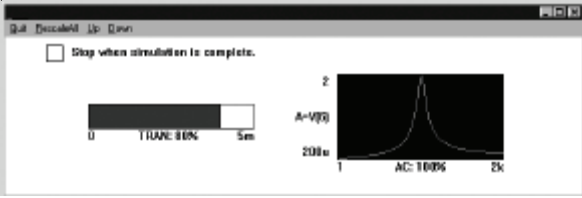
You can edit an existing Run-Time Test Point by double-clicking on it with the Arrow Tool.

To add multiple Run-Time Test Points, hold down the SHIFT key while clicking. You can place Run-Time Test Points on wires to measure node voltages, on device pins to measure current or on devices themselves to measure power dissipation. Waveforms will only be displayed if data is actually collected by either default or user defined Test Points. Waveforms will not be displayed for some devices such as those containing subcircuits. To remove all Run-Time Test Points from the circuit, left-click in any blank area of the circuit window with the Probe Tool.

SPICE for Windows

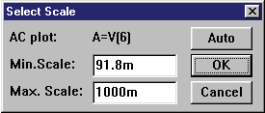
Switch the ON/OFF button to the ON position to start the simulation. An interactive SPICE simulation window will be

displayed showing the progress of the simulation. If there are Run-Time Test Points in the circuit, the corresponding waveforms will be displayed as the data is collected, otherwise the progress of the simulation will be displayed as a simple bar graph. Following is an example of a SPICE simulation window where both Transient and AC analyses are enabled, but only AC Analysis has a corresponding Run-Time Test Point.



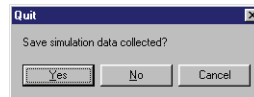
When activated, the “Stop when simulation is complete” check box tells SPICE to pause when the simulation is complete, before returning to CircuitLogix. This can also be set in the Run-Time Test Point dialog box. If enabled, SPICE will ask if you want to exit when it has completed the simulation. If you click on Yes, CircuitLogix will display the graphics windows allowing you to analyze the simulation data. If you click on No, you will remain in SPICE for Windows until you click on Quit in the menu bar.

If you click on Rescale All in the menu bar, all of the plots will be rescaled to fit the min. and max. data that has been collected. You can do this even while the simulation is still running. If you want to change the scale for a single plot window, click on the window to pause the simulation and the following dialog box will appear. Enter the desired scale limits (or click on the Auto button to scale to the data that is already in the window).



If there are more graphs to be displayed than can be shown on your monitor, use the Up and Down menu items to move the graphs into view.

The simulation can be stopped at any time. If you click on Quit in the menu bar, the following dialog box will be displayed, asking if you want to save the simulation data collected. This is very useful if you have been waiting for a long simulation and want to analyze the data which has been collected so far, but don't want to wait for the simulation to complete. If you click on Yes, CircuitLogix will display the graphics windows allowing you to analyze the simulation data that has been collected. *Note: the simulation cannot always be stopped at a location where the data will be available for display.* If you click on No, the simulation data will be discarded. If you click on Cancel, the simulation will continue.



A SPICE netlist is automatically generated and written to disk in a temporary file called "*filename*.NET" where *filename* is the name of the circuit file. The Berkeley SPICE3 program will analyze the netlist file and store the simulation results in a file called *filename*.RAW. This is all done while the interactive SPICE simulation window is being displayed. The amount of time it takes to complete the simulation is based on the analyses that are enabled, their sweep ranges, the complexity of the circuit, and the speed of your computer. The .RAW file can become very large for a complex simulation and is deleted when changes are made to the circuit. The size of the .RAW file can be greatly reduced by reducing the number of Test Points in the circuit and by disabling the ASCIIOUTPUT option in the Analog Options dialog box.

Warning Messages vs Error Messages

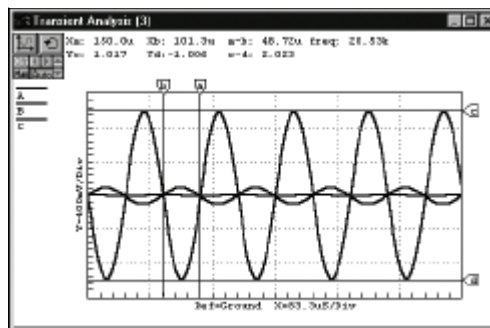
Sometimes CircuitLogix will display warning or error messages during circuit simulation. These messages will also be saved in a text file called *filename*.ERR. CircuitLogix will prompt you to view these messages, displaying them in the Windows Notepad editor.

Warning messages are not fatal to the simulation. They provide information about changes that SPICE had to make to the circuit in order to complete the simulation. These include invalid or missing parameters, etc. *Normally, valid simulation results are generated even if warning messages are reported.* Simcode warnings may include information such as timing violations (tsetup, thold, trec, tw, etc.) or significant drops in power supply voltage on digital components.

Error messages are fatal to the simulation and provide information about problems that SPICE could not resolve. Error messages indicate that simulation results could not be generated, and must be corrected before you will be able to analyze the circuit. Refer to Appendix A, *Troubleshooting SPICE Convergence*, if you need help with SPICE simulation errors.

The Graphics Windows

When a simulation is run, a separate window is displayed for each type of analysis that is enabled. The types of analyses available are Transient (oscilloscope), DC (curve tracer), AC (Bode plotter), and Operating Point (digital meter). See *Analog Analyses* later in this chapter. An example of the graph window in which analysis data is displayed is shown below.



Displaying Waveforms

To plot a variable in the graph window, click on a wire or supply pin with the tip of the Probe Tool. Clicking on another wire will replace the variable with a different one. Holding down the SHIFT key while clicking allows you to plot multiple variables simultaneously. Holding down the CTRL key while clicking allows you to move the voltage reference for the active graph window. To plot a subcircuit's internal variables, click on the subcircuit device with the tip of the Probe Tool

and select the variable from the list. Branch currents through and power dissipation of other devices can be displayed if the appropriate Test Points have been enabled. Branch currents can also be plotted for any branch in the circuit by inserting a Multimeter or 0V voltage source in the branch, then clicking on its pin. You can change the color of any waveform by double-clicking on its variable name in the variable list.

Scaling Waveforms

The graphic analysis windows can be used in either auto scale or manual scale mode. When the Man button is depressed, the window is set to manual scale mode. When the Auto button is depressed, the window is in auto scale mode.



The manual scale controls allow you to specify the units per division for both the X and Y scales. The units per division are displayed only for the selected waveform. A waveform is selected by clicking on its label—A, for example or V(8) if “Display Variable Names” is enabled in the Options menu—in the list on the left-hand side of the graph window. The up and down arrow buttons are used to change the Y scale for the selected waveform only. The left and right arrow buttons are used to change the X scale for all waveforms. Note: only linear scales can be adjusted, manual scaling has no effect on log scales.

In auto scale mode, all of the waveforms are displayed at the same scale. The graph is automatically scaled so that the waveforms fill almost the entire graph. If you click and drag a selection rectangle around a portion of the graph, that portion of the graph is zoomed to fill the graph window. Since the graph window itself can also be sized, it is easy to make any portion of a waveform fill almost the entire screen.

Offsetting Waveforms

Select a waveform by clicking on its name in the variable list on the left-hand side of the graph window. The waveform can then be repositioned on the graph, altering its x and y offsets. Click and drag the waveform itself with the mouse or press the arrow keys on the keyboard to change both the X and Y offsets. If the Off button is depressed, the offset values for the selected waveform are shown near the top of the graph window. If a cursor is selected, pressing an arrow key will move the cursor, not the waveform.

Tip: Waveforms and cursors can be nudged around on the graph by selecting one and pressing the arrow keys on the keyboard.

The Measurement Cursors

The measurement cursors are the heart of the waveform analysis windows. There are four cursors, two on the x axis and two on the y axis. Each cursor may be turned ON or OFF by double-clicking on the tab at the end of the cursor. A cursor may be repositioned either by dragging the tab or by selecting it and pressing the arrow keys on the keyboard. If no cursor is selected, pressing the arrow keys will move the selected waveform. The value at each cursor's position is displayed at the top of the graph window as well as the difference in value for each pair of cursors.

In manual scale mode, the cursor values correspond to the selected waveform only. In auto scale mode, the cursor values correspond to all waveforms, assuming there are no offsets.

The Setup Button:

The Setup button brings up the following dialog box which is used to control the x (horizontal) and y (vertical) axes of the graph and allows you to store and recall waveforms.



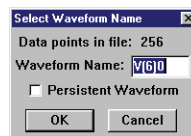
By default, the x axis represents time for the Transient Analysis and frequency for the AC Analysis. For DC it represents the sweep of the primary source. These defaults, however, can be changed to any variable name available in the pop-up menu. The Decibels and Phase options apply to AC Analysis only.

Any waveform which is currently being displayed on the graph can be stored for future reference. Select the waveform in the Store list box, then click on the Store button. A dialog box will appear, allowing you to specify the name of the file in which the waveform will be stored. Each waveform is stored in a separate file.

All waveforms displayed on a graph must have exactly the same datapoints on the X axis. Since SPICE does not always generate data at the same datapoints with each run, the data points must be linearized when they are stored. This guarantees that they can be recalled and placed on the same graph with any other waveform of the same analysis type. The number of datapoints to which the waveform will be linearized when it is stored is determined by the value entered in the Linear Data Points field. By default, this number is determined by the smallest step in datapoints before the waveform is linearized. The Show Data Points check box allows you to view the points at which there is data stored for the waveform.

Note: Anytime you linearize or alter the number of datapoints of a waveform, you may alter the shape of the waveform. Some data may be lost if you store or recall a waveform with too few datapoints. On the other hand, storing too many data points may cause the graph to be drawn slowly or to use up excessive memory.

To recall a stored waveform and place it on the graph, click on the Recall Stored Waveform button and select the appropriate waveform file from the file selection dialog box. The following dialog box is then displayed.



The number of linearized datapoints in the file is indicated in the dialog box. If the waveform is loaded, all SPICE data for the specified analysis type will be linearized to match the number of datapoints shown here.

The waveform may be renamed to avoid naming conflicts with other waveforms. If the Persistent Waveform check box is checked, the waveform will remain displayed on the graph no matter where else you click on the circuit. This type of waveform will have an asterisk (*) placed in front of the name.

If the Persistent Waveform check box is not checked, the waveform will only be displayed when you click on the corresponding point in the circuit (it will be displayed together with the waveform for the loaded circuit). Waveform names are considered to indicate the same point in the circuit if their names are the same up to the “)” or “]” character in the name. Multiple waveforms for the same point in the circuit may be plotted by adding different characters after the “)” or “]”. For example, if one recalled waveform is named v(4)1 and another is named v(4)2, both waveforms will be displayed simultaneously along with the current v(4) waveform when you click on node 4 in the circuit.

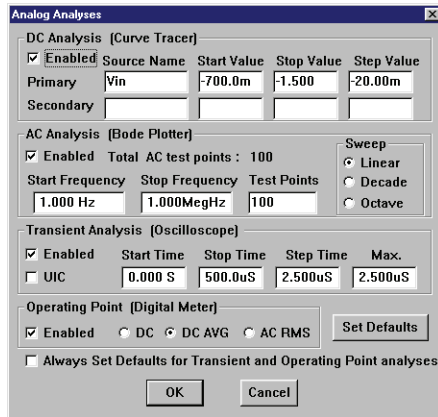
To remove a persistent stored waveform from the graph, select it in the Remove list box, then click on the Remove button. Alternately, to remove any waveform from a graph, select it (click on its label on the left side of the graph window) and press the DELETE key on the keyboard.

The Reset Button: 

Clicking on the Reset button in the graphics window returns all of the waveform offsets to zero and restores the graph to its unzoomed state. It does not change the axes.

Analog Analyses

Several types of analog analyses are available in CircuitLogix. Selecting “Analog Analyses...” from the Options menu brings up the following dialog box.



Operating Point Analysis

Operating Point Analysis generates data similar to the readings of a DC multimeter. It determines the DC bias of the entire circuit with inductors shorted and capacitors opened and determines linearized, small-signal models for all of the nonlinear devices in the circuit. It does not take into account the existence of any AC source. Operating Point Analysis is generally performed automatically before each of the other analyses, even if it has been disabled in the Analog Analyses dialog box. It must be enabled, however, if you want to use the Probe Tool as a multimeter to view the DC, DC AVG or AC RMS values of the circuit or to view OHMS or DC values with the multimeters. (Viewing the DC AVG or AC RMS values also requires that Transient Analysis be enabled.) See *The Probe Tool* earlier in this chapter.

DC Analysis

DC Analysis generates output like that of a curve tracer. It performs an Operating Point Analysis at each of a series of steps defining a DC transfer curve. Source Name is the name of an independent power source (either a fixed voltage or current supply or a Signal Generator) that is to be stepped in the circuit. The start, stop and step values define the sweep range and resolution. The primary source is required while the secondary source is optional. If a secondary source is specified, the primary source is stepped over its entire range for each value of the secondary source.

If DC Analysis is enabled, a Graphics Window will be displayed when the simulation is run, allowing you to view and measure the DC transfer curves for the circuit. In the example shown above, V_{in} will be swept from 0V to 3V in 100mV increments with V_{cc} at 3V, 4V and 5V. See *The Graphics Windows* earlier in this chapter.

AC Analysis

AC Analysis generates output like that of a Bode plotter. It computes the small-signal AC output variables as a function of frequency. It performs an Operating Point Analysis to determine the DC bias of the circuit, replaces the signal source with a fixed amplitude sine wave generator and analyzes the circuit over the user-specified frequency range. The desired output of an AC small-signal analysis is usually a transfer function (voltage gain, transimpedance, etc). In order to do an AC Analysis, at least one Signal Generator must be connected to the circuit and enabled as an AC Analysis source (double-click on the Signal Generator, then click on the Wave button to set up the AC Analysis Source options). When linear sweep is selected, Test Points is the total number of test points in the sweep. When decade is selected, Test Points is the number of test points per decade. When octave is selected, Test Points is the number of test points per octave.

If AC Analysis is enabled, a Graphics Window will be displayed when the simulation is run, allowing you to view and measure the frequency response of the circuit. In the example shown above, the selected AC sources will be swept from 10kHz to 10MHz in a linear sweep with a total of 100 test points. See *The Graphics Windows* earlier in this chapter.

Transient Analysis

Transient Analysis generates output like that of an oscilloscope. It computes the transient output variables (voltage or current) as a function of time over the user-specified time interval. It first performs an Operating Point Analysis to determine the DC bias of the circuit. Transient Analysis always begins at time zero. In the time interval between zero and Start Time, the circuit is analyzed but not displayed. In the time interval between Start Time and Stop Time, the circuit is both analyzed and displayed. Step Time is the suggested computing increment, but the timestep will be varied automati-

cally by SPICE in order to properly converge. Max. Step limits the varying size of the timestep that SPICE is allowed to use when calculating the transient data; for default, the program chooses either Step Time or (Stop Time - Start Time)/50, whichever is smaller. Typically, you will want to set Step Time and Max Step to be the same. When the UIC (Use Initial Conditions) option is enabled, the Transient Analysis begins from an initial condition, bypassing the Operating Point Analysis. This would be useful for viewing the charging of capacitors, etc.

If Transient Analysis is enabled, a Graphics Window will be displayed when the simulation is run, allowing you to view and measure voltage, current and power dissipation waveforms of the circuit, providing the appropriate Test Points have been enabled. In the example shown above, the transient analysis will be calculated from the Operating Point Analysis point (DC stable) to 2 μ s. See *The Graphics Windows* earlier in this chapter.

Set Defaults

The Set Defaults button is used to set up the default parameters of the Operating Point and Transient analyses. For simple DC circuits—those which contain no Signal Generators or reactive devices, for example—only Operating Point Analysis will be enabled. For more complex circuits, Transient Analysis will also be enabled and the default start, stop and step settings will be used. Start Time is set to zero and Stop Time, Step Time and Max Step are set to display 5 cycles of the *lowest* frequency Signal Generator in the circuit with a resolution of 200 data points. These default values are user definable in the Preferences dialog box. If the Always Set Defaults check box is checked, CircuitLogix will act as though this button was pressed automatically before each simulation.

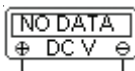
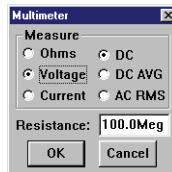
Always Set Defaults

This check box is intended to simplify the task of setting up the Transient and Operating Point analyses. It is primarily for the beginning user who does not completely understand or does not want to be concerned with all of the settings in the Analog Analyses dialog box. When this box is checked, the Set Defaults button will be pressed automatically every time the simulation is run.

Multimeter

CircuitLogix includes a multimeter for measuring resistance or DC, DC AVG or AC RMS voltage or current. You may place as many multimeters into your circuit as you like. When a simulation is run, the measured value will be displayed on the meter. When measuring voltage, connect the meter in parallel with the circuit; when measuring current, connect it in series. When measuring resistance, remove any power sources from the circuit and beware of dangling devices that may cause SPICE errors. Also, since the multimeter forces a current through the circuit to measure ohms, make sure you have only one multimeter set to ohms in the circuit at a time. Note: SPICE sees the current flowing into the positive terminal of a power supply, Multimeter or Signal Generator as positive current.

When you place a Multimeter [Analog/Instruments] in your circuit, the following dialog box appears showing the settings for the multimeter. Note: To measure DC AVG or AC RMS values, Transient Analysis must be enabled and must simulate enough cycles of transient data to make the measurements meaningful. Likewise, Operating Point Analysis must be enabled in order to obtain resistance and DC values.



The resistance of a voltmeter is high and the resistance of an ammeter is low, so it will have little effect on the circuit it is measuring. When measuring voltage on a high resistance circuit it may be desirable to increase the resistance of the voltmeter. When measuring current through a low resistance circuit it may be desirable to decrease the resistance of the ammeter. When measuring pn junction resistance, it may be desirable to increase the forcing current of the ohmmeter.

Multifunction Signal Generator

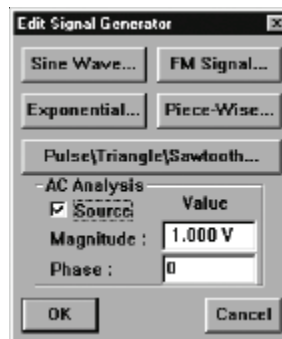
CircuitLogix includes multifunction signal generators, and you can place as many signal generators in your circuit as you would like. Waveform functions include Sinusoidal, Single-Frequency FM, Exponential, Pulse (including Triangle and Sawtooth), and Piece-Wise Linear. The settings for each of these functions are edited through separate dialog boxes. These dialog boxes are described below.

After placing a Signal Gen [Analog/Instruments] (g) in your circuit, double-click on it with the Arrow Tool. A dialog box will appear showing the settings for the currently selected waveform function (the default is Sine Wave). All of the waveform dialog boxes have the following items in common:

Volts/Amps: Allows you to select whether this is a voltage or current source.

Netlist...: Brings up the Edit Device Data dialog box described in the *Schematic Capture* chapter.

Wave...: Brings up the following dialog box, allowing you to change waveform functions for this generator.

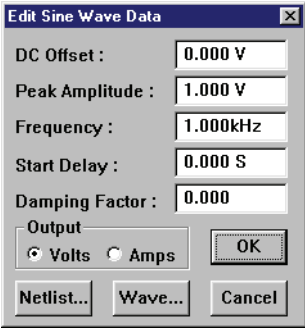
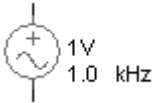


This dialog box allows you to change waveform functions by clicking on one of the function buttons and also allows you to specify this generator's effect when running AC Analysis. AC Analysis temporarily replaces each AC source with a sinusoidal wave of a fixed magnitude and phase. The Source

check box allows you to specify whether this generator will be used as an AC source in the AC Analysis. The Magnitude and Phase edit fields allow you to specify the fixed values that will be used.

Sine Wave...

The following dialog box is used to set the parameters of the sinusoidal waveform.



The waveform, beginning at Start Delay, is described by the following formula where t = instance of time:

$$V(t_0 \text{ to } t_{SD}) = VO$$

$$V(t_{SD} \text{ to } t_{STOP}) = VO + VA \sin(2\pi F (t-SD)) e^{-(t-SD)THETA}$$

DC Offset (VO): Used to adjust the DC bias of the signal generator with respect to the negative terminal (usually ground), measured in volts or amps.

Peak Amplitude (VA): Maximum amplitude of the output swing, excluding the DC Offset, measured in volts or amps.

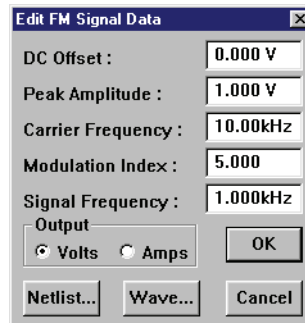
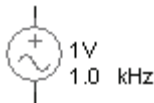
Frequency (F): Frequency of the output in hertz.

Start Delay (SD): Provides a phase shift of the output by delaying the start of the sine wave.

Damping Factor (THETA): A positive value results in an exponentially decreasing amplitude; a negative value results in an exponentially increasing amplitude.

FM Signal...

The following dialog box is used to set the parameters of the single-frequency FM waveform.



The waveform is described by the following formula where t = instance of time:

$$V(t) = V_O + V_A \sin(2\pi F_c t + \text{MDI} \sin(2\pi F_s t))$$

DC Offset (V_O): Used to adjust the DC bias of the signal generator with respect to the negative terminal (usually ground), measured in volts or amps.

Peak Amplitude (V_A): Maximum amplitude of the output swing, excluding the DC Offset, measured in volts or amps.

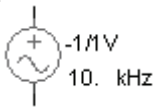
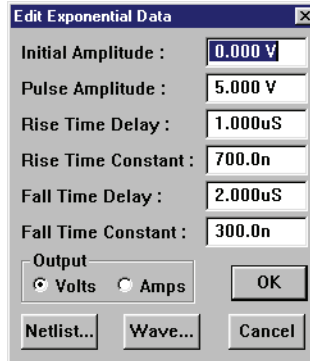
Carrier Frequency (F_c): Frequency of unmodulated output in Hz.

Modulation Index (MDI): Value corresponding to a function of amplitude of the modulating signal indicating the level of modulation. $\text{MDI} = (\text{frequency deviation}) / F_s$

Signal Frequency (F_s): Frequency of modulating signal in Hz.

Exponential...

The following dialog box is used to set the parameters of the exponential waveform.



The waveform is described by the following formulas where t = instance of time:

$$\begin{aligned}
 V(t_0 \text{ to } t_{RD}) &= VI \\
 V(t_{RD} \text{ to } t_{FD}) &= VI + (VP - VI) (1 - e^{-(t-t_{RD})/t_{RT}}) \\
 V(t_{FD} \text{ to } t_{STOP}) &= VI + (VP - VI) (-e^{-(t-t_{RD})/t_{RT}}) + \\
 &\quad (VI - VP) (1 - e^{-(t-t_{FD})/t_{FT}})
 \end{aligned}$$

Initial Amplitude (VI): Initial amplitude of the output with respect to the negative terminal (usually ground), measured in volts or amps.

Pulse Amplitude (VP): Maximum amplitude of the output swing, measured in volts or amps.

Rise Time Delay (RD): The point in time, from t_0 , when the output begins to rise. This provides a phase shift of the output by delaying the start of the exponential waveform.

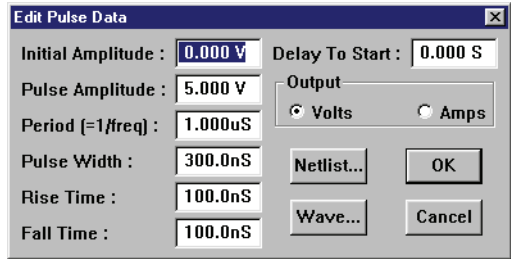
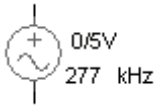
Rise Time Constant (RT): RC charging time constant.

Fall Time Delay (FD): The point in time, from t_0 , when the output begins to fall.

Fall Time Constant (FT): RC discharging time constant.

Pulse\Triangle\Sawtooth...

The following dialog box is used to set the parameters of the pulse waveform.



The waveform is described as follows where t = instance of time. Intermediate points are set by linear interpolation:

$$\begin{aligned} V(t_0) &= V_I \\ V(t_{SD}) &= V_I \\ V(t_{SD} + t_{TR}) &= V_P \\ V(t_{SD} + t_{TR} + t_{PW}) &= V_P \\ V(t_{SD} + t_{TR} + t_{PW} + t_{TF}) &= V_I \\ V(t_{STOP}) &= V_I \end{aligned}$$

Initial Amplitude (V_I): Initial amplitude of the output with respect to the negative terminal, measured in volts or amps.

Pulse Amplitude (V_P): Maximum amplitude of the output swing, measured in volts or amps.

Period ($=1/\text{freq}$): Duration of one complete cycle of the output.

Pulse Width (PW): Duration that output remains at V_P before beginning to ramp toward V_I .

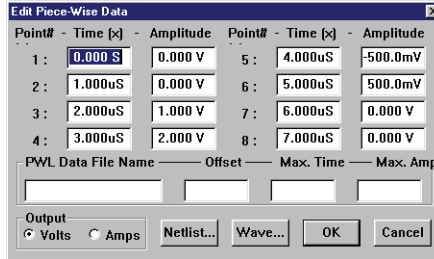
Rise Time (TR): Duration of the ramp from V_I to V_P .

Fall Time (TF): Duration of the ramp from V_P to V_I .

Delay to Start (SD): Duration that the output remains at V_I before beginning to ramp toward V_P the first time.

Piece-Wise...

The following dialog box is used to set the parameters of the piece-wise linear waveform.



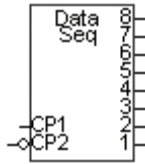
Piece-wise linear data must come from one of two sources: 1) The waveform may be described by a set of up to 8 points entered directly into this dialog box by the user. The time specified for each successive point must be more positive than its predecessor. If it is not, the cycle will end, excluding that and all successive points. 2) Optionally, the waveform may be defined in an ASCII text file containing an indefinite number of points. Values must be entered in pairs and each pair must include a time position followed by an amplitude. The first character of each data line must be a plus sign (+) and each line may contain up to a maximum of 255 characters. Values must be separated by one or more spaces or tabs. Values may be entered in either scientific or engineering notation. Comments may be added to the file by making the first character of the line an asterisk (*). For example:

```
* Random Noise Data
+ 0.00000e-3 0.6667 0.00781e-3 0.6372 0.01563e-3 -0.1177
+ 0.02344e-3 -0.6058 0.03125e-3 0.2386 0.03906e-3 -1.1258
+ 0.04688e-3 1.6164 0.05469e-3 -0.3136 0.06250e-3 -1.0934
+ 0.07031e-3 -0.1087 0.07813e-3 -0.1990 0.08594e-3 -1.1168
+ 0.09375e-3 1.4890 0.10156e-3 -0.2169 0.10938e-3 -1.4915
+ 0.11719e-3 1.4914 0.12500e-3 0.1486
```

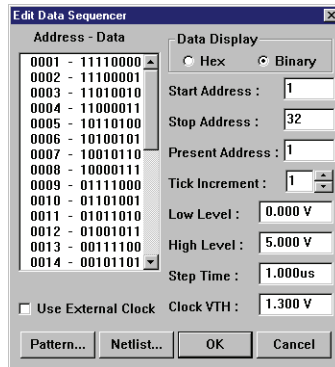
Intermediate points are determined by linear interpolation. If Max Amplitude is specified in the dialog box, the data in the file will be scaled vertically so that the peak-to-peak amplitude of the waveform is equal to the specified Max Amp and centered around zero. If an Offset is specified, the waveform will be offset vertically by the specified amount. If a Max Time is specified, the waveform will be scaled horizontally to fit the specified Max Time. Example circuit: PWL.CKT.

Note: *The .PWL file must be located in the same directory as the circuit you are simulating.*

The Data Sequencer



This device can be used in both digital and analog simulation modes. Also known as a Data or Word Generator, it allows the user to specify up to 32767 8-bit words which can be output in a defined sequence. Since there is no limit to the number of Data Sequencers that can be placed in a circuit, several could be placed in parallel to create a data stream of any width. Double-click on the Data Sequencer with the Arrow Tool to display the following dialog box. For a complete description of the data sequencer, refer to The Data Sequencer in the *Digital Logic Simulation* chapter.



Initial Conditions

.NODESET Statement

General Form:

```
.NODESET V(NODNUM)=VAL V(NODNUM)=VAL ...
```

Netlist Example:

```
.NODESET V(7)=3.33 V(11)=1.5
```

The Nodeset line helps the program find the dc or initial transient solution by making a preliminary pass with the specified nodes held to the given voltages. The restriction is then released and the iteration continues to the true solution. The .NODESET line may be necessary for convergence on bistable or astable circuits. In general, this line should not be necessary. See also: .NODESET.

.IC Statement

General Form:

```
.IC V(NODNUM)=VAL V(NODNUM)=VAL ...
```

Netlist Example:

```
.IC V(3)=2.5 V(4)=-1 V(9)=1
```

The IC line is for setting transient initial conditions. It has two different interpretations, depending on whether or not the UIC parameter is enabled in the Transient Analysis. Also, one should not confuse this line with the .NODESET line. The .NODESET line is only to help dc convergence, and does not affect final bias solution (except for multi-stable circuits). The two interpretations of this line are as follows:

1. When the UIC parameter is enabled in the Transient Analysis, the node voltages specified on the .IC control line are used to compute the capacitor, diode, BJT, JFET, and MOSFET initial conditions. This is equivalent to specifying the IC=... parameter on each device line, but is much more convenient. The IC=... parameter can still be specified and takes precedence over the .IC values. Since no dc bias (initial transient) solution is computed before the transient analysis, one should take care to specify all dc source voltages on the .IC control line if they are to be used to compute device initial conditions.

2. When the UIC parameter is not enabled in the Transient Analysis, the dc bias (initial transient) solution is computed before the transient analysis. In this case, the node voltages specified on the .IC control line are forced to the desired values. During transient analysis, the constraint on these node voltages is removed. This is the preferred method since it allows SPICE to compute a consistent dc solution.

See also: .IC. Example circuit: 555.CKT.

SPICE's Elementary Devices

Each device used in a circuit requires certain information for the SPICE simulation to run. This section describes each type of device and information that each requires. CircuitLogix provides this information to the SPICE netlist if the circuit is properly drawn and labeled by the user.

This information is provided here to help the user better understand how each of the components affects the simulation of the circuit. Parameters that are enclosed in “< >” symbols are optional. Refer to the *SPICE Models* section of the *Macro Devices* chapter for more information about the models mentioned here.

Resistors

General Form:

```
RXXXXXXXX N1 N2 VALUE
```

Netlist Example:

```
R1 1 2 10K
```

Spice Data Example:

```
%D %1 %2 %V
```

N1 and N2 are the two element nodes. Value is the resistance (in ohms) and may be positive or negative, but not zero. See also: Resistor, Var Resistor. Example circuit: ANALOG.CKT.

Semiconductor Resistors

General Form:

```
RXXXXXXXX N1 N2 <VALUE> <MNAME> <L=LENGTH> <W=WIDTH> <TEMP=T>
```

Netlist Examples:

```
R1 1 2 10K
```

```
R3 5 6 RMODEL L=12u W=1u
```

Spice Data Example:

```
%D %1 %2 %M TEMP=100
```

This is the more general form of the resistor. It allows modeling of temperature effects, and calculation of the actual resistance value from strictly geometric information and the specifications of the process. If VALUE is specified, it overrides the geometric information and defines the resistance. If MNAME is specified, then the resistance may be calculated from the process information in the model MNAME and the given

LENGTH and WIDTH. If VALUE is not specified, then MNAME and LENGTH **must** be specified. If Width is not specified, then it is taken from the default width given in the model. The (optional) TEMP value is the temperature at which this device is to operate, and overrides the temperature specification in the Analog Options dialog box.

Capacitors

General Form:

```
CXXXXXXXX N+ N- VALUE <IC=INCOND>
```

Netlist Example:

```
C2 13 0 0.1UF
C5 7 0 10UF IC=3V
```

Spice Data Example:

```
%D %1 %2 %V IC=12V
```

N+ and N- are the positive and negative element nodes, respectively. VALUE is the capacitance in Farads. The initial condition is the initial (time-zero) value of the capacitor voltage. Initial conditions only apply if the UIC option is enabled for the Transient Analysis.

SPICE uses perfect capacitors, that is, capacitors with no DC leakage. Since all nodes in a circuit must have a DC path to ground, you cannot simulate a circuit with capacitors in series as this would completely isolate the nodes that are between the capacitors. One solution to this problem is to connect a large-value resistor, say 1-gigaohm, in parallel with each capacitor to account for the leakage resistance of that capacitor.

See also: Capacitor, Polar Cap, Var Capacitor. Example circuit: 555.CKT.

Semiconductor Capacitors

General Form:

```
CXXXXXXXX N+ N- <VALUE> <MNAME> <L=LENGTH> <W=WIDTH> <IC=INCOND>
```

Netlist Example:

```
C2 13 0 0.1UF
C5 7 0 10UF CMODEL L=12u W=1u
```

Spice Data Example:

```
%D %1 %2 %M L=10U
```

This is the more general form of the Capacitor and allows for the calculation of the actual capacitance value from strictly geometric information and the specifications of the process. If VALUE is specified, it defines the capacitance. If MNAME is specified, then the capacitance is calculated from the process information in the model MNAME and the given LENGTH and WIDTH. If VALUE is not specified, then MNAME and LENGTH **must** be specified. If WIDTH is not specified, then it is taken from the default width given in the model. You may specify either VALUE **or** MNAME, LENGTH, and WIDTH, but not both.

Inductors

General Form:

```
LYYYYYYY N+ N- VALUE <IC=INCOND>
```

Netlist Example:

```
L3 12 9 1UH  
L4 5 0 100UH IC=12.3MA
```

Spice Data Example:

```
%D %1 %2 %V IC=5MA
```

N+ and N- are the positive and negative element nodes, respectively. VALUE is the inductance in Henries. The initial condition is the initial (time-zero) value of the inductor current. Initial conditions only apply if the UIC option is enabled for the Transient Analysis.

Since inductors induce voltage across their coils based on the amount of magnetic flux, they must be considered as voltage sources in SPICE. In SPICE, voltage sources (and therefore inductors) cannot be connected directly in parallel. One solution to this problem is to connect a small-value resistor, say 0.001 ohms, in series with each inductor to account for the winding resistance of that inductor.

See also: Inductor, Var Inductor, Coil 3T, Coil 5T. Example circuit: RESONANT.CKT.

Coupled (Mutual) Inductors

General Form:

```
KXXXXXXXX LYYYYYYY LZZZZZZZ VALUE
```

Netlist Examples:

```
K12 L4 L3 0.999
KXFRMR L1 L2 0.87
```

Spice Data Example: (center-tap inductor)

```
%DA %1 %2 50UH (inductor A)
%DB %2 %3 50UH (inductor B)
K%D %DA %DB .85 (inductive coupling)
```

LYYYYYYY and LZZZZZZZ are the names of two coupled inductors, and VALUE is the coefficient of coupling, K, which must be greater than 0 and less than or equal to 1. Using the “dot” convention, place a “dot” on the first node of each inductor, indicating that the voltages at these node are in phase (the voltages go up and down together).

If more than two inductors are being coupled, SPICE data must be provided for each coupling. For example, a transformer with one primary coil (L1) and two secondary coils (L2 and L3) might be expressed as follows:

```
L1 5 0 10MH
L2 6 7 1MH
L3 8 9 1MH
K12 L1 L2 0.93
K13 L1 L3 0.93
K23 L2 L3 0.97
```

The turns ratio for a given pair of windings can be determined by the following formula where LS and LP are the inductance of the primary and secondary windings, respectively:

$$\text{Turns Ratio} = \sqrt{L_S/L_P}$$

See also: Transformers. Example circuit: VTPWRAMP.CKT.

Voltage/Current Controlled Switches

General Form:

```
SXXXXXXX N+ N- NC+ NC- MODEL <ON><OFF>
WXXXXXXX N+ N- VNAME MODEL <ON><OFF>
```

Netlist Examples:

```
S1 1 2 3 4 SVS1 (V->Switch)
S2 5 6 3 0 SVS2 ON (V->Switch)
W1 1 2 VS1 WIS1 (I->Switch)
```

Spice Data Example:

```
%D %1 %2 %3 %4 %M ON
```

Nodes 1 and 2 are the nodes between which the switch terminals are connected. ON/OFF indicates the initial condition of the switch. For the voltage controlled switch nodes, 3 and 4 are the positive and negative controlling nodes, respectively. For the current controlled switch, the controlling current is that through the specified voltage source. The direction of positive controlling current flow is from the positive node, through the source, to the negative node. See also: I->Switch, V->Switch. Example circuit: SWITCHES.CKT.

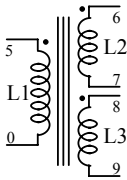
Independent Sources

General Form:

```
VXXXXXXX N+ N- <<DC> VALUE> <AC <MAG <PHASE>>>
IYYYYYYY N+ N- <<DC> VALUE> <AC <MAG <PHASE>>>
```

Netlist Examples:

```
VCC 10 0 DC 6 (V Source)
ISRC 1 2 AC .3 45 SIN(0 1 1MEG) (Signal Gen)
VMEAS 12 9 (Ammeter)
```



Note: A transformer that is simulated in this manner will not reflect the impedance of the secondary winding back into the primary.

Spice Data Example:

```
%D %1 %2 DC 0 SIN(0 1 1k 0 0) AC 1 0 (Signal Gen)
```

N+ and N- are the positive and negative nodes, respectively. Voltage sources need not be grounded. Positive current is assumed to flow into the positive node, through the source, and out of the negative node. A current source of positive value forces current to flow into the N+ node, through the source, and out of the N- node. Voltage sources, in addition to being used for circuit excitation, are the ‘ammeters’ for SPICE; that is, zero valued voltage sources may be inserted into the circuit for the purpose of measuring current. They, of course, have no effect on circuit operation since they represent short-circuits.

VALUE is the DC (operating point) value or offset of the source. If the source value is zero it may be omitted. If the source is time-invariant (e.g., a power supply), then the value can be preceded by the letters DC.

The letters AC indicated a small-signal AC source. MAG (AC magnitude) and PHASE (AC phase) are used for AC analysis only. If MAG is omitted following the keyword AC, a value of 1 is assumed. If PHASE is omitted, a value of 0 is assumed.

Any independent source can be assigned a time-dependent value for transient analysis. If a source is assigned a time-dependent value, the time-zero value is used for DC (operating point) analysis. There are five independent source functions: pulse, exponential, sinusoidal, piece-wise linear, and single-frequency FM. These are discussed in the *Multifunction Signal Generator* section of this chapter.

Note: For SPICE simulation, voltage sources cannot be placed directly in parallel, and current sources cannot be placed directly in series.

See also: +V, V Source, I Source, Battery, Signal Gen. Example circuit: CEAMP.CKT.

Linear Voltage-Controlled Current Sources

General Form:

```
GXXXXXXX N+ N- NC+ NC- VALUE
```

Netlist Example:

```
G1 2 0 5 0 0.1MMHO
```

Spice Data Example:

```
%D %1 %2 %3 %4 %V
```

N+ and N- are the positive and negative nodes, respectively. Current flow is from the positive node, through the source, to the negative node. NC+ and NC- are the positive and negative controlling nodes, respectively. VALUE is the transconductance (in mhos). See also: V->I Source. Example circuit: 741.CKT.

Note: For SPICE simulation, current sources cannot be placed directly in series.

Linear Voltage-Controlled Voltage Sources

General Form:

```
EXXXXXXX N+ N- NC+ NC- VALUE
```

Netlist Example:

```
E1 2 3 14 1 2.0
```

Spice Data Example:

```
%D %1 %2 %3 %4 %V
```

N+ is the positive node and N- is the negative node. Current flow is from the positive node, through the source, to the negative node. VALUE is voltage gain. See also: V->V Source.

Note: For SPICE simulation, voltage sources cannot be placed directly in parallel.

Linear Current-Controlled Current Sources

General Form:

```
FXXXXXXX N+ N- VNAME VALUE
```

Netlist Example:

```
F1 5 17 VS2 0.5K
```

Spice Data Example:

```
V%D %3 %4 DC 0V (measures controlling current)  
%D %1 %2 V%D %V (current source)
```

N+ and N- are the positive and negative nodes, respectively. Current flow is from the positive node, through the source, to the negative node. VNAME is the name of a voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of VNAME. VALUE is the current gain. See also: I->I Source.

Note: For SPICE simulation, current sources cannot be placed directly in series.

Linear Current-Controlled Voltage Sources

General Form:

```
HXXXXXXX N+ N- VNAME VALUE
```

Netlist Example:

```
H1 5 17 VS2 0.5K
```

Spice Data Example:

```
V%D %3 %4 DC 0V      (measures controlling current)
%D %1 %2 V%D %V      (voltage source)
```

N+ and N- are the positive and negative nodes, respectively. VNAME is the name of a voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of VNAME. VALUE is the transresistance (in ohms). See also: I->V Source. Example circuit: 741.CKT.

Note: For SPICE simulation, voltage sources cannot be placed directly in parallel.

Nonlinear Dependent Sources

General Form:

```
BXXXXXXX N+ N- <I=EXPR> <V=EXPR>
```

Netlist Examples:

```
BSO1 0 1 I=cos(V(1))+sin(V(2))
BVS1 0 1 V=LN(COS(LOG(V(1,2)^2)))-
V(3)^4+V(2)^V(1)
```

Spice Data Example:

```
%D %1 %2 V=%L
```

N+ is the positive node, N- is the negative node. The values of the V and I parameters determine the voltages and currents across and through the device, respectively. If I is given then the device is a current source, and if V is given the device is a voltage source. One and only one of these parameters must be specified for each source.

The expressions given for V and I may be any function of voltages and currents through voltage sources in the system (e.g., V(1) indicates the DC voltage at node 2 referenced to ground, V(3,4) indicates the voltage difference between nodes 3 and 4, and I(VS2) indicates the DC current through the voltage supply VS2). The following functions of real variables are defined:

abs	acos	acosh	asin	asinh	atan
atanh	cos	cosh	exp	ln	log
sin	sinh	sqrt	tan	u	uramp

“u” is the unit step function, with a value of one for arguments greater than one and a value of zero for arguments less than zero. “uramp” is the integral of the unit step: for an input x , the value is zero if x is less than zero, or if x is greater than zero the value is x . These two functions are useful in synthesizing piecewise nonlinear functions, though convergence may be adversely affected.

The following standard operators are defined:

+ - * / ^ unary -

In addition, the following boolean operators are defined. Input threshold values (BOOLT) and output values (BOOLL and BOOLH) are universally defined in the Analog Options dialog box.

& (AND) | (OR) ! (XOR) ~ (NOT)

Older versions of SPICE used a POLY function to describe nonlinear sources. For example, the following statements are all equivalent:

```
E1 19 0 POLY(2) 7 4 2 0 3 .1 .5      (Spice2)
E1 19 0 POLY(2) (7,4) (2,0) 3 .1 .5 (Spice2)
B1 19 0 V = 3 + .1*V(7,4) + .5*V(2,0)
(Spice3)
E1 19 0 V = 3 + .1*V(7,4) + .5*V(2) (Spice3)
```

Each statement indicates that the voltage at node 19 will equal 3 volts plus .1 times the voltage across nodes 7 and 4 plus .5 times the voltage at node 2, using ground (node 0) as a reference. Many existing SPICE subcircuits contain this type of nonlinear source. CircuitLogix automatically converts them to the SPICE3 format each time you run a simulation.

Note: For SPICE simulation, voltage sources cannot be placed directly in parallel and current sources cannot be placed directly in series.

See also: NLV Source, NLI Source, I-Math, V-Math. Example circuit: 741.CKT.

Lossless Transmission Lines

General Form:

```
XXXXXXXX N1 N2 N3 N4 Z0=VALUE <TD=VALUE>  
+ <F=FREQ <NL=NRMLEN>> <IC=V1, I1, V2, I2>
```

Netlist Example:

```
T1 3 0 2 0 Z0=50 TD=20NS
```

Spice Data Example:

```
%D %1 %2 %3 %4 Z0=%V TD=10NS
```

N1 and N2 are the nodes at port1; N3 and N4 are the nodes at port 2. Z0 is the characteristic impedance. The length of the line may be expressed in either of two forms (one form must be specified). The transmission delay, TD, may be specified directly (as TD=10NS, for example). Alternately, a frequency F may be given, together with NL (the normalized electrical length of the transmission line with respect to the wavelength in the line at the frequency F). If a frequency is specified but NL is omitted, 0.25 is assumed (that is, the frequency is assumed to be the quarter-wave frequency).

The initial condition specification consists of the voltage and current at each of the transmission line ports. Initial conditions only apply if the UIC option is enabled for the Transient Analysis.

The lossy transmission line described below with zero loss may be more accurate than the lossless transmission line due to implementation details.

Lossy Transmission Lines

General Form:

```
OXXXXXXXX N1 N2 N3 N4 MNAME
```

Netlist Example:

```
O2 3 0 2 0 OXLINE
```

Spice Data Example:

```
%D %1 %2 %3 %4 %M
```

This is a two-port convolution model for single-conductor lossy transmission lines. N1 and N2 are the nodes at port1; N3 and N4 are the nodes at port 2. Note that a lossy transmission line with zero loss may be more accurate than the lossless transmission line due to implementation details.

Uniform Distributed RC Lines (Lossy)

General Form:

```
UXXXXXXX N1 N2 N3 MNAME L=LEN <N=LUMPS>
```

Netlist Example:

```
U1 1 2 0 UXLINE L=50UM N=6
```

Spice Data Example:

```
%D %1 %2 %3 %M L=25u
```

N1 and N2 are the two element nodes the RC line connects, while N3 is the node to which the capacitances are connected. MNAME is the model name, LEN is the length of the RC line in meters. LUMPS, if specified, is the number of lumped segments to use in modeling the RC line (if omitted, a default value based on the model parameters will be used). See also: URC-Line.

Junction Diodes

General Form:

```
DXXXXXXX N+ N- MNAME <AREA> <OFF> <IC=VD>  
<TEMP=T>
```

Netlist Example:

```
D3 2 10 D1N914 OFF  
D5 7 12 D1N4001 3.0 IC=0.2
```

Spice Data Example:

```
%D %1 %2 %M OFF IC=.6 TEMP=70
```

N+ and N- are the positive and negative nodes, respectively. MNAME is the model name, AREA is the area factor, and OFF indicates an optional starting condition on the device for operating point analysis. The initial condition specification using IC=VD only applies if the UIC option is enabled for the Transient Analysis. The TEMP value is the temperature at which this device is to operate, and overrides the temperature specification in the Analog Options dialog. See also: Diode, Zener Diode. Example circuit: PS1.CKT.

Bipolar Junction Transistors (BJTs)

General Form:

```
QXXXXXXX NC NB NE <NS> MNAME <AREA> <OFF> <IC=VBE, VCE>  
+ <TEMP=T>
```

Netlist Example:

```
Q5 11 26 4 Q2N3904 IC=0.6, 5.0
Q3 5 2 6 9 QNPN .67
```

Spice Data Example:

```
%D %1 %2 %3 %M .67 OFF
```

NC, NB and NE are the collector, base and emitter nodes, respectively. NS is the optional substrate node; if unspecified, ground is used. MNAME is the model name, AREA is the area factor, and OFF indicates an optional starting condition on the device for operating point analysis. The initial condition specification using IC=VBE, VCE only applies if the UIC option is enabled for the Transient Analysis. The TEMP value is the temperature at which this device is to operate, and overrides the temperature specification in the Analog Options dialog. See also: NPN Trans, PNP Trans. Example circuit: CEAMP.CKT.

Junction Field-Effect Transistors (JFETs)

General Form:

```
JXXXXXXX ND NG NS MNAME <AREA> <OFF> <IC=VDS, VGS> <TEMP=T>
```

Netlist Example:

```
J2 6 3 21 J2N3819 OFF
```

Spice Data Example:

```
%D %1 %2 %3 %M .67
```

ND, NG and NS are the drain, gate and source nodes, respectively. MNAME is the model name, AREA is the area factor, and OFF indicates an optional initial condition on the device for operating point analysis. The initial condition specification using IC=VDS, VGS only applies if the UIC option is enabled for the Transient Analysis. The TEMP value is the temperature at which this device is to operate, and overrides the temperature specification in the Analog Options dialog. See also: N-JFET, P-JFET. Example circuit: CSJFAMP.CKT.

MOSFETs

General Form:

```
MXXXXXX ND NG NS NB MNAME <L=VAL> <W=VAL> <AD=VAL>
+ <AS=VAL> <PD=VAL> <PS=VAL> <NRD=VAL> <NRS=VAL> <OFF>
+ <IC=VDS, VGS, VBS> <TEMP=T>
```

Netlist Example:

```
M6 23 16 0 17 MRF150
```

Spice Data Example:

```
%D %1 %2 %3 %3 %M TEMP=55
```

ND, NG, NS and NB are the drain, gate, source and bulk (substrate) nodes, respectively. MNAME is the model name. L and W are the channel length and width, in meters. AD and AS are the areas of the drain and source diffusions, in meters². Note that the suffix U specifies microns (1e-6 m) and P sq-microns (1e-12 m²). If any of L, W, AD, or AS are not specified, default values are used. The use of defaults simplifies input file preparation, as well as the editing required if device geometries are to be changed. PD and PS are the perimeters of the drain and source junctions, in meters. NRD and NRS designate the equivalent number of squares of the drain and source diffusions; these values multiply the sheet resistance RSH specified in the model for an accurate representation of the parasitic series drain and source resistance of each transistor. PD and PS default to 0.0 while NRD and NRS to 1.0. OFF indicates an optional starting condition on the device for DC analysis. The initial condition specification (optional) using IC=VDS, VGS, VBS only applies if the UIC option is enabled for the Transient Analysis, when a transient analysis is desired starting from other than the quiescent operating point. See the .IC device for a better and more convenient way to specify transient initial conditions. The TEMP value (optional) is the temperature at which this device is to operate, and overrides the temperature specification in the Analog Options dialog. The temperature specification is ONLY valid for level 1, 2, 3, and 6 MOSFETs, not for level 4 or 5 (BSIM) devices. See also: N-MOSFET 3T, N-MOSFET 4T, P-MOSFET 3T, P-MOSFET 4T.

MESFETs(GaAsFETs)

General Form:

```
ZXXXXXXX ND NG NS MNAME <AREA> <OFF> <IC=VDS,  
VGS>
```

Netlist Example:

```
Z1 3 5 6 ZM2 OFF
```

Spice Data Example:

```
%D %1 %2 %3 %M OFF
```

ND, NG and NS are the drain, gate and source nodes, respectively. MNAME is the model name, AREA is the area factor, and OFF indicates an optional starting condition on the device for operating point analysis. The initial condition specification using IC=VDS, VGS only applies if the UIC option is enabled for the Transient Analysis. See also: N-MESFET, P-MESFET.

Subcircuits

General Form:

```
XXXXXXXX N1 <N2 N3 ...> SUBNAM
```

Netlist Example:

```
XU1 7 5 6 12 3 XLM741
```

Spice Data Example:

```
%D %1 %2 %3 %4 %5 %S
```

Subcircuits are used in SPICE by specifying the device designation beginning with the letter X, followed by the circuit nodes to be used in expanding the subcircuit, followed by the subcircuit name. See also: Subcircuits. Example circuit: ANALOG.CKT.

Simcode Devices

General Form:

```
AXXXXXX [NPI NGI NI1I <NI2I ...>][NPO NI1O <NI2O ...> NO1O  
+ <NO2O ...>] MNAME
```

Netlist Example:

```
A2 [6 8 4] [7 9 5] 2404B
```

Spice Data Example:

```
%D [%4bi %2bi %1i] [%4bo %1o %2o] %M
```


Chapter 4

Digital Logic Simulation



One of the most important features of CircuitLogix is its ability to simulate your design, enabling you to detect and correct design errors before you invest time and money in the construction of actual hardware prototypes.

Digital electronics is the world of the computer. The binary 1's and 0's of the computer are actually the high and low voltage levels of tiny electronic devices known as integrated circuits. Digital logic simulation, then, becomes a relatively simple task because of the limited number of digital states that must be represented. CircuitLogix digital logic simulator is quick and fully interactive, meaning that you can flip switches, altering the circuit with the simulation free running and immediately see the response of the circuit.

All *Digital Only* and *Mixed-Mode* devices provided with CircuitLogix — 4xxx and 74xxx devices, Data Sequencer, Pulser, etc.—can be used in the digital logic simulator. In the Device Selection dialog box, above the picture of each device symbol appear the words “Digital Only Device”, “Analog Only Device”, “Mixed-Mode Device”, or “Schematic Only Device”. This indicates the simulation mode for which the currently displayed device is intended.

Refer to the Device Library chapter for a description of each device and the simulation mode for which it is intended. The Edit Device Data dialog box contains two check boxes, Analog and Digital. These check boxes are used to identify the simulation mode in which each device can be simulated. If you attempt to simulate a device using a simulation mode for which the device is not intended, CircuitLogix will display a warning message and that device will be ignored, producing an open circuit where that device is located.



The Trace Button



The Probe Tool



The Open Button

Using the Digital Logic Simulator

Digital simulation is completely interactive, meaning that the circuit responds immediately to changes from input stimulus, and the operation of the circuit is shown as it happens right on the screen. Operation of the circuit can be observed in several ways:

- With the CircuitLogix “Trace” feature enabled, the state of every node in the circuit is shown simultaneously as the simulation runs. In this mode wires at a logic one are shown as red, wires at a logic zero as blue, and wires at an unknown or tristate as green (these colors may be changed with the “Select Colors...” command in the Options menu).
- Any number of SCOPE probes can be connected to any nodes in the circuit, so that the timing diagrams for those nodes are shown in a separate digital Waveforms window. The timing information is updated continuously to show changes as they happen.
- Circuit operation can be observed by connecting any of a variety of displays and noting the conditions shown on them.
- You can use the Probe Tool to probe any wire in the circuit either during simulation or after it has been stopped. The logic states seen by the Probe Tool can also be charted in the Waveforms window.

Getting Started—Digital Circuit Examples

The best way to see how the digital simulation works is to load an example circuit and try some of the available commands. First, launch CircuitLogix by double-clicking on its icon. After a few seconds CircuitLogix will load into memory and will then be ready to use. To experiment with a pre-built circuit, follow these steps:

1. Click on the Open button in the Toolbar.
2. Select the file “SIM.CKT” from the list of available circuits. This circuit contains several mini-circuits and is useful for demonstrating CircuitLogix digital simulation features.



3. Click on the ON/OFF button in the Toolbar to start the simulation. The fact that the simulation is running will be evident because you will see a Hex Display showing a count sequence.



The Probe Tool

4. Select the Probe Tool from the Toolbar and touch its tip to the wire just to the left of the label “Probe Wire To the Left”. If the Probe Tool shows a small triangle pointing down, the wire is at a zero state, if it shows a triangle pointing up, the wire is at a one state, and if the probe indicator is empty, this would indicate an unknown or tristate.



The Waveforms Button

5. Move the tip of the Probe Tool to the Logic Switch labeled “Toggle Switch” and click near its center. The Logic Display connected to the output of this mini-circuit should then start to toggle on and off rapidly.

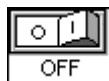
6. Click on the Waveforms button in the Toolbar to open the digital Waveforms window. Each node in the circuit that has a SCOPE attached is charted in this window.



The Trace Button

7. Select “Scope Probe” from the Options menu. A new waveform titled “Probe” will be displayed in the Waveforms window. Watch what happens to this waveform as you move the Probe Tool around the circuit.

8. Click on the Trace button in the Toolbar to see the state of every wire in the circuit as it changes.



9. Click on the ON/OFF button again in the Toolbar to stop the simulation.

Simulation Tools

Several buttons in the Toolbar are used specifically for simulation. These buttons are described below. The use of these tools is demonstrated in the examples shown earlier in this chapter. It should be noted that the functionality of these tools is somewhat different for analog/mixed-mode simulation.



The Simulation Tools

The Digital/Analog Button:  

Click on the Digital/Analog button to choose the type of simulation you want to run. When the AND gate icon is displayed, digital simulation is enabled; when the transistor icon is displayed, analog simulation is enabled.

The Reset Button: 

Click on the Reset button to initialize the simulation. Reset can also be selected from the Options menu or by pressing Ctrl+Q.

The ON/OFF Button: 

Click on the ON/OFF button to start the simulation. Click on the ON/OFF button again to stop the simulation. This can also be selected in the Options menu or by pressing F10. When the simulation is running you can't perform edit operations such as move and delete (if you try, CircuitLogix will generate an audible "beep"). You can flip switches with the Arrow or Probe Tool, and view or toggle the state of a wire with the Probe Tool.

The Probe Tool: 

The Probe Tool can also be selected from the "Cursor Tools" option in the Options menu or by pressing Alt+P. Use the Probe Tool to monitor the state of any node in the circuit or to inject a state into a node. To see the state of a node, either while the simulation is running or after it has stopped, touch the Probe Tool's tip on a wire or device pin. If the small triangle in the head of the Probe Tool points up the node is at a one state, if it points down it is at a zero state, and if the triangle doesn't appear the node is at an unknown or tristate. If the "Scope Probe" option is enabled, the states seen by the Probe Tool will be charted in the Waveforms window.

To inject a state into a node touch the tip of the Probe Tool on a wire or device pin and click the left mouse button. The state of that node will then be changed to be opposite of what it was (a one becomes a zero and a zero becomes a one). To inject a tristate signal, hold down the SHIFT key and click. Remember, if the node is driven by some other device the state change will be immediately overridden because the device will drive the node back to its original state.

The Probe Tool can also be used to flip switches while the simulation is running.

The Trace Button: 

Click on the Trace button to turn the trace feature on or off. Trace can also be selected from the Options menu or by pressing F11. With trace on, the states of all nodes within the circuit are monitored as the circuit runs. CircuitLogix does this by drawing the wires in different colors to show the state of each wire. A wire at a one state will be shown as red, a wire at a zero state will be shown as blue, and a tristate wire will be shown as green. This feature can be used to debug your circuit or to simply provide a convenient way of observing operation of the circuit.

NOTE: Because the wires in the circuit will be redrawn each time they change states, turning this option on may slow down the speed at which the simulation runs.

The Waveforms Button:

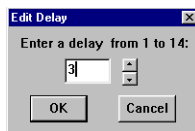


In digital mode, clicking on the Waveforms button will open or close the digital Waveforms window. This can also be selected in the Options menu or by pressing F12. Waveforms are described in detail later in this chapter.

Propagation Delays

The delay of a device determines how many simulation ticks it takes for a signal to propagate from the input to the output of the device. The default delay for all devices is 1, but this value can be changed to any value from 1 to 14. It is left to the user to determine the real-time value of each tick. The concept is that if one device has a delay of one and another a delay of three, then in the real world the second device would have a propagation delay three times larger than the first device.

To change the delay of one or more devices, first select the device(s), then select the “Set Prop Delay(s)...” command from the Edit menu. The following dialog box will then be displayed allowing you to enter a new value for the delay or to cancel the command.



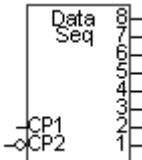
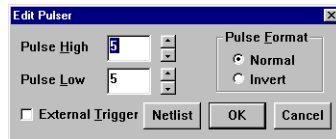
Use the “Show Prop Delays” command in the View menu to display the propagation delay of all devices in the circuit. The delay values will be shown within a rounded rectangle located near the center of each device. Some devices (Pulsers, Logic Display’s, macro devices, etc.) do not have a delay, so no value will be shown. In the case of macro devices, the delay is determined by the individual delay setting of each device within the macro.

The Pulser



The Pulser is a digital pulse generator which provides a continuous stream of highs and lows. The pulse format, time high, time low, and trigger mode are individually programmable for each Pulser in the circuit.

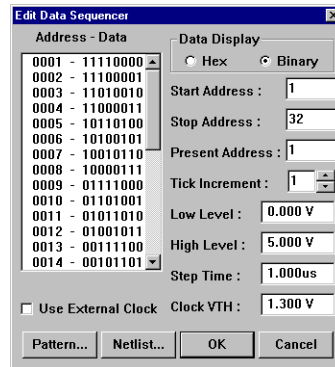
To edit the Pulser settings, double-click on the Pulser with the Arrow Tool. The following dialog box will then appear. You can change the number of simulation ticks for which the pulse will stay high and low, the format of the pulse (normal or inverted), and whether the pulser is in free run or external trigger mode.



The External Trigger option allows the Pulser to be used as a programmable one-shot. In this mode, the CP1 and CP2 inputs serve as rising and falling edge trigger inputs respectively. If either pin receives a trigger pulse then the outputs of the Pulser will go active on the next simulation tick and remain active for “Pulse High” ticks. The Pulser can be retriggered in this mode so additional trigger pulses occurring before the completion of the cycle will cause the active time to be extended for another cycle.

The Data Sequencer

This device can be used in both digital and analog simulation modes. Also known as a Data or Word Generator, it allows the user to specify up to 32767 8-bit words which can be output in a defined sequence. Since there is no limit to the number of Data Sequencers that can be used in a circuit, several could be placed in parallel to create a data stream of any width. Double-click on the Data Sequencer with the Arrow Tool to display the following dialog box.



The Start Address is the address of the data that will be output first when the simulation begins. The Stop Address is the address of the data that will be output last before the sequence repeats.

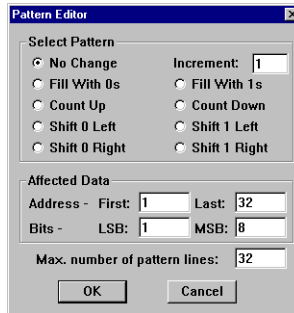
When Use External Clock is enabled, the CP1 and CP2 inputs serve as rising and falling edge clock inputs, respectively. If either pin is clocked, the Data Sequencer will advance to the next address.

Digital simulation mode only: The Present Address indicates the address of the data that will be output next if the circuit is not reset or modified. The Tick Increment specifies how many simulation ticks will occur before the output is advanced to the next address when the external clock is disabled.

Analog simulation mode only: Low Level and High Level indicate the output voltage levels. Step Time indicates the length of time that the outputs remain at each address when the external clock is disabled. Clock VTH is the voltage threshold level at which the external clock pins will cause the outputs to advance to the next address.

The Pattern Editor

Pattern data (the Data field in the Data Sequencer dialog box) can be typed directly into the list box. However, when creating a large pattern this method would become quite time consuming. The Pattern Editor helps you to quickly create large, complex patterns. When you click on the Pattern button, the following dialog box will appear.



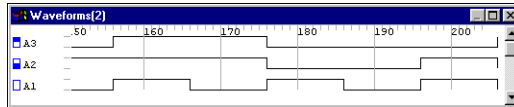
The Pattern Editor not only allows you to enter predefined pattern sequences, it also allows you to specify which rows (addresses) and columns (bits) will be affected. For example, you could fill just one column with a count up sequence to produce a stream of ones and zeros on a single output. Or you could fill rows 23-67 with ones in just 5 columns and fill the same rows with a repeating Shift 0 Left in the other 3 columns.

Increment indicates how many rows will contain the same data before the next change in pattern. For example, with an increment of 3, a Shift 1 Left pattern will only shift on every third pattern row.

The maximum pattern size for each Data Sequencer can be set by the user. This allows you to create small patterns for several Data Sequencers without allocating large amounts of memory or creating large circuit files. The Max. number of pattern rows, by default, is set to 32, but can be increased as needed to as much as 32767. It can never be smaller than the Stop Address. When you increase the maximum number of pattern rows, the new rows will be filled with zeros. Any time you *decrease* the maximum number of pattern rows, any pattern data that was stored in the upper addresses will be permanently lost.

Digital Waveforms

By attaching SCOPes [Digital/Instrument] (t) to points of interest in the circuit, the states of these nodes can be graphed over time as the simulation runs. Choose “Waveforms” in the Options menu or click on the Waveforms button in the Toolbar to display or hide the digital Waveforms window. An example of what the window looks like is shown below.



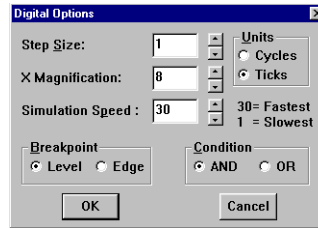
Before you can view timing waveforms for any node in your circuit, you must connect a SCOPE to each node to be monitored, or, if you have enabled the “Scope Probe” option in the Options menu, you can monitor the states of the Probe Tool in the Waveforms window.

The order in which the waveforms are displayed can be changed. Simply use the mouse to point at any of the scope labels in the window and then press and continue to hold down the left mouse button. A rectangle will be drawn around the label indicating that it is selected and you can then move it to the desired position. Once you have moved a label to the desired position release the mouse button and the waveforms will be automatically reordered in the window. You can repeat this process as often as desired in order to position the waveforms in any order. When you save a circuit to disk, the order of the waveforms is saved also, so upon opening the circuit later the waveform ordering is retained.

The digital Waveforms window is used in conjunction with the “Digital Options” (see below) to set breakpoints in a circuit. To set a breakpoint, click in the small square to the left of a SCOPE’s label in the Waveforms window. Clicking once will fill the bottom portion of the square indicating a break on zero condition. Clicking a second time will fill the top portion of the square indicating a break on one condition. Clicking a third time will return the square to its empty state indicating no break condition.

Digital Options

The Digital Options dialog box is used to control the size of a step when running the simulation in single step mode, to set the conditions for break points and to set the simulation speed. The following dialog box will appear when you choose “Digital Options...” from the Options menu.



Step size can be measured in either ticks or cycles. A cycle always consists of 10 ticks. A tick is the smallest unit of delay for the digital simulator. It takes one tick to perform a single step of the simulation for all devices.

X Magnification can be adjusted to view a larger or smaller section of the waveforms in the digital Waveforms window. By default, the magnification is set to 8. A smaller value will zoom out, a greater value will zoom in.

The Simulation Speed text edit field enables you to determine how fast the simulation runs. This would be useful, for example, if the simulation is running too fast to view the states of a seven-segment display. By setting this field to a lower number, the simulation would be slowed down enabling you to view the changes of the display. Another method of slowing the simulation would be to run it in single step mode or set breakpoints.

The break options are used in conjunction with the waveforms window to set breakpoints. When set for “Level - And”, all break conditions must be met before the simulation will stop. When set for “Level - Or”, any one of the break conditions will stop the simulation. When set for “Edge - And” the simulation will stop when the proper edge occurs on all of the specified waveforms. When set for “Edge - Or” the simulation will stop if a transition to any of the specified conditions occurs.

Digital Tutorial

In this section, a series of simple experiments will all be performed using CircuitLogix. The experiments are designed to assist in teaching someone the basic theory of digital electronics as well as how to design simple digital circuits. It should be noted however, that this course in itself is probably not sufficient to teach someone with no prior electronics experience the entire concept of digital circuit theory and design. It is recommended that if you do not have at least some prior digital circuit knowledge, that you obtain an additional in depth beginners level book on the subject to aid you in the learning process.

The approach taken with this tutorial is to first teach you the meaning of various terms and symbols which are used to describe the circuits which will be built, and then to present a series of experiments which are performed with the aid of CircuitLogix. Through this series of experiments you will learn how to wire the available parts to form a specific circuit which will be used as the basis for a series of questions. These questions when answered correctly will provide the theoretical knowledge of digital electronics for this course. Wiring the circuits and getting them to work as well as your own experimentation will provide the practical or hands on knowledge for this course.

What is Digital Electronics?

The first step we need to take is to gain an understanding of what is meant by digital electronics. A digital system can be defined as one in which all communication within the system is carried out in a digital manner. Typically, this digital communication means that all signals within the system can have only three possible conditions or states. A signal can be on, off, or tristate. Digital electronics is the branch of electronics concerned with electronic systems which are composed of elements that exhibit this digital behavior. Because a digital system can only exhibit one of three possible states, they are usually easier to understand than analog systems which can have an infinite number of states. A large number of digital systems are even simpler than those just described because they have only two states, on or off. The light in the living room of your home is probably a digital system. It is either on or off and by flipping the switch you control this simple digital system.

In this tutorial we will limit ourselves to the study of digital electronic circuits which have only two possible states. Tristate devices will not be covered in this tutorial. You will learn the language used to describe and talk about these circuits as well as how to construct and verify operation of the circuits.

One barrier which must be overcome first is the language or jargon barrier. Digital electronics has a language all its own and it is necessary for you to understand this language in order to successfully understand and construct the circuits.

Presented on the next few pages is a short list of the most important terms which you will see used and the meaning of each. Please study this list before beginning any of the experiments.

Frequently Used Terms

clock pulse: A cycle within a circuit from a logic 0 to a logic 1 and back to a logic 0 or visa versa. Clock pulses are commonly used as control signals within a circuit.

clock generator: A device which produces a clock pulse(s). Clock generators are available as a single integrated circuit and are commonly used as the clock source for microprocessor and other integrated circuits which require a clock in order to function. A clock generator called a “pulser” is provided in CircuitLogix and the pulser is completely programmable from within CircuitLogix.

complement: The opposite state. The complement of a logic 0 is a logic 1 and the complement of a logic 1 is a logic 0. Inverters are commonly used to invert or complement signals within a circuit.

component: The smallest subsystem of a circuit. The integrated circuits and LED’s are components of a circuit.

connection: The joint or attachment between two components in a circuit. In order to make any circuit work it is necessary to connect the components together in the correct way.

device: See component.

DIP: See dual in-line package.

dual in-line package: A type of container or housing for integrated circuits. The dual in-line package or DIP has pins extending from it to which wires can be connected in order to wire a circuit.

frequency: The number of times that a logic level changes states each second. The unit of frequency is the hertz. If a level changes from 0 to 1 to 0 100 times each second in some circuit that signal is said to have a frequency of 100 hertz.

ground: The reference point for all voltage measurements within a circuit. It is called ground because often times this point is actually connected to the earth through a conductor.

IC: Integrated circuit. A small device which is composed of many tiny transistors and other components connected together to perform some useful function. Integrated circuits are packaged in a manner which makes them easy to use in designing circuits. IC's can be compared to tiny black boxes which can be wired together to form the heart of everything from computers to digital clocks.

LED: Light emitting diode. A semiconductor device which resembles a tiny light bulb and can be turned on or off. LED's are available in several colors such as red, green, and yellow and are commonly used as power indicators and such on computers and other electronic devices.

logic 0: One of the possible voltage levels within a digital circuit. In many circuits a logic 0 is 0 volts.

logic 1: The opposite or complement of a logic 0. In many circuits a logic 1 is 5 volts.

logic diagram: A special type of diagram used to show the logic elements and the way they are interconnected in a digital circuit.

Logic Display: This is a device used in CircuitLogix which indicates the state of the wire to which it is connected. When the wire is in a 1 state, the display 'lights'.

Logic Switch: This is a device used in CircuitLogix which acts as a connection to either +V or to ground depending on which way it is switched.

node: Any point along a wire or wires that are connected between two points in a circuit. Any point along a wire connected between two integrated circuits would be called a node.

power supply: The source of electrical power for a circuit. The power supply provides the voltage and current necessary

to operate a circuit. In CircuitLogix you are not required to connect a power supply to the devices to make them function. It is very important that you realize that in the real world a circuit will not function unless every IC is connected to a power supply of the correct voltage, usually 5 volts.

pulse: A change from a logic 0 to logic 1 and back to logic 0 or visa versa.

schematic diagram: A special type of drawing which shows the interconnections of all devices which form a circuit.

seven-segment LED: A special type of LED in which seven individual LED's are formed into the shape of an 8. All segments are individually controllable and can be turned on in various combinations to form the numbers 0 through 9.

state: The logic level at some point in a circuit.

truth table: A tabular table used to show the behavior of a circuit for all possible logic states presented at the inputs of the circuit.

voltage: Potential difference between two points in a circuit. Voltage represents the pressure that causes electrons to move within a circuit. The symbol for voltage is "V".

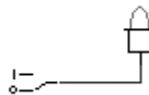
Experiment 1

Introduction - The purpose of this experiment is to demonstrate what a logic 0 and logic 1 state are within a circuit by using the Logic Switch and Logic Display devices. You will also learn the necessity of wiring components together correctly in a circuit in order for the circuit to function as planned.

All digital circuits operate based on three possible logic levels present in the circuit. These logic levels are logic 0, logic 1, and tristate. The tristate logic level is a special case, and will not be studied in this series of experiments. Logic levels 0 and 1 (also known as low and high levels) refer to voltage levels and for the type of integrated circuits available in CircuitLogix these levels are typically 0V and 4.5V respectively. Because digital circuits operate based on only two levels, learning the basics of digital circuit theory and design is not difficult. The logic 0 and logic 1 are the control signals which will cause all integrated circuits to operate. The operation of a digital circuit can be visualized by imagining a black box into which various control signals are passed (logic levels 0 and 1) and out of which some other combination of logic levels comes.

The Logic Display is a simple device used in CircuitLogix to indicate the state of any given node or point in a digital circuit. Typically, however, in a real digital logic circuit you would use a Light Emitting Diode (LED) which, when its anode is connected through a current limiting resistor to +5V and its cathode is pulled to a Low state, will emit light. LED's are widely used as display indicators in everything from personal computers to stereos. They are used to show such things as power is applied and that a cycle has finished. LED's are available in several colors such as red, green, and yellow.

Schematic Diagram of Circuit



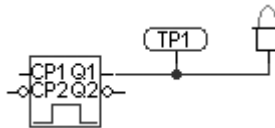
- Step 1** Wire the circuit shown in the schematic diagram above using a “Logic Switch” [Switches/Digital] (s) and a “Logic Display” [Displays/Digital Only] (9). Be careful to wire it as shown in the diagram. If you need help using CircuitLogix at any point along the way refer to the appropriate section of the users manual.
- Step 2** Click on the “Run” button in the Toolbar. Now move the mouse until it is pointing at the switch. Click on the switch several times and observe that it moves up and down. This is a special type of switch which outputs a logic 0 when down and a logic one when up. Use the mouse to change the switch to a logic 0. What do you observe on the Logic Display?
- Step 3** Use the mouse to change the switch to a logic 1. Now what do you observe on the Logic Display?
- Step 4** Select the “Delete Tool” from the Toolbar. Now point at the wire between the switch and the Logic Display and click. Now click on the “Run” icon in the Toolbar. What do you observe on the Logic Display now?
- Step 5** Change the switch to a logic 0. Now what do you observe on the Logic Display? How can you easily tell what logic state is present at any point in a circuit at any time?

Experiment 2

Introduction - The purpose of this experiment is to learn how the Pulser and SCOPE operate in CircuitLogix and how they are used in the design and testing of digital circuits.

The SCOPE available in CircuitLogix performs the function of a logic analyzer which is used in the real world to view logic levels in a circuit. The Pulser performs the function of a programmable clock source which would typically be built out of several integrated circuits in a real circuit. These two devices will be used extensively throughout the experiments so it is important that you understand how they function.

Schematic Diagram of Circuit



Step 1

Wire the circuit shown in the schematic diagram above using the following devices:

Pulser	[Digital/Instrument] (p)
Logic Display	[Displays/Digital Only] (9)
SCOPE	[Digital/Instrument] (t)

Step 2

Start the simulation by clicking on the “Run” icon in the Toolbar. What do you observe on the Logic Display?

Step 3

Click on the “Waveforms” icon in the Toolbar. What do you observe?

Step 4

Stop the simulation by clicking on the “Stop Sign” icon in the Toolbar. Select the “Arrow Tool” from the Toolbar and then double-click on the Pulser. A dialog box will then appear which enables you to edit the Pulser. Change the “Pulse High” to 1 and the “Pulse Low” to 9. Close the dialog box and then start the simulation again. Now what do you observe?

Step 5

Edit the Pulser again but this time change the pulse format to inverted by clicking the “Invert” button in the dialog box. Start the simulation again. How is the waveform different this time?

Experiment 3

Introduction - The purpose of this experiment is to learn what an AND gate is and what the truth table for it is.

Integrated circuit gates are the fundamental building blocks of digital circuits. A gate is simply an integrated circuit which produces specific outputs for any given input condition. There are six types of gates that are very common in digital circuits. They are the INVERTER, the AND gate, the OR gate, the NAND gate, the NOR gate, and the XOR gate. These gates are available as integrated circuits which combine several gates into one package. The AND gate for example can be purchased at any local electronics store as an integrated circuit called the 74LS08 and this one IC contains four AND gates in the same package. The name 74LS08 is derived by the manufacturer and is numerical in order to make it easy to keep track of a wide variety of IC's.

The term gate is used because these devices serve as an entrance through which signals in a circuit pass. A gate can be pictured as a device into which digital signals enter and which has an output or outputs from which new signals come. The resulting signals out of a gate depend on what the input signals were and are based on a very precise set of rules called a truth table. Each type of gate has its own truth table which determines what the outputs will be given any set of inputs.

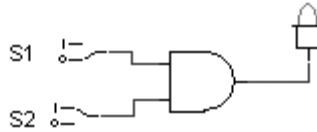
In this experiment you will use the AND gate. You will learn what the truth table for the AND gate is by experimental methods.

Step 1

Wire the circuit as shown below using the following devices:

2-In AND [Digital/Gates] (3)
Logic Display [Displays/Digital Only] (9)
Logic Switch [Switches/Digital Only] (s)

Schematic Diagram of Circuit



Step 2

Start the simulation. Then use the mouse to toggle the two switches to each of the four combinations shown in the following table. Record in the following table the state of the Logic Display for each combination of the switches.

S1	S2	Output of AND
0	0	
0	1	
1	0	
1	1	

You have now discovered the truth table for the AND gate. Can you see why it is called an AND gate? It's because both input one **AND** input two must be at a logic 1 in order for the output to be a logic 1. Can you guess what the truth table for the OR gate might be? Proceed to the next experiment to see if you are right.

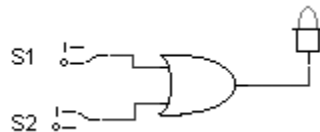
Experiment 4

Introduction - The purpose of this experiment is to learn what the truth tables for the OR, INVERTER, NAND, NOR, and XOR gates are. These gates are fundamental to almost all digital circuits and understanding their truth tables is a prerequisite to gaining an in depth understanding of digital electronics. Spend as much time as is required to thoroughly understand these devices. It will pay off later by allowing you to use your creativity to design circuits on your own.

Step 1

Wire the circuit shown below for the OR gate.

Schematic Diagram of Circuit



Step 2

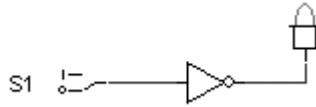
Fill in the truth table below for the OR gate by changing the switches to all four possible combinations and recording the output state in the following truth table.

S1	S2	Output of OR
0	0	
0	1	
1	0	
1	1	

Did the truth table turn out as you expected?

Step 3

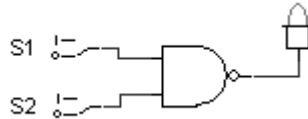
Wire the circuit for the INVERTER and then fill in its truth table.



S1	Output of Inverter
0	
1	

Step 4

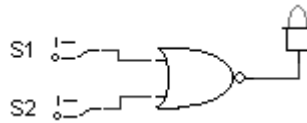
Wire the circuit for the NAND gate and then fill in its truth table.



S1	S2	Output of NAND
0	0	
0	1	
1	0	
1	1	

Step 5

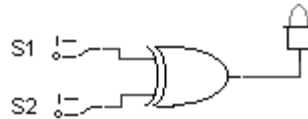
Wire the circuit for the NOR gate and then fill in its truth table.



S1	S2	Output of NOR
0	0	
0	1	
1	0	
1	1	

Step 6

Wire the circuit for the XOR gate and then fill in its truth table.



S1	S2	Output of XOR
0	0	
0	1	
1	0	
1	1	

Experiment 5

Introduction - The purpose of this experiment is to learn how one more complex type of integrated circuit works by examining the functionality of the dual D flip-flop.

This IC is commonly available as the 74LS74 integrated circuit. It is a very useful device and is frequently used in computer circuits. It is called a flip-flop because it has two pairs of complementary outputs which are usually opposite states from each other. The outputs are called Q and QN respectively and there are two Qs and two QNs on a 74LS74. Each Q and QN pair function independently of the other pair so it is really like having two of the exact same thing in one package. If Q is high then QN will be low and visa versa (except under one special condition).

The truth table for the 74LS74 is shown below. Only one of the two flip-flops is shown for this device because both behave identically. Each line of the truth table will now be explained. The “X” in the truth table means that it doesn’t matter what level that pin is at.

Truth Table for 1/2 of the 74LS74

	CP	D	S	R	Q	QN
Line 1	↑	L	1	1	0	1
Line 2	↑	H	1	1	1	0
Line 3	X	X	1	0	0	1
Line 4	X	X	0	1	1	0
Line 5	X	X	0	0	1	1

Line 1 - The arrow pointing up means that in order for that line of the truth table to work as shown the CP pin must move from a logic 0 to a logic 1. The “L” means that the D pin must be at a logic 0 prior to the 0 to 1 transition of the CP pin and it must stay at a 0 until this transition has occurred. If all these conditions are met then a 0 is “clocked” into the flip-flop.

Line 2 - The “H” means that the D pin must be at a logic 1 prior to the 0 to 1 transition of the CP pin and it must stay at a 1 until this transition has occurred. This is the same as line 1 except for the state of the D pin. If all conditions shown in line 2 are met then a 1 is “clocked” into the flip-flop.

Line 3 - This is the “reset” mode. If the R pin goes to a logic 0 while all other inputs are at the specified levels the flip-flop is said to “reset” and Q goes to a logic 0 while QN goes to a logic 1.

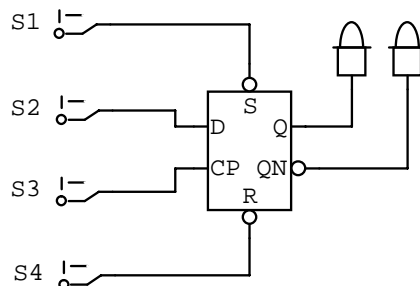
Line 4 - This is the “set” mode. If the S pin goes to a logic 0 while all other inputs are at the specified levels the flip-flop is said to “set” and Q goes to a logic 1 while QN goes to a logic 0.

Line 5 - This is the “undetermined” mode. It is called this because if both S and R go to a 1 simultaneously the state of the outputs is unpredictable.

Step 1

Wire the circuit as shown in the following schematic diagram using the following devices:

74LS74	[Digital ICs by Number/74LSxx]
Logic Switch	[Switches/Digital] (s)
Logic Display	[Displays/Digital Only] (9)



- Step 2** Set switch S4 (R pin) and switch S1 (S pin) to logic 1. Now start the simulation by selecting the “Run” command. Flip switch S3 (CP) to a logic 1. What do you observe on the Logic Displays connected to Q and QN? Which line of the truth table above does this condition correspond to?
- Step 3** Flip switch S3 (CP) back to 0. Do the outputs change state? Why not?
- Step 4** Flip switch S2 (D) to a logic 1. Now switch S3 (CP) from a 0 to a 1. What state are the outputs in now?
- Step 5** Change all switches so that they are as shown on line 3 of the truth table. Do the outputs correspond to the conditions for the “reset” state?
- Step 6** Change all switches so that they are as shown on line 4 of the truth table. Do the outputs correspond to the conditions for the “set” state?
- Step 7** Change all switches so that they are as shown on line 5 of the truth table. Do the outputs correspond to the conditions for the “undetermined” state?

Experiment 6

Introduction - The purpose of this experiment is to learn how the Hex Display works and how it can be used as a display indicator. The Binary Coded Decimal (BCD) counter (74LS168) will be used to drive the Hex Display so it will be studied also.

The Hex Display is a specialized version of the Logic Display studied in the previous experiment. A Hex Display is simply a collection of specially shaped Logic Displays arranged in one package in a manner such that the individual display segments can be turned on to form numbers. Seven-segment LED's are used extensively in electronic devices to show such things as the time of day and the cook time on microwave ovens.

The 74LS168 is a counter IC. It has four outputs called Q0, Q1, Q2, and Q3. For any device which has multiple outputs the states of all the outputs at any given time can be represented by a decimal number. Thus instead of saying that Q0=0, Q1=1, Q2=0, and Q3=1 we can say the Q's are at 9. This number is obtained by using a simple number system called the "binary number system". It is based on having an assigned value or weight for each output. The final value is then obtained by adding the weights for all the outputs.

The binary number system assigns the lowest member a weight of 1, the next highest member a weight of 2, the next 4, the next 8, and so on. For the 74LS168 Q0 is the lowest member so it has a weight of 1, Q1 a weight of 2, Q2 a weight of 4, and Q3 a weight of eight. Thus if all Q's were at a logic 1 we would say that the Q's have a decimal value of 15 (1+2+4+8).

Step 1

Given the following table of possible output states for the 74LS168, fill in the equivalent decimal values.

Q3	Q2	Q1	Q0	Decimal Value
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

You have just learned the binary number system. This system is used extensively in digital circuit design so it is important that you understand it.

The 74LS168 has several valid modes of operation. It is capable of counting up (0 to 9 and back to 0) as shown in the above table and it is also capable of counting down (9 to 0 and back to 9). It also has several other useful modes but these will not be covered in this experiment. The modes used in this experiment are shown in the following table.

CP	CEP	CET	PE	UD	Mode
^	0	0	1	1	Count Up
^	0	0	1	0	Count Down

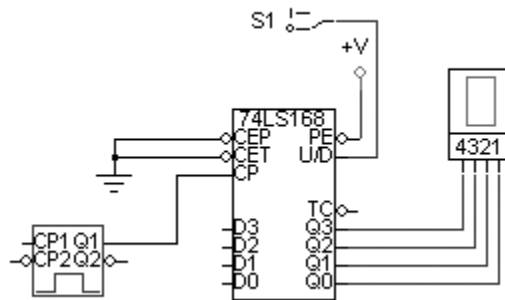
The Hex Display has four inputs. The state of these inputs determines what segments turn on. The Hex Display is constructed so that if a binary 0111 is present at the inputs, the segments are turned on so that a seven is formed on the display. If a binary 0100 is applied, the segments turn on so that a two is formed, etc.

In this experiment, the 74LS168 will be connected to the Hex Display and configured so that it can count up or down based on the position of a switch. The result will be that the sequence 0 to 9 and back to 0; or 9 to 0 and back to 9 will be observed on the Hex Display.

Step 2

Wire the circuit as shown in the following schematic diagram using the following devices:

74LS168	[Digital ICs by Number/74LSxxx]
Pulser	[Digital/Instrument] (p)
Hex Display	[Displays/Digital Only] (h)
+V	[Digital/Power] (1)
Ground	[Digital/Power] (0 (zero))
Logic Switch	[Switches/Digital] (s)



Step 3

Start the simulation. What do you observe on the Hex Display?

Step 4

Flip S1 to 0. Now what do you observe on the Hex Display?

Conclusion

Hopefully you have learned many new things through this series of experiments. This is only a beginning, however, as the world of digital electronics is a vast and ever changing one.

At this point it is up to you to seek increased knowledge about digital electronics. This series of experiments has introduced you to digital circuits, but you have barely scratched the surface of this vast field. Your local or university library probably has several good books that can provide a more in depth understanding of the subject. As you read and study on your own you will find example circuits that you would like to build and experiment with their functions. CircuitLogix is ready and able to help you test these designs as well as any you come up with on your own. Don't be afraid to try things out just to see what will happen, with CircuitLogix you will never destroy an expensive IC or other device.

Answers to Experiment 1

Step 2 - The Logic Display should be off. The Logic Displays in CircuitLogix are wired internally so that a logic 0 causes them to turn off and a logic 1 causes them to turn on.

Step 3 - The Logic Display should be on since a logic 1 is present.

Step 4 - The Logic Display should be off since the wire connecting the switch to the display has been removed. When the input to the display is disconnected the display will be off.

Step 5 - The Logic Display should still be off since the wire connecting the switch to the display has been removed. Since the display is disconnected from the circuit it doesn't matter what state the switch is in, the display will remain off.

Answers to Experiment 2

Step 2 - The Logic Display should be toggling between being on and off. This occurs because the Pulser is producing an alternating 0 and 1 condition.

Step 3 - The waveform for the node connected to the SCOPE will be shown in the "Waveforms" window. When the waveform is at a logic 1 it corresponds to the Logic Display being on and when the waveform is at a logic 0 it corresponds to the Logic Display being off.

Step 4 - The waveform is now high for only one tick whereas it was high for 4 ticks before. In CircuitLogix ticks are used as the measure of time whereas in the real world the time that the waveform is high would be measure in some fraction of a second such as nanoseconds. CircuitLogix uses the tick because it is not possible to simulate the circuit at the same speed as a real circuit.

Step 5 - The waveform has been turned upside down or inverted.

Answers to Experiment 3

Step 2 - The output column of the truth table should contain 0, 0, 0, 1. For an AND gate the output will only be high if both input 1 and input 2 are high.

Answers to Experiment 4

Step 2 - The output column of the truth table for the OR gate should contain 0, 1, 1, 1. For an OR gate the output will be high if either input 1 or input 2 is high. The output will be low only if both input 1 and input 2 are low.

Step 3 - The output column of the truth table for the inverter should contain 1, 0. The inverter complements or inverts whatever logic level appears at its input.

Step 4 - The output column of the truth table for the NAND gate should contain 1, 1, 1, 0. The NAND gate is just an AND gate with an inverter connected to its output.

Step 5 - The output column of the truth table for the NOR gate should contain 1, 0, 0, 0. The NOR gate is just an OR gate with an inverter connected to its output.

Step 6 - The output column for the XOR or exclusive OR gate should contain 0, 1, 1, 0. This gate produces a high output if either input 1 or input 2 are high but not if both inputs are high.

Answers to Experiment 5

Step 2 - Q1 should be low and Q1N should be high since we loaded a 0 into the flip-flop. This corresponds to line 1 of the truth table.

Step 3 - No, CP1 must change from 0 to 1 in order to load a new state into the flip-flop. Even if this had occurred the outputs wouldn't have changed since D1 stayed at a logic 0.

Step 4 - Q1 is low, Q1N is high. We loaded a 1 into the flip-flop. This corresponds to line 2 of the truth table.

Step 5 - Yes, the flip-flop is reset corresponding to line 3 of the truth table.

Step 6 - Yes, the flip-flop is set corresponding to line 4 of the truth table.

Step 7 - Yes, the flip-flop is in the undetermined state which corresponds to line 5 of the truth table.

Answers to Experiment 6

Step 1 - The decimal equivalents are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The 74LS168 produces the binary representation of these decimal numbers when counting.

Step 3 - The Hex Display should be counting up and each time it reaches 9 it should start over at 0. This is because the counter is in the "count up" mode.

Step 4 - The Hex Display should be counting down and each time it reaches 0 it should start over at 9. This is because the counter is in the "count down" mode.

Chapter 5

File Menu

The File menu contains commands that enable you to open, save and print circuits and waveforms.



File	
New	Ctrl+N
Open...	Ctrl+O
Close	
Revert...	
Merge...	
Save	Ctrl+S
Save As...	
Export Options...	
Export Circuit...	
Export Waveforms...	
Print Setup...	
Print Circuit...	Ctrl+P
Print Waveforms...	
Script Functions...	Shift+SPACE
Preferences...	
Exit	

New

Select the “New” command to clear the current circuit from the work area and begin a new circuit. If an unsaved circuit is present in the work area when you select this command, you will be asked if you want to save the current circuit first.

Open...

Choose the “Open...” command to abandon the current circuit and load a new circuit into the work area. If an unsaved circuit is present in the work area, you will be asked if you want to save the current circuit first. A file selector dialog will then be displayed allowing you to choose the circuit to load into the work area. The file extension used by default is .CKT.

Close

Use the “Close” command to close the open windows. If changes have been made to the current circuit you will be asked if you want to save the circuit before closing.

Revert...

Choose the “Revert...” command to abandon any changes made since last saving the circuit to disk and load the last saved version back into the work area. CircuitLogix will then display a dialog box asking you to confirm that you want to revert to the last saved version.

Merge...

The “Merge...” command enables you to add a circuit saved on disk to the circuit in the work area. This command is useful because it allows you to save commonly-used circuits or portions of circuits to disk, then reuse them as often as you need without having to rebuild them from scratch each time.

Select the “Merge” command and choose the circuit to be added to the work area. The circuit will be placed starting at the top left corner of the screen, so be sure to position the work area and existing circuit accordingly.

Save

Choose the “Save” command to save the current circuit in the work area to disk using the name shown in the Title Bar. When you save a circuit whose title is UNTITLED.CKT, a file selector dialog will be displayed allowing you to save it under a different name.

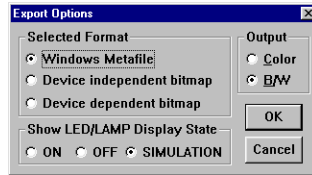
Save As...

Choose the “Save As...” command to save the current circuit to disk using a file name that is different than the one shown in the Title Bar. CircuitLogix will display a file selector dialog allowing you to choose the path and a file name of up to eight characters. The circuit is saved in an ASCII format with a default file extension of .CKT.

Export Options...

The “Export Options...” allows you to specify the format used when transferring graphics to the clipboard or when exporting a graphic to a file.

Windows Metafile format is Windows’ format for vector graphics. A device independent bitmap is a pixel-by-pixel representation of a graphic that looks good at any resolution. A device dependent bitmap is a pixel-by-pixel representation of a graphic that only looks good at its original resolution.



Export Circuit...

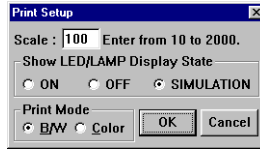
The “Export Circuit...” command allows you to save the circuit on disk, in the format selected in the *Export Options* described above (as a Windows Metafile, Device Independent Bitmap or Device Dependent Bitmap). When you choose this command, a dialog box will be presented enabling you to select the name of the file where the circuit will be saved.

Export Waveforms...

The “Export Waveforms...” command allows you to save the waveforms for the current circuit on disk, in the format selected in the *Export Options* described above (as a Windows Metafile, Device Independent Bitmap or Device Dependent Bitmap). Only the information currently displayed in the active Waveforms or Analysis window will be saved. When you choose this command, a dialog box will be presented enabling you to select the file where the waveforms will be saved.

Print Setup...

The “Print Setup...” command allows the desired printer and page orientation to be selected before printing. This is done through two dialog boxes. The first dialog box is the normal Windows Print Setup dialog. The second dialog box, shown below, appears after you have clicked the OK button in the Print Setup dialog box. This second dialog box allows you to scale the output to the printer 10% to 2000%, and choose between black & white and color.



Print Circuit...

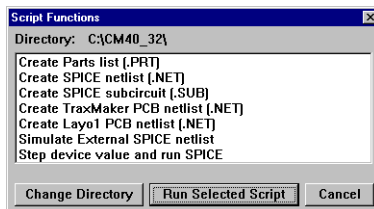
Choose the “Print Circuit...” command to print the current circuit to a printer or to save it to a print file. A Printer dialog box will be displayed, allowing you to select the desired output. If the circuit is too large to fit onto a single page it will automatically be divided into page size blocks as it is printed (see “Show Page Breaks” in the View menu). All printers and printer options which can be selected from the Printer dialog box are supported. Printer setups will vary depending on the device you are printing to, but typically such options as page orientation and reduction factor are available.

Print Waveforms...

Choose the “Print Waveforms...” command to print the waveforms to a printer or to save it to a print file. Only the information currently displayed in the active Waveforms or Analysis window will be printed.

Script Functions...

CircuitLogix has the ability to execute special script files to perform various tasks. When you select the “Script Functions...” command, the following dialog box will appear. The Change Directory button is provided to allow you to select the scripts from a directory other than the CircuitLogix home directory.



Create Parts list (.PRT)

Run the “Create Parts list (.PRT)” script to create a parts list file and write it to disk. This file contains information about how many and which kinds of parts are used in your circuit.

Example:

```
Parts list for circuit:  
C:\Cm50\Bp.ckt  
-----  
C2,C1: .02uf CAP (2)  
IC1: UA741 (1)  
R1: 39.8k RESISTOR (1)  
RF: 159k RESISTOR (1)  
R2: 401.9 RESISTOR (1)
```

Create SPICE netlist (.NET)

Run the “Create SPICE netlist (.NET)” script to create a SPICE compatible text file describing your circuit and write it to disk. This text file can be imported into other SPICE compatible simulation programs.

Create SPICE subcircuit (.SUB)

Run the “Create SPICE subcircuit (.SUB)” script to create a .SUB file describing your circuit. The text can then be copied into a .SUB file which corresponds to the device symbol you intend to use. Connect SCOPE [Digital/Instrument] (t) devices to your circuit to indicate the connecting nodes of the subcircuit. Label them TP1 (1st node), TP2 (2nd node), etc., or anything that ends with a number (1, 2, 3, etc.) up to a maximum of 64 connecting nodes.

Create PCB netlist (.NET)

Run the “Create PCB netlist (.NET)” script to create a PCB netlist file. This PCB netlist file generated by CircuitLogix can be imported into any Printed Circuit Board layout program capable of importing Tango netlists. To create a PCB netlist, your circuit must meet the following requirements:

1. Each device in the circuit must have a unique Designation (ie. D1, C3, etc.).
2. Each device must have a value or label (ie. 10k, 2N2222A, etc.).
3. Each device must have a Package type which should match one of the available Patterns in your PCB Layout program (ie. DIP14, AXIAL0.4, etc.).
4. Pin numbers must be assigned for each device in the circuit.

If any of these required items is missing, an error message will be shown. See the step-by-step example in Appendix C.

Simulate external SPICE netlist

The “Simulate external SPICE netlist” script allows you to import and simulate a SPICE netlist that was not created in CircuitLogix. The netlist is converted to a compatible-SPICE format and saved in an intermediate .TXT file. SPICE will then run, creating a .RAW file in which the simulation data is stored. A new circuit is opened and the Plot Var device is placed in the circuit. Click on this device to get a list of the variables that can be plotted, then select a variable from the list. The waveform for that variable will be plotted on the graph. To view multiple waveforms, SHIFT-click on the Plot Var device. When you stop the simulation, the circuit will be cleared.

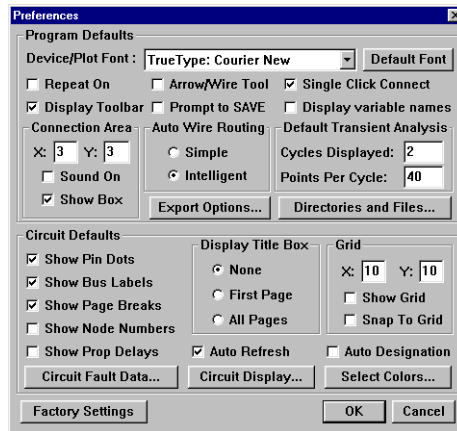
Step device value and run SPICE

The “Step device value and run SPICE” script allows you to automatically run multiple SPICE simulations, stepping the value of a device over a user-definable range. The waveform is stored for each node that has a **Run-Time Test Point** attached. Run-Time Test Points are described in the Analog Simulation chapter. When the simulation runs are complete, all the stored waveforms will be displayed. When you click on a point in the circuit for which data has been collected, the waveforms that correspond to that point—including any stored waveforms—will be displayed. The following device’s values may be stepped: Resistors, Capacitors, Inductors and DC Voltage and Current Sources. When you run this script, you will be presented with a short series of dialog boxes, requesting that you provide the following information.

1. The number of simulation runs.
2. The number of linearized data points to save for each run.
3. The device for which the value is to be stepped.
4. The value for the device on the first simulation run. If the device is a Var Resistor, you must supply the position of the wiper as a percentage of the whole for the first simulation run.
5. The value for the device on the last simulation run. If the device is a Var Resistor, you must supply the position of the wiper as a percentage of the whole for the last simulation run.

Preferences...

Choose the “Preferences...” command to change the default settings of CircuitLogix. Preferences are stored in the CIRLOGIX.DAT file. The following dialog box appears:



The items in the **Program Defaults** section are used no matter what circuit is loaded, while the items in the **Circuit Defaults** section are saved with each individual circuit.

Click on the **Factory Settings** button to reset all the Preferences to their factory default settings.

Device/Plot Font

This option lets you specify the default font used in CircuitLogix. This font is used on all the device labels, wire labels, pin names, key caps, ASCII displays and in the plot windows.

Default Font Button

Sets CircuitLogix default font to its factory default which is “Courier New”. This is a TrueType (scalable/rotatable), monospaced font that comes with the Windows operating system.

Repeat On

See *Repeat On* in the Options Menu chapter.

Arrow/Wire Tool

See *Arrow/Wire* in the Options Menu chapter.

Single Click Connect

This option is used for manual routing of wires. When Single Click is enabled, you can terminate a wire by a single mouse click on any valid connection point. You can still terminate a wire at any location with a double-click, even if the end of the wire is not at a valid connection point. When the Single Click option is disabled, you must double-click to end the wire. This double-click mode lets you turn a wire very close to other wires or pins without connecting to them.

Display Toolbar

See *Display Toolbar* in the Options Menu chapter.

Display variable names

See *Display Variable Names* in the Options Menu chapter.

Prompt to SAVE

If modifications have been made to the circuit, you may want to save the circuit before you run the simulation. If this box is checked, then each time you run the simulation CircuitLogix will warn you if the circuit has not been saved.

Connection Area

The connection area defines a rectangle around valid connec-

tion points (a valid connection point is any device pin or wire). The SmartWires™ feature allows you to connect a wire to a valid connection point, even if your cursor is not exactly on that point. By setting the “X” and “Y” values, you specify how close (within how many pixels) you must be to the connection point before the wire will snap to that point. By default, the cursor must be within 3 pixels of the connection point. When the “Sound On” option is enabled, the system will beep through the PC’s speaker each time the cursor enters a connection area. When the “Show Box” option is enabled, a rectangle will be appear around the connection area each time the cursor enters that area.

Default Transient Analysis

This section allows you to control the parameters supplied to the Transient Analysis when the Default Setups button is pressed in the Analog Analyses dialog box. These values are only used in circuits which contain one or more Signal Generators. “Cycles Displayed” determines the number of cycles of the *lowest* frequency Signal Generator to be analyzed. “Points Per Cycle” determines the resolution of the analysis.

Export Options Button

See *Export Options* earlier in this chapter.

Directories and Files Button

The default directories and files used by CircuitLogix are user definable. “Circuit Directory” is the path to where your circuit (.CKT) files are located. “Model Directory” is the path to where your SPICE model libraries are located. “Script Directory” is the path to where your script (.SRP) files are located. “User Library File” is the path and file name of your User.lib (Macros) file. “Device Library File” is the path and file name of the Device.lib file.

Note: *The Devicedb.dat, Hotkeydb.dat and Symboldb.dat files must be in the same directory as User.lib.*

Show Pin Dots

See *Show Pin Dots* in the View Menu chapter.

Show Bus Labels

See *Show Bus Labels* in the View Menu chapter.

Show Page Breaks

See *Show Page Breaks* in the View Menu chapter.

Show Node Numbers

See *Show Node Numbers* in the View Menu chapter.

Show Prop Delays

See *Show Prop Delays* in the View Menu chapter.

Display Title Box

See *Title Box* in the View Menu chapter.

Grid

See *Drawing Grid* in the View Menu chapter.

Circuit Fault Data Button

See *Circuit Faults* in the Fault Simulation chapter.

Circuit Display Button

See *Circuit Display Data* in the View Menu chapter.

Select Colors Button

See *Select Colors* in the View Menu chapter.

Auto Refresh

See *Auto Refresh* in the Options Menu chapter.

Auto Designation

See *Auto Designation* in the Options Menu chapter.

Exit

Choose the “Exit” command to exit CircuitLogix and return to Windows. If there are unsaved changes, you will be asked if you want to save them.

Chapter 6

Edit Menu

Edit	
Undo Move	Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Move	Shift+INSERT
Delete Items	DELETE
Duplicate	Ctrl+D
Select All	Ctrl+A
Copy Circuit	
Copy Waveforms	
Refresh Screen	Ctrl+F
Rotate 90	Alt+R
Mirror	Alt+M
Straighten Selected Wire(s)	Alt+I
Place Selected Label(s)	Alt+L
Set Prop Delay(s)...	
Set Auto Designation...	Ctrl+F4
Edit Selected Item	▶
Group Selected Items	
Select Font...	



The commands in the Edit menu provide some of the tools necessary to construct and modify circuit diagrams.

Undo

Many of the CircuitLogix editing commands can be undone. Edit operations which can be undone are Cut, Paste, Delete and Move. Only the most recent edit can be undone and many commands will void the ability to undo. Commands which will void the ability to undo include Run, Single Step, Reset, Define New Macro, Expand Macro and Delete Macro.

Cut

The “Cut” command will remove all selected items and place them in the paste buffer and on the system clipboard.

Copy

The “Copy” command will place all selected items in the paste buffer and on the system clipboard.

Paste

The “Paste” command will display the contents of the paste buffer in the work area and the displayed items will follow the mouse around the screen. The cursor is replaced by the paste cursor, indicating the top left corner of the paste area. When the items are positioned at the desired location, click the mouse to finalize the paste. To cancel the paste, press any key or double-click the mouse.

Move

The “Move” command performs both a “Cut” and a “Paste”. This provides a quick way to move an entire circuit or portion of a circuit to a new position on the screen.

Delete Items

The “Delete Items” command removes all selected items from the work area.

Duplicate

Choose the “Duplicate” command to make a copy of all selected items. When this command is selected the duplicated items will follow the mouse around the screen. The cursor is replaced by the paste cursor, indicating the top left corner of the paste area. When the items are positioned at the desired location click the mouse to finalize the duplication. To cancel the duplication press any key or double-click the mouse.

Select All

This command selects all items in the work area. This is useful when you want to cut, copy or move the entire circuit.

Copy Circuit

The “Copy Circuit” command allows you to copy the entire circuit to the system clipboard in one of three formats—Windows Metafile, Device Independent Bitmap or Device Dependent Bitmap—depending on the format selected in the “Export Options...” option in the File menu. This enables you to later paste the contents of the clipboard into a graphics program for further manipulation, printout, etc. When this command has been executed, an alert box will appear informing you that the circuit has been copied to the clipboard.

It should be noted that you cannot paste items into the CircuitLogix work area which were copied using this command. It is designed only to copy the circuit to the system clipboard.

Copy Waveforms

The “Copy Waveforms” command allows you to copy the contents of the Waveforms window to the system clipboard in one of three formats—Windows Metafile, Device Independent Bitmap or Device Dependent Bitmap—depending on the format selected in the “Export Options...” option in the File menu. When this command has been executed, an alert box will be displayed informing you that the waveforms have been copied to the clipboard.

In order to use this command, at least one oscilloscope must be connected in your circuit and the Waveforms window must be open. Only the information currently visible in the Wave-

forms window will be copied to the clipboard. To ensure that meaningful waveform information is copied to the clipboard, you must run the simulation prior to selecting this command.

Refresh Screen

Choose the “Refresh Screen” command to redraw the entire circuit. Redrawing may be desirable following a command or operation which causes parts of the screen to become “dirty”. Also see *Auto Refresh* in the Options Menu chapter.



The Rotate 90° Button



The Mirror Button

Rotate 90

This is the same as the Rotate 90 button described in the Schematic Capture chapter.

Mirror

This is the same as the Mirror button described in the Schematic Capture chapter.

Straighten Selected Wire(s)

When you have wires with multiple bends, sometimes you may want to clean them up a little. This is just a quick way of taking out some of the extra “kinks” without having to adjust the wire manually. All selected wires may be affected.

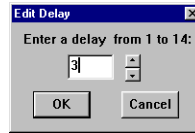
Place Selected Label(s)

This command allows you to place the labels of all selected devices in their standard positions without rotating or mirroring.

Set Prop Delay(s)... (for Digital Simulation Mode)

This command allows you to alter the propagation delay of all selected devices. The delay of a device determines how many simulation ticks it takes for a signal to propagate from the input to the output of the device. The default delay for all devices is one, but by using the “Set Prop Delay(s)...” command this value can be changed to any integer from 1 to 14. The units assigned to delay are arbitrary, and it is left to the user to determine what they are. The concept is that if one device has a delay of one and another a delay of three, then in the real world the second device would have a propagation delay three times larger than the first device.

To invoke this command first select at least one device, then select the “Set Prop Delay(s)...” command from the Edit menu. The following dialog box will then be displayed allowing you to enter a new value for the delay or to cancel the command.



Use the “Show Prop Delays” command in the View menu to display the propagation delay of all devices in the circuit. The delay values will be shown within a rounded rectangle located near the center of each device. Some devices (pulsers, LED’s, macro devices, etc.) do not have a delay, so no value will be shown on them. In the case of macro devices, the delay is determined by the individual delay setting of each device contained within the macro.

Set Auto Designation...

Use this command to automatically generate the designations (R1, C3, Q2, etc.) for each device in the circuit. If any of the devices in the circuit already have a designation, that designation will be replaced. The designation prefix for each device can be set in the Device Data dialog box described in the Netlist Information section of the Schematic Capture chapter. Each device must have a unique designation in order for analog simulation to work.

Edit Selected Item

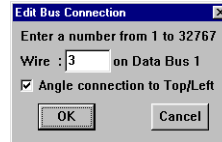
This sub-menu allows the same functions as double-clicking on an item in the circuit. It is included here as a reference to the various functions. Select a single item in the circuit, then select the proper command from this menu. The functions are described below.

Edit Bus Connection...

This feature allows the number associated with an existing wire that is connected to a bus wire to be modified. It also

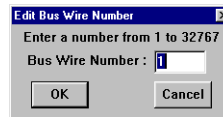


allows the angle on the connecting wire to be selected. Select the wire (connected to a bus) to be edited by clicking on it once, then select the “Edit Bus Connection...” command and enter a new number or angle. Alternately, click on the wire with the Naming Tool or double-click on it with the Arrow Tool and enter a new number or angle.



Edit Bus Wire Number...

This feature allows the number associated with an existing bus wire to be changed. Select the bus wire to be edited by clicking on it once, then select the “Edit Bus Wire Number...” command and type in a new number. Alternately, click on the bus wire with the Naming Tool or double-click on it with the Arrow Tool and enter a new number.



Edit Device Data...

Choose the “Edit Device Data...” command to change device labels, values, SPICE data, etc. Refer to *Netlist Information* in the Schematic Capture chapter for details.

Edit Digital Params...

Choose the “Edit Digital Params...” command to change digital simcode device parameters. Refer to *Digital Simcode Devices* in the Macro Devices chapter for details.

Edit Run-Time Test Point...

Choose the “Edit Run-Time Test Point...” command to change the setting for the selected Run-Time Test Point. Refer to *Run-*

Time Test Points in the Analog/Mixed-Mode Simulation chapter for details.

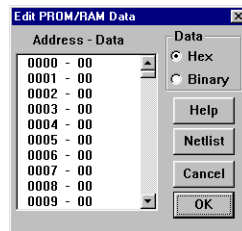
Edit/Select Spice Model...

Choose the “Edit/Select Spice Model...” command to change the SPICE model for the selected device. Refer to *SPICE Models* in the Edit Devices chapter for details.

Edit PROM/RAM...

Use the “Edit PROM/RAM...” command to program the 32 x 8 PROM or the 1k x 8 RAM. Once a PROM is programmed, it retains its contents until reprogrammed. When a circuit containing a PROM is saved to disk the contents of the PROM are also saved. For debugging purposes, the contents of the RAM can be viewed and edited, but the data will not be saved with the circuit.

To invoke this command, click on a single PROM or RAM to select it, then select the “Edit PROM/RAM...” command. The following dialog box will then appear allowing you to enter either hexadecimal or binary numbers.



Edit Pulser...

Choose the “Edit Pulser...” command to change the programmed settings of a Pulser. The pulse format, time high, time low, and trigger mode are individually programmable for each Pulser in the circuit. For a complete description of the Pulser, refer to *The Pulser* in the Digital Logic Simulation chapter.

Edit Multimeter...

This feature allows you to change the function of the selected meter (voltmeter or ammeter). Select the Multimeter to be changed by clicking on it once, then select the “Edit Multim-

eter...” command. Alternately, double-click on it with the Arrow Tool. Refer to *Multimeter* in the Analog/Mixed-Mode Simulation chapter.

Edit Input/Output...

This feature allows the name associated with an INPUT or OUTPUT device to be edited. Select the Input (or Output) device to be modified by clicking on it once, then select the “Edit Input/Output...” command and type in a new name. Alternately, click on the Input (or Output) device with the Naming Tool or double-click on it with the Arrow Tool and enter a new name.



Edit Data Sequencer...

Choose the “Edit Data Sequencer...” command to change the programmed settings of a Data Sequencer. This device allows the user to specify up to 1024 bytes which can be output in a defined sequence. To edit this data, select the Data Sequencer to be modified by clicking on it once, then select the “Edit Data Sequencer...” command (or just double-click on the Data Sequencer with the Arrow Tool). For a complete description of the Data Sequencer, refer to *The Data Sequencer* in the Digital Logic Simulation chapter.

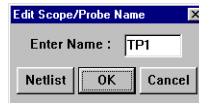
Edit Signal Generator...

Choose the “Edit Signal Generator...” command to change the programmed settings of the selected multifunction signal generator. For a complete description of the signal generator, refer to *Multifunction Signal Generator* in the Analog/Mixed-Mode Simulation chapter.

Edit Scope/Probe Name...

This feature allows an existing SCOPE or Analysis Probe name to be changed. Select the scope or probe to be changed by clicking on it once, then select the “Edit Scope/Probe Name...”

command and type in a new name. Alternately, click on the scope or probe with the Naming Tool or double-click on it with the Arrow Tool and enter a new name.



Group Selected Items

This command is provided to allow the user to specify which devices are contained within the same package. For example, a 7400 Quad 2-Input NAND gate actually has 4 gates in the same package. The pin numbers are different for each gate. The individual gates are grouped together automatically as they are placed in the circuit. The gates can later be regrouped as needed using this command.

Select Font...

Choose the “Select Font...” command to change the font attributes of any selected, free-floating text field, or the default font for new text.

Chapter 7

Fault Simulation

An exciting educational feature of CircuitLogix is the ability to place faulty devices in circuits, allowing an instructor to create troubleshooting exercises for students. An instructor can create a working circuit, then “zap” one or more of the devices by applying shorts, opens or other faults to selected pins on those devices. The instructor can then password protect the fault data and limit the resources available to the student. Faults can be added to both digital and analog circuits.

You should always create a working circuit before adding any fault data. CircuitLogix comprehensive fault simulator has the capability of creating multiple, complex faults. However, discretion should be used when adding faults to a circuit. One or two damaged pins on an IC, a burnt resistor, or a shorted capacitor may be enough to present a challenge for the student.

Device Faults

Each device may have multiple faults, but each device pin may only have a single fault. There are 6 basic types of device faults that can be simulated in CircuitLogix: pin(s) stuck high, pin(s) stuck low, pin(s) open, pins shorted together, wrong values, and user-defined faults.

Pin(s) stuck high. In digital simulation mode the specified pins will be connected to a logic high. In analog simulation mode the specified pins will be connected to an invisible, independent voltage source of a user-specified value.

Pin(s) stuck low. In digital simulation mode the specified pins will be connected to a logic low. In analog simulation mode the specified pins will be connected to an invisible, independent voltage source of a user-specified value.

Pin(s) open. In digital simulation mode the specified pins will be disconnected from the device. In analog simulation mode the specified pins will be connected to the device through an invisible resistor of a user-specified value.

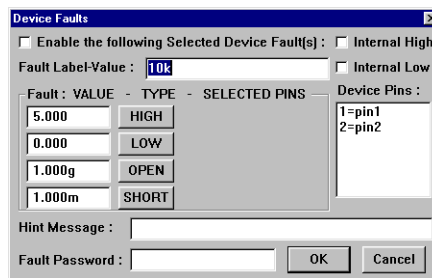
Pins shorted together. In digital simulation mode the specified pins will be shorted directly together. In analog simulation mode the specified pins will be connected together through a daisy-chain of invisible resistors of a user-specified value.

Wrong value. For analog simulation mode only. The Label-Value for the device is replaced during simulation by the Fault Label-Value.

User defined. For analog simulation mode only. The SPICE model or subcircuit specified in the Label-Value for the device is replaced during simulation by the model or subcircuit specified in the Fault Label-Value. The faulty model or subcircuit may be one that has been modified by the instructor to operate incorrectly.

Adding Device Faults

To add faults to a device, double-click on the device to get the Edit Device Data dialog box. Then click on the Faults button to bring up the Device Faults dialog box. Following is a description of the various items.



- **Enable check box:** When checked, the faults specified for this device will be enabled. It is enabled automatically when you edit the fault data.
- **Fault Label-Value:** This field is used for the “wrong value” and “user defined” faults mentioned previously. If a value is entered in this field, it will replace the device’s Label-Value when the simulation is run. Faulty SPICE model and subcircuit names may also be entered here. If the field is left blank, the device’s real Label-Value will be used in the simulation.

- **Faults and Device Pins:** Select one or more pins from the Device Pins list, then click on one of the fault buttons. The selected pins will be removed from the selection list and placed to the right of the associated fault button. Pressing a fault button again will remove that fault from the pins listed next to it, and place the pins back in the selection list.

The HIGH button is used for the “pin(s) stuck high” fault. The value to the left of the button is used in analog simulation mode to indicate the voltage of an invisible voltage source to which the pin is stuck.

The LOW button is used for the “pin(s) stuck low” fault. The value to the left of the button is used in analog simulation mode to indicate the voltage of an invisible voltage source to which the pin is stuck.

The OPEN button is used for the “pin(s) open” fault. The value to the left of the button is used in analog simulation mode to indicate the resistance (very high) between the pin and the device.

The SHORT button is used for the “pins shorted together” fault. The value to the left of the button is used in analog simulation mode to indicate the value of the resistors (very low) that are daisy-chained between each of the selected pins.

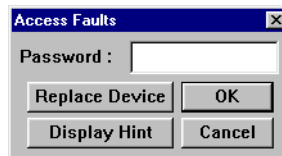
- **Internal High/Low check boxes:** These are used for digital simulation only; they do not affect analog simulation. Stuck high and stuck low faults are assumed, by default, to be external to the device. Due to the nature of the digital simulation, it is desirable to use internal high/low faults on input pins and external high/low faults on output pins. For this reason you should not have an input pin and an output pin both stuck high or both stuck low on the same device.
- **Hint Message:** The instructor may place a brief hint about the nature of the fault in this field that can be accessed by the student while troubleshooting the circuit.

- **Fault Password:** The instructor may enter a password in this field to restrict access to the fault data. If a password is entered here, all fault data will be password protected. If the password is deleted, the fault data will no longer be protected. Passwords are case sensitive and may be up to 15 characters in length. They may include any printable character, excluding TAB or ENTER. Password protection:
 - restricts access to the Device Faults dialog box.
 - restricts access to the Circuit Faults dialog box.
 - eliminates fault comments from the .NET file.

Access Faults

If the fault data has been password protected, the Access Faults dialog box will be displayed whenever the student:

- a. clicks on the “Circuit Fault Data...” button in the Preferences dialog box, or
- b. clicks on the Faults button in the Edit Device Data dialog box, or
- c. double-clicks on a device (if “Device Data Display” has been checked in the Circuit Faults dialog box).



If “Replace Device” has not been disabled (in the Circuit Faults dialog box), the student can click on this button to replace a device. This disables the fault data for the device, but does not actually delete the fault data.

If “Replacement Status” has not been disabled (in the Circuit Faults dialog box), a message will then be displayed indicating whether the replaced device was good or faulty. “Replacement Status” can be disabled from the Circuit Faults dialog box.

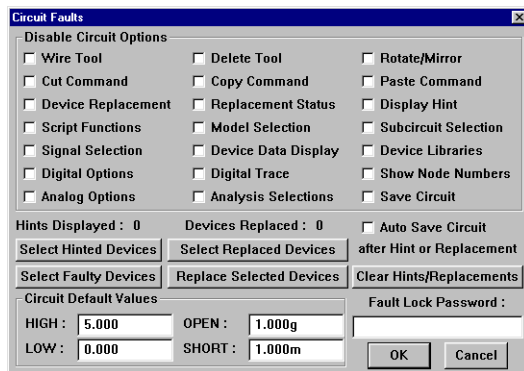
If “Display Hint” has not been disabled (in the Circuit Faults dialog box), the student can click on the Display Hint button to view any hint the instructor has provided for that device. “Display Hint” can be disabled from the Circuit Faults dialog box.

The instructor can see how many devices were replaced, and how many hints were displayed, all from the Circuit Faults dialog box. The instructor can also enter the password to gain access to the Device Faults or Circuit Faults dialog box.

Circuit Faults

When faults are enabled, the instructor can control which resources are available to the students. The Circuit Faults dialog box is used for this purpose, as well as defining the default analog fault values. The settings specified in this dialog box are saved with the circuit.

From the Preferences dialog box, click on the Circuit Fault Data button to bring up this dialog box:



Disable Circuit Options

This section is used to limit the features available to the student. **Only the items which are not checked** are available to the student. By default, none of the items are checked, meaning that the student has access to all CircuitLogix features.

Wire Tool. With this box checked the student will be unable to use the Wire Tool or the Arrow/Wire Tool to modify the circuit.

Delete Tool. With this box checked the student will be unable to use the Delete Tool or the Delete key to modify the circuit. This does not, however, prevent the student from replacing a device (by clicking on the Replace Device button in the Access Faults dialog box).

Rotate/Mirror. With this box checked the student will be unable to rotate or mirror devices in the circuit.

Cut Command. With this box checked the student will be unable to use the Cut and Move commands to modify the circuit.

Copy Command. With this box checked the student will be unable to use the Copy, Duplicate and Copy Circuit commands. *This prevents the student from copying faulty devices into another circuit which is not password protected.*

Paste Command. With this box checked the student will be unable to use the Paste and Duplicate commands to modify the circuit.

Device Replacement. With this box checked the student will be unable to use the Replace Device button in the Access Faults dialog box. This does *not* prevent the student from deleting the device or selecting a new device from the device library.

Replacement Status. With this box checked, no message is displayed when a device is replaced. The message would indicate whether or not the replaced device was faulty.

Display Hints. With this box checked the student will be unable to view any hint messages.

Script Functions. With this box checked the student will be unable to run any of the script functions.

Model Selection. With this box checked the student will be unable to select a new SPICE model for any device.

Subcircuit Selection. With this box checked the student will be unable to select a new SPICE subcircuit for any device.

Signal Selection. With this box checked the student will be unable to change the settings of the Signal Generators.

Device Data Display. With this box checked the student will be unable to view or edit the device data in the Edit Device Data dialog box, and will be unable to change the Visible status of the device labels and values.

Device Libraries. With this box checked the student will be unable to select new devices from the device library. This does not prevent the student from replacing a device by clicking on the Replace Device button in the Access Faults dialog box.

Digital Options. With this box checked the student will be unable to access the Digital Options dialog box.

Digital Trace. With this box checked the student will be unable to use the Trace feature. The Trace feature is used in digital simulation to view the logic states of all the wires in the circuit.

Show Node Numbers. With this box checked the student will be unable to view the node numbers for the circuit. Viewing the node numbers could help identify opens and shorts in a circuit.

Analog Options. With this box checked the student will be unable to access the Analog Options dialog box.

Analysis Selections. With this box checked the student will be unable to change the enabled/disabled status of the analog analyses. This limits the student to analysis types specified by the instructor. The student will, however, be able to change the settings of the enabled analyses and run a modified simulation.

Save Circuit. With this box checked the student will be unable to save the circuit. This cannot be checked if “Auto Save Circuit after Hint or Replacement” is checked (see Hints and Replacements below).

Hints and Replacements

Each time the student presses the Replace Device or Display

Hint button in the Access Faults dialog box a flag is set for the selected device. If “Auto Save Circuit after Hint or Replacement” is checked, the circuit will be saved automatically each time they are pressed, preserving the student’s work. The instructor can view the number of devices that were replaced and the number of hints that were viewed by the student. Buttons are provided to let the instructor see (highlight) all of the devices that were replaced, all of the devices for which the student viewed a hint, or all of the devices that have fault data in them (whether the faults are enabled for that device or not). The Replace Selected Devices button disables the fault data for all of the selected devices; it does not delete the fault data. The Clear Hints/Replacements button clears the flags.

Circuit Default Values

This section of the Circuit Faults dialog box is used to set the default values that correspond to the various faults in analog simulation mode. These defaults can be overridden for each device individually.

- | | |
|--------------|---|
| HIGH | This is the default value for the invisible voltage source to which the specified pins of a device are connected when a HIGH fault is specified. The factory default is +5V. |
| LOW | This is the default value for the invisible voltage source to which the specified pins of a device are connected when a LOW fault is specified. The factory default is 0V. |
| OPEN | This is the default value for the invisible resistor placed between the specified pins and the device. The factory default is 1G ohm. |
| SHORT | This is the default value for the invisible resistors that are daisy-chained between the specified pins of the device. The factory default is 1m ohm. |

Fault Lock Password

The instructor may enter a password in this field to restrict access to the fault data. If a password is entered here, all fault data will be password protected. If the password is deleted, the fault data will no longer be protected. Passwords are case sensitive and may be up to 15 characters in length. They may

include any printable character, excluding TAB or ENTER.

Warning: If you forget the password, you will be unable to access the fault data for the circuit.

“Black Box” Macros

The instructor may lock a macro so that it cannot be expanded by the students, thus creating a “black box”. To do this, expand the macro, select “Macro Lock” from the Macros menu, enter any 4-digit number (1-9999), then save the macro. When you attempt to expand the macro again, you will be prompted to enter the 4-digit code. A code of 0 (zero) leaves the macro unlocked.

Warning: If you forget the code, you will be unable to expand the macro.

Fault Example

1. Open the example circuit PS1.CKT.
2. Press the Run button in the toolbar to run the simulation.
3. Select the Transient Analysis window, then probe around on the circuit to verify that the circuit is working properly. If you click on the positive side of the filter capacitor (C1), you should see a ripple on waveform at about 15VDC. If you click on the emitter side of the transistor (Q1), you should see a DC voltage of about 6V.
4. Stop the simulation by clicking on the Stop button in the toolbar.
5. Double-click on the filter capacitor (C1) to bring up the Edit Device Data dialog box. Click on the Faults button. Click on OK to exit the Device Faults dialog box, then click on OK to exit the Edit Device Data dialog box.
10. Select “Preferences...” from the File menu to bring up the Preferences dialog box. Click on the Circuit Fault Data button to bring up the Circuit Faults dialog box.
11. Click on Device Data Display so it is checked.
12. Click on the Clear Hints/Replacements button to clear these flags.

13. Type “xxx” into the Fault Lock Password field.
14. Click on OK to exit the Circuit Faults dialog box, then click on OK to exit the Preferences dialog box.
15. If the fault settings for the circuit are to be saved (not required for this example), you must now save the circuit.
16. Press the Run button in the toolbar to run the simulation.
17. Look at the waveform on the positive side of the filter capacitor. You should see a sine wave of very low amplitude (about 150mV peak-to-peak). The student might determine from this measurement that the filter capacitor is shorted and want to replace it.
18. Stop the simulation by clicking on the Stop button in the toolbar.
19. Double-click on the filter capacitor (C1) to bring up the Access Faults dialog box. Click on the Display Hint button to view the hint.
20. Double-click on the filter capacitor again to bring up the Access Faults dialog box. Click on the Replace Device button to disable the fault data for this device.
21. Run the simulation again to see if the problem has been fixed.
22. Stop the simulation and select “Preferences...” from the File menu to bring up the Preferences dialog box. Click on the Circuit Fault Data button to bring up the Access Faults dialog box.
23. Type “xxx” into the Password field and click OK to bring up the Circuit Faults dialog box.
24. Note that there was one device replaced and one hint displayed.
25. Click on the Device Data Display check box to remove the check. Delete the “xxx” from the Fault Lock Password field.
26. Click on the Select Replaced Devices button.
27. Click on OK to exit the Circuit Faults dialog box, then click on OK to exit the Preferences dialog box. The device that was replaced (the filter cap) will be selected.

Chapter 8

Device Editor

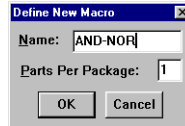
CircuitLogix is fully expandable because it allows you to create your own completely functional devices and nonfunctional device symbols. You will see the term “macro device” used loosely throughout this manual to refer to any device created by or whose functionality can be modified by the user. New devices can be created in the following manner:

1. A macro device can be created. This is a device which contains internal circuitry, built using CircuitLogix and its existing devices, which performs the desired function of the new device. The symbol of a new macro device is created using the Symbol Editor. Macro devices can be nested, meaning that a macro can be used within another macro, thus allowing the user to create building blocks and piece them together in order to create the final device. When a macro is saved, all the circuitry is hidden within its package. The new macro can then be used in any circuit and functions according to the circuitry hidden within. Refer to the section *Creating a New Macro* later in this chapter for a step-by-step tutorial about creating macro devices.

2. SPICE model and subcircuit information can be attached to a new or existing device. For new devices, a symbol is created using the Symbol Editor. When using this method for creating new devices it is helpful if the user is familiar with SPICE, an industry standard program for simulating circuits. For those interested in learning more about SPICE, a short list of reference books is listed in Appendix A. With this method, a text editor is used to create or modify an ASCII file containing the SPICE information. For more information, refer to the section *SPICE Models and Subcircuits* later in this chapter.

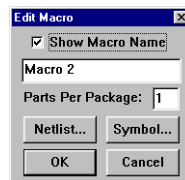
The Macro Button:

The Macro button in the Toolbar is used to create or edit a macro device. To create a new macro, just click on the button and the following dialog box will appear. This is the same as selecting “Define New Macro...” from the Macros menu.



Enter a unique name for the macro of 13 characters or less and specify how many of these parts would be found in a single IC package. The Macro Name is used to identify the macro device in the library. When you click on the OK button, the Symbol Editor will appear, allowing you to create a symbol for the new macro.

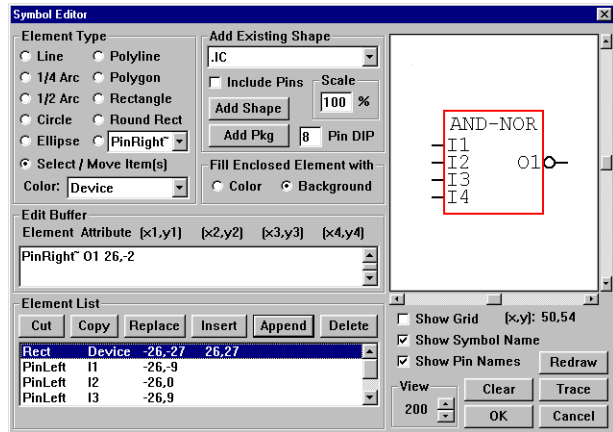
To edit an existing macro, select the macro to be edited by clicking on it once with the Arrow Tool, then click on the Macro button. This is the same as selecting “Expand Macro” from the Macros menu. It will clear the work area and expand the macro. At this point, internal circuitry may be added or edited, or you may double-click on the macro symbol to edit the attributes of the macro package itself. When you double-click on the macro symbol, the following dialog box will appear.



From this dialog box, the macro name may be changed (when you save the macro, it will be saved under the new name, but will not delete the old one). You can also change the number of these parts that would be found in a single IC package. The Netlist button takes you to the Edit Device Data dialog. The Symbol button takes you into the Symbol Editor, allowing you to edit the schematic symbol of the expanded macro.

The Symbol Editor

New macro packages (schematic symbols) are created using the Symbol Editor. The Symbol Editor is invoked as described above (see *The Macro Button*).



The symbol name, if enabled, will only be displayed when there are other elements in the symbol. You can click and drag the name at any time to position it where desired, or you can disable it. The View selection allows you to zoom in or out (from 25%-800%) to better view the device while you are working on it. You can also turn on the drawing grid. The size of the drawing grid is controlled by the Grid Setup dialog box (described in the View Menu chapter). The Snap-To-Grid feature is not used in the Symbol Editor. The grid color can be changed in the Select Colors dialog box available in the View menu.

Drawing with the Mouse

Symbols or parts of symbols may be drawn freehand with the mouse. Select an element type by clicking on its radio button. The following element types are defined: Line, 1/4 Arc, 1/2 Arc, Circle, Ellipse, Polyline, Polygon, Rectangle, Round Rect, and Pins. Pins can point up, down, left or right and can have bubbles to indicate negative logic. Pins with bubbles are indicated with the tilde (~).

Before drawing an element, select the color and fill. Any enclosed element will be filled with either the element color or the background color.

When an element has been placed, its description is placed at the end of the Element List and the element itself remains highlighted. The editor will remain in drawing mode until you click on the Select/Move Item(s) radio button.

Once an element has been placed, it cannot be edited (re-shaped) with the mouse, but it may be moved with the mouse. All elements that are highlighted will be moved when you click and drag any one of them with the mouse. You can select multiple elements by holding down the SHIFT key while clicking on the elements. To delete or make changes to an existing element, refer to *The Element List and the Edit Buffer* later in this chapter.

Adding an Existing Shape

Device symbols which are already in the library can be added to the new symbol, thus making it easy to base a new symbol on one which already exists. Select the device by name in the drop-down list box. If you want to include that device's pins in the new symbol, check the Include Pins check box. If you want to scale the incoming device (make it larger or smaller), specify the scale. If pins are included, they will not be scaled.

To import a metafile device from an external drawing program, place the metafile object on the clipboard. Select "Clipboard-WMF" as the existing shape, then click on the Add Shape button. The metafile will be converted to CircuitLogix device format.

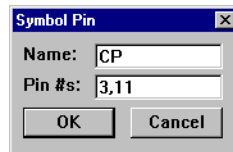
Click on the Add Shape button to bring the device into the drawing window. Press the R key on the keyboard (or click the RIGHT mouse button) to rotate the incoming device. Press the M key on the keyboard to mirror the incoming device. Click the LEFT mouse button to place the device. Once the device is placed, it will remain highlighted allowing you to easily move or delete it. To delete or make changes to a device that has been placed, refer to *The Element List and the Edit Buffer* later in this chapter.

Adding a DIP Package

Dual In-line Packages (DIPs) can be added quickly in the Symbol Editor. Just specify the number of pins—in 2 pin increments—then click on the Add Pkg button. DIPs cannot be scaled directly, but they can be rotated and mirrored.

Editing Pin Information

To edit pin names and numbers, double-click on the pin. The following dialog box will appear.



Each pin should be given a name (up to 16 characters) and a number, even if the names are not displayed. Pin names will help you to verify that any links to SPICE models will be listed in the correct order. For multipart packages, one pin number should be listed for each part in the package, separated by commas. For compatibility with PCB design software, each pin number should match the Pad Designation of the corresponding pad in the PCB software.

The Element List and the Edit Buffer

As elements are added to the drawing, they are also added to the Element List. The Element List is a list of all of the elements in the drawing, displayed in text format. The link between the drawing window and the Element List is interactive. When you select an element in the drawing window, it is highlighted in the Element List. Likewise, when you select an element in the Element List, it is highlighted in the drawing window.

Elements can only be edited in the Edit Buffer. The following buttons are available to help in editing. They also allow you to change the order of the elements. Elements at the beginning of the list are drawn first so they are in the back of the drawing (bottom layer); those at the end of the list are drawn last so they will in front (top layer).

Cut	Removes the selected elements from the Element List and places them in the Edit Buffer.
Copy	Places a copy of the selected elements in the Edit Buffer.
Replace	Replaces the selected elements with the contents of the Edit Buffer.
Insert	Inserts the contents of the Edit Buffer immediately before the selected element in the Element List.
Append	Appends the contents of the Edit Buffer to the end of the Element List.
Delete	Removes the selected elements from the Element List.

When editing an element, strict rules of syntax must be observed. The definition for each element must state the element type, attribute (line/fill color or pin name), and a set of xy coordinates. A more complete description of the syntax required for each element is given under *Element Definitions* later in this chapter.

The Trace Feature

The Trace button allows you to step through the Element List, highlighting each element one at a time. Stepping will begin at the currently selected element. To see how the symbol is drawn from the beginning, element by element, select the first element in the list, click on the Clear button to erase the drawing window, then click on the Trace button repeatedly. This will help you identify elements that are hidden behind other elements, etc.

Element Definitions

Element: **Line**
 Attribute: pen color
 x1,y1: start point of the line
 x2,y2: end point of the line
 x3,y3: n/a
 x4,y4: n/a

Element: **Polyline**
 Attribute: pen color
 x1,y1: 1st point of the polyline

x2,y2: 2nd point of the polyline (optional if +Polyline)
x3,y3: 3rd point of the polyline (optional)
x4,y4: 4th point of the polyline (optional)
Note: A preceding plus sign (+Polyline) indicates that this is an extension to the preceding polyline element.

Element: **Polygon**
Attribute: pen color*
x1,y1: 1st point of the polygon
x2,y2: 2nd point of the polygon (optional if +Polygon)
x3,y3: 3rd point of the polygon (optional if +Polygon)
x4,y4: 4th point of the polygon (optional)
Note: A preceding plus sign (+Polygon) indicates that this is an extension to the preceding polygon element.

Element: **Rect**
Attribute: pen color*
x1,y1: left top corner of the rectangle
x2,y2: right bottom corner of the rectangle
x3,y3: n/a
x4,y4: n/a

Element: **RRect**
Attribute: pen color*
x1,y1: left top corner of the rounded rectangle
x2,y2: right bottom corner of the rounded rectangle
x3,y3: width and height of ellipse defining the corners
x4,y4: n/a

Element: **Ellipse**
Attribute: pen color*
x1,y1: left top corner of the defining rectangle
x2,y2: right bottom corner of the defining rectangle
x3,y3: n/a
x4,y4: n/a

Element: **Arc**
Attribute: pen color
x1,y1: left top corner of the rectangle defining the complete ellipse
x2,y2: right bottom corner of the rectangle defining the complete ellipse
x3,y3: end point of line 1 whose start point is at the center of the ellipse
x4,y4: end point of line 2 whose start point is at the center of the ellipse
Note: The arc follows the outline of the ellipse and is drawn counterclockwise from line 1 to line 2.

Element:	PinUp, PinDown, PinLeft, PinRight PinUp~, PinDown~, PinLeft~, PinRight~
Attribute:	pin name (16 characters max.)
x1,y1:	point where pin attaches to package
x2,y2:	n/a
x3,y3:	n/a
x4,y4:	n/a
Note:	A pin is a connection point for wires. Pins can have bubbles to indicate negative logic. Pins with bubbles are indicated with the tilde (~).

* When the pen color on an enclosed element is immediately followed by an asterisk (for example, LtBlue*), the fill color will be the same as the pen color, otherwise the fill color will be the background color.

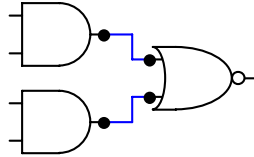
When the Symbol is Complete

When you have completed drawing the symbol, click on the OK button. You will return to the expanded macro where circuitry may be added or edited before the new device is saved. In the following section is a step-by-step procedure for creating a new macro device.

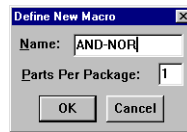
Creating a New Macro

Following is a step-by-step example of how to use the Symbol Editor to create your own macro devices.

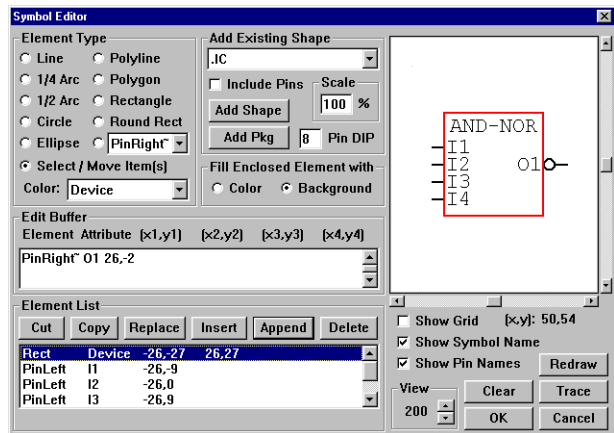
1. It is a good idea to make a backup copy of the USER.LIB file prior to creating or deleting a macro. That way, you will always have a copy of the library to revert to in case the library is damaged or altered in any undesirable way.
2. If your macro is to be nonfunctional or is to contain only SPICE data for analog simulation, skip to the next step. Otherwise, construct a circuit which performs the function which your new macro device is to have. For example, if you wanted to build a device which performs an AND-NOR function you would construct something like the circuit shown below.



3. Select the “Define New Macro...” command from the Macros menu (or click on the Macro button in the Toolbar). The following dialog box will be displayed.

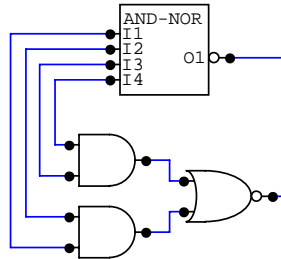


4. Enter a unique name up to 13 characters in length in the Macro Name text edit field. The name you enter here will be used later to select the device from the library. In this example, we will use the name “AND-NOR”.
5. Specify the number of parts per package. This refers to the number of these devices that are found in the same chip. For example, a 7400 package contains 4 NAND gates and a 556 package contains 2 timers. In this example, we will use the default of 1 part per package. Click on the OK button to invoke the Symbol Editor.

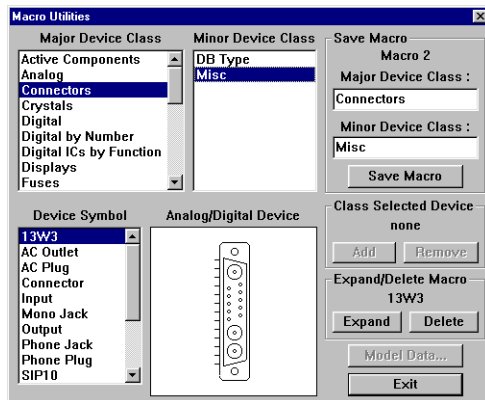


6. For this example, we will begin by placing an 8 Pin DIP package in the drawing window. Type “8” in the Pin DIP field, then click on the Add Pkg button. Click the left mouse button to place the package in the center of the drawing window.
7. The macro name “AND-NOR” will be displayed in the center of the drawing window. Click and drag this name into position near the top of the DIP.
8. Change the View to 200% by clicking on the up arrow next to the View field.
9. This device consists of 1 Rect, 4 PinLefts and 4 PinRights. We will remove 3 of the 4 PinRights. Click on the top right-hand pin in the drawing to select it, then click on the Delete button. Repeat for the 2 bottom right-hand pins.
10. Click on the remaining right-hand pin to select it. Click on the Copy button to copy the pin from the Element List to the Edit Buffer. Change “PinRight” to “PinRight~” and change “P7” to “O1”. Click on the Replace button to replace the selected pin with the new pin. The pin in the drawing will now have a bubble on it.
11. Click on the Show Pin Names check box to enable it.
12. Click on the top left-hand pin to select it. Click on the Copy button to copy the pin from the Element List to the Edit Buffer. Change “P1” to “I1”. Click on the Replace button to replace the selected pin with the new pin. Repeat for each remaining pin, naming them “I2”, “I3” and “I4”.
13. Click on the OK button. The new symbol you created will be displayed and will follow the mouse around the work area. Click the mouse to place the symbol in the work area.
14. Select the Wire Tool from the Toolbar and connect wires from the macro package pins to points within your circuit

from which you want to be able to connect later. Shown below is an example of a completely wired macro device.



15. If your macro device is to have any default data associated with it, the data should be added at this time. To do this, double-click on the macro package, then click on the Netlist... button. Any data entered in the Device Data dialog box at this time will be there every time this macro is selected from the library menus. Now is a good time to add the default Package, Auto Designation Prefix, Spice Prefix Character(s) and Spice Data if required.
16. If you are satisfied with your macro select the “Macro Utilities...” option from the Macros menu. The Macro Utilities dialog box will then be presented.



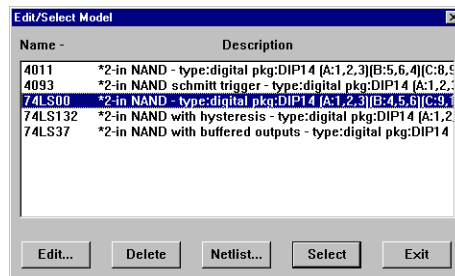
You must specify the Major and Minor Device Class under which your macro will be shown in the Device Selection dialog box. To place the device in an existing Major or Minor Device Class, click on the appropriate items in the lists. To create a new Major or Minor Device Class, type in the Major or Minor Device Class name in

the appropriate text edit fields. Finally, click on the Save Macro button. This will cause the new macro to be saved in the file USER.LIB and the work area will be cleared. To use the new macro simply select it from the device library and use it just as you would any other device.

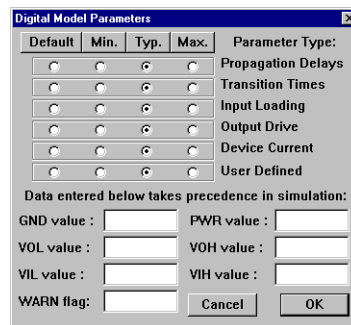
Digital Simcode Devices

Digital simcode devices are mixed-mode digital devices which can be simulated in analog mode using features of XSpice. These devices use event-driven behavioral models created by Logic Design Inc. New simcode models cannot be added by the user, but simcode devices can be used in macro circuits to create new mixed-mode digital devices.

When you double-click on a device that has digital simcode models associated with it, the following dialog box will appear.



While you cannot edit the model itself, you can alter certain parameters of the model. To do so, click on it with the mouse, then click on the Edit button. The following dialog box will appear.



The radio buttons allow you specify whether each of the parameters of the model will be minimum, typical or maximum values for the selected device(s). Refer to *Edit Selected Items* in the Edit menu for information on changing the parameters of multiple devices. The Default setting allows the parameter to remain unchanged. All parameters can vary from pin to pin, part to part and family to family. The parameters are:

- Propagation Delays: The time it takes for a signal change on an input to affect the data on the output.
- Transition Times: The rise and fall times of the outputs.
- Input Loading: The amount of load resistance that will be applied to the output of the driving device.
- Output Drive: The amount of output current available.
- Device Current: The amount of current drawn through the supply pin to ground.
- User Defined: This parameter does not affect digital models provided with CircuitLogix.

The edit fields allow specific values to be entered for certain parameters. These values take precedence over family specific values. Under normal conditions they should be left blank. The parameters are:

- GND & PWR: These two parameters must be programmed as a pair (if you set one, you must also set the other). Setting these voltages will override any other power and ground voltages specified for the selected devices.

VOL & VOH:	These two parameters will override the family defaults.
VIL & VIH:	These two parameters will override the family defaults.
WARN Flag:	When set to nonzero, warning messages will be generated when there are timing or supply voltage violations on the device.

SPICE Models and Subcircuits

There are 3 basic types of components in SPICE: 1) **elementary** components such as resistors, capacitors, power sources, etc., 2) **models** defining single discrete devices such as BJTs, J-FETs, MOSFETs, etc., and 3) **subcircuits** which combine multiple items—elementary components, models, and other subcircuits—to create a single, more complex device. In order to make these devices work in analog simulation, there must be SPICE data available for each device. The model and subcircuit data is stored in ASCII text files, typically identified by a .LIB extension.

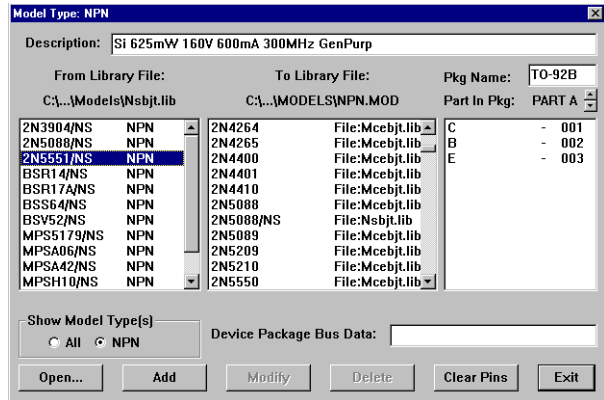
Note: If a macro device contains internal circuitry as well as SPICE data, the SPICE data will be ignored.

CircuitLogix also allows you to add your own SPICE model libraries. SPICE models are available from many sources. Many component manufacturers provide SPICE models for their products upon request. They can also be found on the internet, in engineering magazines, and other sources which are available to the general public.

Model and Subcircuit Linking Files

Model and Subcircuit data is stored in ASCII text files, typically with the .LIB extension. These files can be edited directly with any ASCII text editor.

Each analog device symbol in CircuitLogix has a linking file associated with it. In order to use SPICE models in CircuitLogix, they must be linked to the corresponding device symbol by placing a reference to the model in the linking file. To create a new link or to modify an existing link, select Macro Utilities, from the Macros menu. Locate the symbol that you want to use for the specific model or subcircuit that you are adding, then click on the Model Data button. The following dialog box will be displayed.



Initially, only the center list box will have anything in it. This is a list of the devices in the linking file associated with the selected symbol. If you click on one of the models to select it, the description, package and pinout information will be displayed. To modify this information, just type in the desired changes and click on the Modify button. To remove a reference from the linking file, click on the model to select it, then click on the Delete button.

To add new references to the linking file, click on the Open button and open the library file in which the actual model or subcircuit resides. All of the model and subcircuit names found in this library will be displayed in the list box on the left. If desired, the list can be limited to only those devices which are compatible with the selected device symbol. Enter appropriate description, package and pin information for the model. If there are multiple parts per package, be sure to enter pin numbers for each of PART A, PART B, etc. If the model is similar to another part that is already in the linking file, it may be simpler to first select the similar part to fill in the blanks, make any necessary corrections, then select the new model. Click on the Add button to add the new reference.

The names of the linking files correspond to the device symbol names, with the following exceptions. Linking file names are limited to 8 characters, so the device symbol name is truncated to 8 characters when searching for a match. All spaces and punctuation characters are removed. If the last 1 or 2 characters of a truncated device symbol name are numbers, those

numbers replace the last characters in the file name. If a device symbol name includes a colon (:), the colon and all characters after the colon are ignored. For example, the devices “NPN Trans”, “NPN Trans:A”, “NPN Trans:B” and “NPN Trans:C” are all associated with the linking file “NPNTRANS.MOD”.

If CircuitLogix cannot find a linking file using the file name formed from the device symbol name, and if that symbol can be simulated using simple SPICE models, a default file name is used instead. The default file name is based on the SPICE Prefix Character(s). Following is a list of the SPICE Prefix Character(s) and their corresponding default file names:

C	CAP .MOD	QN	NPN .MOD
D	DIODE .MOD	QP	PNP .MOD
DZ	ZENER .MOD	R	RESISTOR .MOD
JN	NJFET .MOD	S	SW .MOD
JP	PJFET .MOD	U	URC .MOD
MN	NMOS .MOD	W	CSW .MOD
MP	PMOS .MOD	ZN	NMESFET .MOD
O	LTRA .MOD	ZP	PMESFET .MOD

All others, including subcircuit-only device symbols, have no default file name. Subcircuit-only device symbols (symbols that cannot be simulated with a simple model) must use linking files with the .SUB extension.

Transistors are in fact an example of when a default file name is used. Since the file “NPNTRANS.MOD” does not exist, CircuitLogix looks in the default file NPN.MOD for the model reference information.

.MOD linking files contain references to a collection of SPICE models and subcircuits specific to a particular discrete device symbol in CircuitLogix. In some cases, it is desirable to replace a model with a subcircuit which will more accurately model a particular device. For example, some RF or Power Transistors are not modeled well by a simple SPICE model. In such cases, it is OK to reference the subcircuit in the .MOD file just like it was a model. For more information on using models, see *SPICE Models* later in this chapter.

.SUB linking files contain references to a collection of SPICE subcircuits specific to a particular device symbol in CircuitLogix. For more information on using subcircuits, see *SPICE Subcircuits* later in this chapter.

In most cases, the linking files are not edited directly. The file format is described below for information only.

The general format for a reference in a linking file is:

```
*Device Description pkg:PACKAGE [DVCC=14;DGND=7;] 1,2,3,...  
.PARAM QXXXXXXXXX File:Filename.lib
```

where PACKAGE [DVCC=14;DGND=7;] 1,2,3,... is the component name and pad designations in PCB design software (the bus data listed here is for digital component symbols that do not have external power and ground pins), Q is the Spice prefix character for the specific device type, XXXXXXXX is the model name and Filename.lib is the name of the library file in which the model or subcircuit actually resides.

There should be an appropriate description of the device on the line *immediately before* each .PARAM line. This line will be displayed as the description in the Model or Subcircuit Selections dialog box. The first character of this description line must be an asterisk (*). When you double-click on a device to select a model, subcircuits that are found in the .MOD file will also be displayed, but the first character in the description will appear as an “x” instead of a “p”. You should also include appropriate package and pin information at the end of the description line. This information is required when creating a PCB netlist. The pin information indicates the pinout of the device in the package. For example:

```
*500mW 40V 800mA pkg:TO-18 3,2,1  
.PARAM Q2N2222A File:Mcebjts.lib
```

Note that the pin order is specified as 3,2,1. SPICE defines the pin order of a BJT as collector, base, emitter (see *Bipolar Junction Transistors (BJTs)* in the Analog Simulation chapter). In a TO-18 package containing a 2N2222A transistor, the emitter is connected to pin 1 (next to the tab), the base is connected to pin 2 and the collector is connected to pin 3.

CircuitLogix identifies pins by the order of placement in a device symbol. This information can be used to create a netlist linking one program to the other.

If the same device is available in different packages, you may specify additional packages using an alias. Notice the SPICE prefix character is replaced with the letter “A” and the file reference is removed:

```
*350mW 40V 800mA alias:Q2N2222A pkg:TO-92B  
1, 2, 3  
.PARAM APN2222A
```

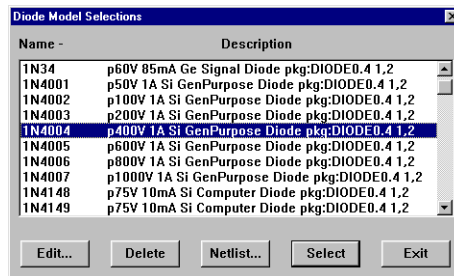
If there are multiple devices in the same package, the pins need to be specified for each device with a letter to indicate the designation extension:

```
*Dual Op Amp pkg:DIP8  
(A:3, 2, 8, 4, 1) (B:5, 6, 8, 4, 7)  
.PARAM XMC1458 File:Mcemods.lib
```

These two methods may be combined if you have multiple devices that are available in different packages.

SPICE Models

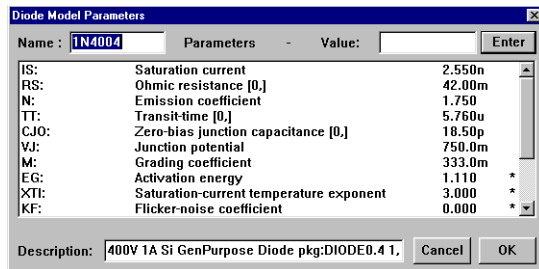
When you double-click on a device that has SPICE models associated with it, the following dialog box will appear.



The currently selected model will be highlighted in the list. To select a different model, click on it with the mouse, then click on the Select button (or just double-click on the model). If any subcircuits are found in the .MOD file, they are indicated by an “x” as the first character in the description instead of a “p”.

To bring up the Edit Device Data dialog box discussed in the *Schematic Capture* chapter, click on the Netlist button.

To edit an existing model, click on it with the mouse, then click on the Edit button. The following dialog box will appear.



The model parameters displayed are the SPICE model parameters and must not be confused with actual data book parameters. Unless you are familiar with SPICE modeling, it is recommended that you do not attempt to modify the existing SPICE models.

The values listed for each parameter represent values defined for that specific device type. If the value is followed by and asterisk (*), that value is a default value, representative of the DEFAULT model, *not necessarily the correct value for this specific device*. To change the value of a SPICE parameter, select the parameter and type the new value into the Value edit field and press Enter. If you want to set a specific parameter to equal that of the DEFAULT model, type an asterisk in the Value edit field and press Enter.

To save the model, click on the OK button. To save it under a new name, type the new name into the Name field, then click on the OK button. *Note: Models created in this manner are stored directly in the .MOD linking file in place of the model reference.* The original .LIB file remains unchanged. When viewed in the Model Selections dialog box, the first character of the device description will be an asterisk (*).

To remove an existing model, select it by clicking on it with the mouse, then click on the Delete button. This will only remove the model reference from the linking file; it will not remove the actual model from the library.

Adding New Models to an Existing Symbol

When you obtain new models from an outside source, they are generally provided in a single ASCII library file. *If you edit this file with a word processor, be sure to save the file in TEXT ONLY format.* Use the Model Data button in the Macro Utilities dialog box to create a link to each model in the new library file.

Example: Add a 2N5209 transistor to CircuitLogix.

1. Select Macro Utilities from the Macros menu.
2. Locate the symbol for an NPN transistor, then click on the Model Data button.
3. Click on the Open button and open the library file (Mcebjt.lib) in which the 2N5209 model resides.
4. All of the model and subcircuit names found in this library will be displayed in the list box on the left. Click on the 2N5209.
5. Enter appropriate information about the model in the Description field (for example: Si 625mW 50V 50mA 30MHz Amp).
6. In the Pkg Name field enter TO-92B to match the name of the component pattern in PCB design software.
7. Enter pin numbers to match the pad designations of the component. On the TO-92B, with the flat side facing you, pin 1 is on the right, pin 2 is in the middle and pin 3 is on the left. For the 2N5902, these pins correspond to the collector, base and emitter, respectively (C=1, B=2, E=3).
8. Click on the Add button to add the new reference. The new model can now be selected alphabetically from the list of npn transistors.

Adding existing models to a new macro symbol:

Create a new nonfunctional macro device symbol as described earlier in this chapter. Be sure to place the pins in the same order as they are listed in the syntax for the corresponding

model. For example, if you are creating a new symbol for a diode, you should place the anode pin first, then the cathode pin to correspond with the SPICE definition of a diode. This is not required, but it will be easier to understand when filling in the Spice Data field for the device. In order to link the new symbol to an existing model file, you must name the new symbol appropriately. For example, you may want to name the symbol “Diode:A” to link to the DIODE.MOD file or “Schottky:A” to link to the SCHOTTKY.MOD file.

With the new macro expanded, double-click on the device, then click on the Netlist button to view the Edit Device Data dialog box. Enter the following data:

- Enter an Auto Designation Prefix for the device. For example, Q for transistors, D for Diodes, etc.
- Set the Spice Prefix Character(s) for this device to be representative of the type of device models you are using. For example, an NPN Bipolar Junction Transistor must have the prefix “QN”. Refer to *Netlist Information* in the Schematic Capture chapter for a listing of valid prefix characters.
- Enter an appropriate SPICE instruction for this device into the Spice Data field. Refer to *Netlist Information* in the Schematic Capture chapter for more information or refer to the examples provided with other devices. Click on the OK button.

Save the macro by selecting “Save Macro” in Macros menu.

Example: Create a new device symbol for standard junction diodes.

1. Create a new macro symbol for a diode using the symbol editor described earlier in this chapter. Name the new macro symbol DIODE:A. When placing pins on the new device, place the *first* pin on the anode (N+), the *second* pin on the cathode (N-). They should be placed in this order to match the syntax for the diode model.
2. Expand the new macro symbol (if it is not already expanded) by clicking on it once with the left mouse button, then click on the Macro button in the Toolbar.
3. Double-click on the device symbol, and click on the Netlist button. Enter the following information in the Edit Device Data dialog box of the expanded macro (to set the

defaults for this macro device):

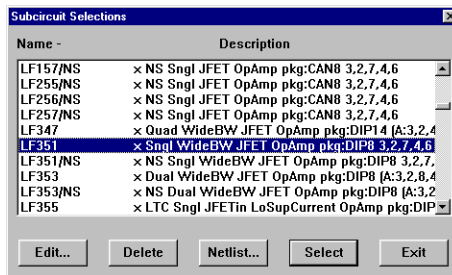
Auto Designation Prefix: D
Spice Prefix Character(s): D
Spice Data field: %D %1 %2 %M

4. Save the macro. See *Save Macro* in the Macros Menu chapter for more information.

SPICE Subcircuits

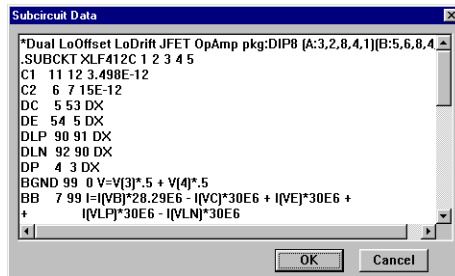
SPICE subcircuits fall into three basic categories, 1) component models, 2) macromodels and 3) equivalent circuits. Component models are basically a complete schematic of the chip, simulated using discrete components. This type of subcircuit is generally more accurate than a macromodel, but requires more time to simulate. The macromodel is more of a block diagram of the chip, where inputs and outputs may be simulated using discrete components, but the internal workings consist of simpler items such as gain blocks, etc. This type of subcircuit simulates rather quickly and in most cases is accurate enough that a component model is not needed. Equivalent circuits may be needed to simulate discrete devices that have no SPICE model. For example, an SCR can be roughly equated to a pair of NPN and PNP transistors coupled together.

When you double-click on a device that has subcircuits in the library the following dialog box will appear.



The currently selected subcircuit will be highlighted in the list. To select a different subcircuit, click on it with the mouse, then click on the Select button (or just double-click on it with the mouse).

To edit or view an existing subcircuit, click on it with the mouse, then click on the Edit button. The following dialog box will appear.



The subcircuit is listed in the edit dialog, beginning with the description line and ending with the .ENDS line. Any changes made to the subcircuit will cause the modified subcircuit to replace the reference in the subcircuit linking file. The original subcircuit will remain unchanged in the library file, but the reference to it will be lost. Naming conventions for .SUB files are discussed in *Model and Subcircuit Linking Files* earlier in this chapter.

Adding New Subcircuits to an Existing Symbol

When you obtain new subcircuits from an outside source, they are generally provided in a single ASCII library file. *If you edit this file with a word processor, be sure to save the file in TEXT ONLY format.* Use the Model Data button in the Macro Utilities dialog box to create a link to each model in the new library file.

Example: Add an LF412C operational amplifier (5-pin subcircuit) to CircuitLogix.

1. Select Macro Utilities from the Macros menu.
2. Locate the symbol for the 5-pin opamp (Op-Amp5), then click on the Model Data button.
3. Click on the Open button and open the library file (Mcemods.lib) in which the LF412C model resides.
4. All of the model and subcircuit names found in this library will be displayed in the list box on the left. Click on the LF412C.

5. Enter appropriate information about the model in the Description field (for example: Dual LoOffset LoDrift JFET OpAmp).
6. In the Pkg Name field enter DIP8 to match the name of the component pattern.
7. Enter pin numbers to match the pad designations of the component. Since the LF412C is a dual op amp, you must enter pin numbers for both PART A and PART B. Click on the up or down arrow to switch between the two. For PART A the pin numbers should be: IN+ = 3, IN- = 2, V+ = 8, V- = 4 and Out = 1. For PART B the pin numbers should be: IN+ = 5, IN- = 6, V+ = 8, V- = 4 and Out = 7. Note that the V+ and V- power supply pins are the same for both op amps.
8. Click on the Add button to add the new reference. The new subcircuit can now be selected alphabetically from the list of 5-pin op amps.

Adding existing subcircuits to a new macro symbol:

Create a new nonfunctional macro device symbol as described earlier in this chapter. Be sure to place the pins in the same order as they are listed in the corresponding subcircuit. For example, if you are creating a new symbol for a 5-pin opamp, you should place the +Input pin first, then the -Input pin, the +Vsupply pin, the -Vsupply pin and finally the Output pin to correspond with the SPICE node connections for the 5-pin opamp subcircuit. This is not required, but it will be easier to understand when filling in the Spice Data field for the device. In order to link the new symbol to an existing subcircuit file, you must name the new symbol appropriately. For example, you may want to name the symbol “Op-Amp5:A” to link to the OPAMP5.SUB file.

Since CircuitLogix identifies pins by the order of placement in a device symbol, for clarity we recommend that the placement order of the pins match the order of the nodes listed in the subcircuits. The actual link between device symbol pins and SPICE subcircuits is controlled by the Spice Data field of the device where %n refers to the nth pin in the element list. For example, in the Spice Data

%D %1 %2 %3 %4 %S

the node number connected to the first pin placed on the device symbol is represented by %1, the second pin by %2, etc. When used with a subcircuit which begins

```
.SUBCKT XMR9933A 21 23 6 15
```

node 21 in the subcircuit connects to the first pin on the device symbol, etc.

With the new macro expanded, double-click on the device, then click on the Netlist button to view the Edit Device Data dialog box. Enter the following data:

- Enter an appropriate Auto Designation Prefix for the device. For example, U or IC for integrated circuits, etc.
- Set the Spice Prefix Character(s) to “X” for subcircuit devices.
- Place an appropriate SPICE instruction for this device into the Spice Data field. Refer to *Netlist Information* in the Schematic Capture chapter for more information or refer to the examples provided with other devices. Click on the OK button.

Save the macro by selecting the “Save Macro” command in the Macros menu.

Example: Create a new device symbol for optoisolators.

1. Create a new macro symbol for an optoisolator using the symbol editor described earlier in this chapter. Name the new macro device symbol Opto Isol:A. When placing pins on the new device, place the *first* pin on the anode (N+), the *second* pin on the cathode (N-), the *third* pin on the collector, and the *fourth* pin on the emitter. They should be placed in this order to match the syntax for the optoisolator subcircuits.
2. Expand the new macro symbol (if it is not already expanded) by clicking on it once with the left mouse button, then click on the Macro button in the Toolbar.
3. Double-click on the device symbol, and click on the Netlist button. Enter the following information in the Edit

Device Data dialog box of the expanded macro (to set the defaults for this macro device):

Auto Designation Prefix: OP
Spice Prefix Character(s): X
Spice Data field: %D %1 %2 %3 %4 %S

4. Save the macro. See *Save Macro* in the Macros Menu chapter for more information.

Parameter Passing

Parameter passing simplifies the task of creating new components. It allows you to pass databook values directly into generic SPICE models or subcircuits, using mathematical equations to create SPICE model parameters. The generic model is placed in the linking file associated with the device symbol, then referenced by an alias.

Devices created in this manner are selected and edited just like any other device model. To edit the parameters being passed, double-click on the device, then click on the Edit button.

General Form (generic model):

```
*MNAME:Device Title
*MNAME:P1:|P1 Description
[<<Min>,<Max>>]|Default
*MNAME:P2:|P2 Description
[<<Min>,<Max>>]|Default
:
*MNAME:Pn:|Pn Description
[<<Min>,<Max>>]|Default
*{P1=Default P1=Default .. Pn=Default}
*Desc pkg:Package Pins
Generic .MODEL or .SUBCKT using {P1, P2, etc. in math expressions}
```

General Form (alias):

```
*Desc alias:MNAME {P1=Val P2=Val} pkg:Package
Pins
.PARAM ANAME
```

MNAME is the name of the generic model or subcircuit, including the appropriate SPICE prefix character. P1, P2, etc. are the names of the parameters being passed. P1 Description, P2 Description, etc. are the description of the parameters. Min and Max are optional items which limit the value that can be entered for each parameter. Default is the default value for each parameter if no value is specified. Desc is a description of the device. Package and Pins information is used for export to TraxMaker for PCB layout. ANAME is the alias name (the name of the specific device).

Valid operators in the math expressions include:

+ - * /

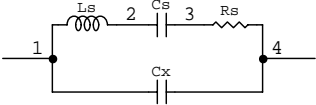
Following is an example of a generic subcircuit for a crystal which can be used as a building block for creating other crystals.

```
*XCRYSTAL:Crystal Subcircuit Parameters
*XCRYSTAL:FREQ:|Fundamental frequency
[1,]|1MEG
*XCRYSTAL:RS:|Series resistance [1,]|750
*XCRYSTAL:CX:|Parallel capacitance [0,]|13pf
*XCRYSTAL:Q:|Quality Factor [10,1000]|1000
*{FREQ=1Meg RS=750 C=13pf Q=1000}
*Generic 1MHz Crystal:crystal pkg:XTAL1 1,2
.SUBCKT XCRYSTAL 1 2
LS 1 2 {((Q*RS)/(6.2831852*FREQ))} IC=0.5M
CS 2 3 {(1/(Q*6.2831852*FREQ*RS))}
RS 3 4 {RS}
CX 1 4 {CX}
.ENDS XCRYSTAL

*alias:XCRYSTAL {FREQ=2E6 RS=250} pkg:XTAL1
1,2
.PARAM A2.000MHZ
```

Since CX and Q are not passed in the alias parameter list, the

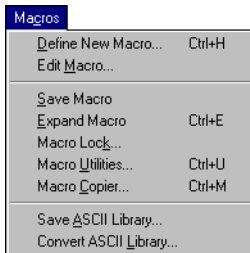
default values of 13pF and 1000 will be used.



Crystal Subcircuit

Chapter 9

Device Editor Menu



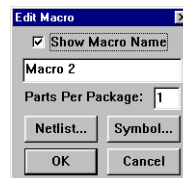
Using commands in the Device Editor (Macros) menu, you can expand CircuitLogix to meet your exact needs. This chapter describes the commands used to create, edit and manipulate macros. Refer to the Device Editor chapter for a step-by-step tutorial about creating macro devices.

Define New Macro...

Select the “Define New Macro...” command to make the current circuit into a macro device. Macros are saved in a disk file called “USER.LIB” and once created, are available for use in any circuit or in another macro device.

Edit Macro...

Click on an expanded macro’s symbol to select it, then select the “Edit Macro...” command to edit its name, device data or symbol. Alternately, you can double-click on the expanded macro’s package with the Arrow Tool. If an existing macro is expanded and saved with a new name, it is saved as a new macro and does not overwrite the existing macro. The following dialog box will be presented upon executing the Edit Macro command:



If no macro (either a newly defined or an expanded one) is present in the work area, this menu item will be dimmed indicating that it is not available. A macro may be defined as having multiple parts per package; that is, you may specify that there are more than one of these macro circuits, all identical, in a single chip. See *Groups* in the Index for more information.

Save Macro

The “Save Macro” command enables you to save or resave a macro device. An existing macro saved in this manner will be placed in its original position in the device library. All devices require a Major Device Class designation, so if you use the Save Macro command to save a new macro, CircuitLogix will display the Macro Utilities dialog box allowing you to specify both a major and minor class. To save an existing macro in a new location, refer to the *Macro Utilities* section below. To save under a different name, refer to *Edit Macro* above. Macro devices are stored in the file USER.LIB.

Expand Macro

The “Expand Macro” command enables you to edit a previously saved macro device. Expanding a macro device means that the black box containing the macro’s circuitry is opened and all internal circuitry is visible.

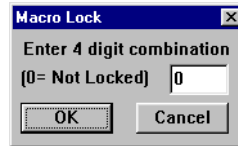


To expand a macro, click on a single macro device to select it and choose the “Expand Macro” command from the Macros menu (or click on the Macro button in the Toolbar). An alert box will appear warning that expansion of the macro will clear the work area. *If you haven’t saved the present work, click on the Cancel button to abort the expansion, then save the current work area and reselect the “Expand Macro” command.* If you don’t need to save the work area, click on the OK button to complete the expansion of the macro.

Make changes to an expanded macro using any of the available editing commands. When you have finished making changes to the macro, select the “Save Macro” command in the Macros menu.

Macro Lock...

An instructor may want to lock a macro so that it cannot be expanded by the students, thus creating a “black box”. To do this, expand the macro, click on the expanded macro’s symbol to select it, then select the “Macro Lock...” command. The following dialog box will be displayed.

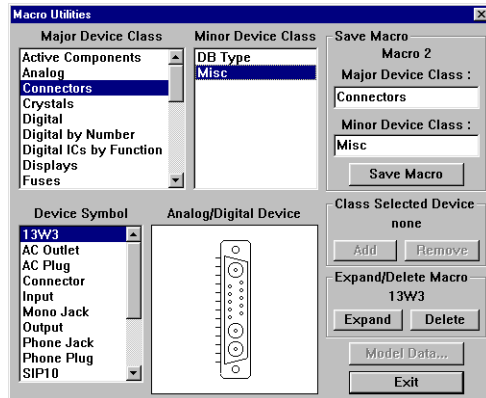


Enter any 4-digit number (1-9999), then save the macro. When you attempt to expand the macro again, you will be prompted to enter the 4-digit code. A code of 0 (zero) leaves the macro unlocked.

Warning: *If you forget the code, you will be unable to expand the macro.*

Macro Utilities...

Selecting this option will bring up the following dialog box which enables you to save, expand or delete a macro.



Save Macro

This button enables you to save or resave a macro device. If there is no macro (expanded or newly defined) in the work area, this button will be dimmed. Refer also to the section *Save Macro* described earlier in this chapter.

To save a macro, select both a major and minor device class to indicate where the device will be saved in the device library, then click the Save Macro button. The macro will then be saved in USER.LIB and the work area will be cleared. New major and minor class names may be added by typing in a new name in the Major Device Class and Minor Device Class text edit fields.

Class Selected Device

The Add and Remove buttons are used to move devices into different major and minor device classes. First, select the device in the circuit window, then select Macro Utilities from the Macros menu. Select the major and minor device classes from the list boxes. Click on the Add button to add the device to the selected classes. Click on the Remove button to remove the device from the selected classes. If a device is removed from all classes, it will be inserted in the User Defined major class. To remove a device completely, see *Delete Macro*.

Expand Macro

The function of this button was explained earlier in this section. To expand a macro device, select the macro and then click the Expand Macro button.

Delete Macro

This button enables you to delete a macro device from USER.LIB. To delete a macro device, select the macro, then click the Delete Macro button. Remember to save anything in the work area prior to executing this command because following deletion of a macro the work area will be cleared! It is also a good idea to make a backup of USER.LIB before creating or deleting macros in case something goes wrong and you want to restore the original library. Note: Many of the devices listed

in the Macro Utilities dialog box are actually in the DEVICE.LIB file and cannot be deleted.

Warning: *You cannot open a circuit which uses a macro that has been deleted from the library unless you first create a new macro with the same name.*

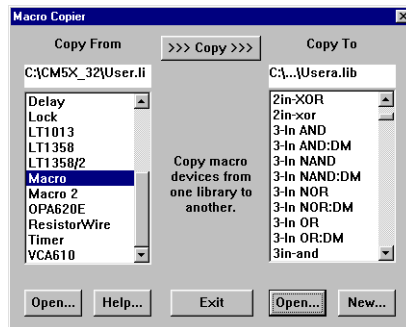
ModelData

This button is used to add new SPICE models to the CircuitLogix library. It brings up a dialog box which allows you to place new references into the linking file for selected symbol. Refer to *Model and Subcircuit Linking Files* in the Macro Devices chapter.

Macro Copier...

The following steps illustrate how you would use the Macro Copier to copy your macros into a new version of USER.LIB.

1. Choose the “Macro Copier...” command from the Macros menu. The following dialog box will appear.



2. Click on the Open button on the left side of the dialog to display the Open dialog box. You will be asked if you want to list only the user defined devices. If you are copying devices that you have created, answer “Yes”. Select the file from which macros will be copied. This will be the library file which has your macros in it (probably USER.LIB in your old CircuitLogix directory).

3. Click on the Open or New button on the right side of the dialog to display either the Open or the Save As dialog box. Select the file to which macros will be copied. This would normally be USER.LIB in your new CircuitLogix directory.
4. Select a macro to copy from the left hand file list. If a macro already exists with the same name, the existing macro will be deleted.
5. Click on the >>>Copy>>> button. You may be prompted for information regarding the simulation mode for which the device is intended. If the device can be used in digital simulations, check the Digital box; if it can be used in analog simulations, check the Analog box. If it can be used in either simulation mode, check both boxes.
6. Repeat steps 4 and 5 as needed.

Save ASCII Library...

Choose this command to write the currently loaded USER.LIB file to an ASCII file. A file selector dialog will be presented asking for the name for the ASCII file. In this format, user defined symbols which have been saved in Windows metafile or bitmap format will be lost and replaced by a simple rectangle. The pins, however, will remain intact. The ASCII format is used for conversion between 16- and 32-bit systems. This library may be converted back to binary format using the "Convert ASCII Library..." command.

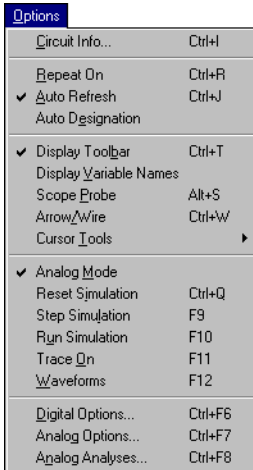
Convert ASCII Library...

Choose this command to convert an ASCII user library file (created with the "Save ASCII Library" command described above) to binary format. A file selector dialog will appear asking for the name of the ASCII user library file to be converted. Then a second file selector dialog will appear, asking for the name of the new binary file. This command is provided for compatibility between 16- and 32-bit systems.

Chapter 10

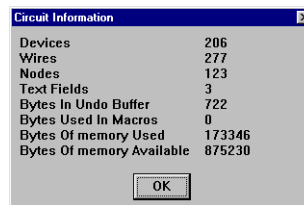
Options Menu

The Options menu contains commands which enable you to control various display, editing and simulation options.



Circuit Info...

Choose the “Circuit Info...” command to see how many items are currently present in the work area and to find out how much free memory is available. The following dialog box will appear showing you this information.



Repeat On

This option enables you to control how devices are selected from the library. With “Repeat On” checked, as soon as a device is selected from the library and placed, another identical device is automatically selected and made ready for placement. This repeat placement process continues until you cancel it by pressing any key on the keyboard or by double-clicking the mouse. With “Repeat On” not checked, you must select and place each device separately.

Auto Refresh

This option enables you to control the refresh mode. When Auto Refresh is enabled, the screen will refresh automatically as the circuit is edited. When it is disabled, the screen must be refreshed manually while editing (see the “Refresh Screen” command in the Edit menu). Disabling this option will allow you to work more quickly on a slower computer system.

Auto Designation

When this option is checked, the “Set Auto Designation” command (see the Edit menu) will be executed automatically each time you run an analog simulation. This option must be

disabled if you want to use your own device designations.

Display Toolbar

The “Display Toolbar” command shows or hides the Toolbar at the top of the CircuitLogix window.

Display Variable Names

The “Display Variable Names” option replaces the A, B, C labels used on the analog analysis graphs with the variable names such as V(8), etc. This allows you to identify the various points in the circuit more consistently with those being graphed in analysis windows.

Scope Probe

This option, when checked, causes the logic levels detected by the Probe Tool to be displayed in the digital Waveforms window while running digital simulation.

Arrow/Wire

This option, when checked, allows you to initiate a wire by clicking once on a device pin with the arrow tool. You can only use the Arrow Tool to initiate a wire on a device pin or to extend an existing wire. You cannot initiate a wire from the middle of another wire using the Arrow Tool as you can with the Wire Tool. This option can be used for both auto and manual routing.

Cursor Tools

This submenu provides an alternate method of selecting the Arrow, Wire, Text, Delete, Naming, Zoom, Probe and Help Tools.

Digital/Analog Mode

This works just like the Digital/Analog button in the Toolbar which is described in the Digital Logic Simulation chapter. The name that is displayed in the menu is the simulation mode that is currently selected.

Reset Simulation

This works just like the Reset button in the Toolbar which is described in the Digital Logic and Analog/Mixed-Mode Simulation chapters.

Step Simulation

Allows you to run the simulation for one step. When you select this option, the simulation runs for one step and then stops. This command is useful for debugging a circuit, especially when used in conjunction with the Trace button.

Run Simulation

Used instead of the ON/OFF button to switch on the power and run the simulation.

Trace On

This works just like the Trace button in the Toolbar which is described in the Digital Logic Simulation chapter.

Waveforms

This works just like the Waveforms button in the Toolbar which is described in the Digital Logic and Analog/Mixed-Mode Simulation chapters.

Digital Options...

This dialog box is used to control the size of a step when running the simulation in single step mode, to set the conditions for break points and to set the simulation speed. Refer to *Digital Options* in the Digital Logic Simulation chapter.

Analog Options...

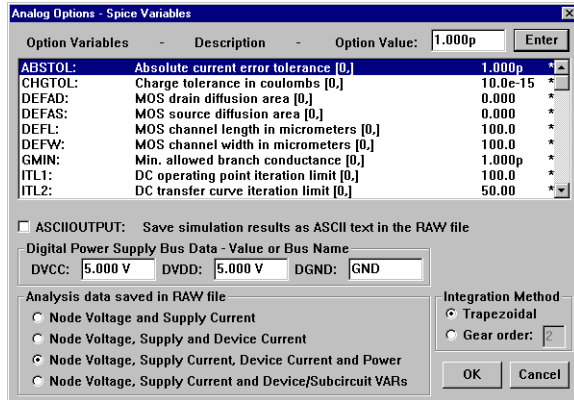
This dialog box is used to control the SPICE simulation options such as temperature, tolerances, default Test Points, etc. Refer to *Analog Options* in the Analog Simulation chapter.

Analog Analyses...

This dialog box is used to set up the SPICE analyses to be performed. Refer to the section *Analog Analyses* in the Analog/Mixed-Mode Simulation chapter.

Analog Options

SPICE allows you to control certain aspects of its simulation such as iteration limits, temperature, etc. When you select “Analog Options...” from the Options menu the following dialog box will appear.



ASCIIOUTPUT Check box

By default, the simulation data is saved in the .RAW file in binary format, creating smaller .RAW files. By enabling this check box, SPICE will save the simulation data in ASCII format, allowing you to read the data directly or load it into other applications. Default= binary format.

DVCC, DVDD and DGND

These are default values of the Vcc, Vdd and ground buses for the digital devices. If these buses are not specified in the circuit, these values will be used. If a bus name is entered instead of a value, that bus will be connected to the devices. By default, DVCC and DVDD=+5V, DGND=“GND”.

Integration Method

This sets the numerical integration method used by SPICE. The trapezoidal method is relatively fast and accurate, but tends to oscillate under certain conditions. The gear method requires longer simulation times, but tends to be more stable. The gear order must be a value between 2 and 6. Using a higher gear order theoretically leads to more accurate results, but increases simulation time. Default=Trapezoidal.

Analysis data saved in RAW file

Four radio buttons are used to select the default level at which variables are saved in the .RAW file. This determines how much data is actually stored in the .RAW file and which variables can be plotted in the analysis windows when the simulation is run. When the top button is selected, only the node voltages and supply currents are stored. When the second button is selected, node voltages, supply currents and device pin currents are stored. When the third button is selected, node voltages, supply currents, device pin currents and device power dissipation are stored. When the bottom button is selected, node voltages, supply currents, device pin variables and subcircuit internal variables are stored. If any user defined Test Points have been added to the circuit, data will be saved only for each user defined Test Point and these radio buttons will be disabled.

Option Variables

Default values for each option are identified in the dialog box by an asterisk (*). To change the value of a SPICE option, select the parameter and type the new value into the Value edit field. If you want to set a specific option to its default value, type an asterisk in the Value edit field. Following is a list of the options and their effect on the simulation.

<u>Option</u>	<u>Effect</u>
ABSTOL	Sets the absolute current error tolerance of the program. Set ABSTOL=RELTOL* (lowest current magnitude in the circuit). Default=1 picoamp.
CHGTOL	Used in the LTE timestep control algorithm as a lower limit on capacitor charge or inductor flux. Default=1.0e-14 coulombs.

DEFAD	Sets the MOS drain diffusion area. Default=0.0 meters ² .
DEFAS	Sets the MOS source diffusion area. Default=0.0 meters ² .
DEFL	Sets the MOS channel length. Default=100.0 micrometer.
DEFW	Sets the MOS channel width. Default=100.0 micrometer.
GMIN	Sets the minimum conductance (maximum resistance) of any device in the circuit. It also sets the value of the conductance that is placed in parallel with each pn junction in the circuit. Default=1.0e-12 mhos. <i>Raising this value may help with simulation convergence in many circuits, but decrease accuracy.</i>
ITL1	Sets the Operating Point Analysis iteration limit. Default=100 iterations. <i>This may need to be raised as high as 500 for many circuits.</i>
ITL2	Sets the DC Analysis iteration limit. Default=50 iterations. <i>This may need to be raised as high as 200 for some circuits.</i>
ITL3	Sets the lower Transient Analysis iteration limit. Default=4 iterations. <i>This is not implemented in SPICE3. It is provided in CircuitLogix for compatibility in creating SPICE2 netlists.</i>
ITL4	Sets the Transient Analysis timepoint iteration limit. Default=10 iterations. <i>Raising this value to 100 or more may eliminate "timestep too small" errors improving both convergence and simulation speed.</i>
ITL5	Sets the Transient Analysis total iteration limit. Default=5000 iterations. <i>This is not implemented in SPICE3. It is provided in CircuitLogix for compatibility in creating SPICE2 netlists.</i>

PIVREL	Sets the relative ratio between the largest column entry in the matrix and an acceptable pivot value. The value must be between 0 and 1. Default=1.0e-3. In the numerical pivoting algorithm the allowed minimum pivot is determined by $EPSREL=AMAX1(PIVREL*MAXVAL, PIVTOL)$ where MAXVAL is the maximum element in the column where a pivot is sought (partial pivoting).
PIVTOL	Sets the absolute minimum value for a matrix entry to be accepted as a pivot. Default=1.0e-13.
RELTOL	Sets the relative error tolerance of the program. The value must be between 0 and 1. Default is 0.001 (0.1%). <i>Larger values mean faster simulation time, but less accuracy.</i>
TEMP	Sets the actual operating temperature of the circuit. Any deviation from TNOM will produce a change in the simulation results. Default=27°C. <i>TEMP can be overridden by a temperature specification on any temperature dependent instance.</i>
TNOM	Sets the nominal temperature for which device models are created. Default=27°C. <i>TNOM can be overridden by a specification on any temperature dependent device model.</i>
TRTOL	Used in the LTE timestep control algorithm. This is an estimate of the factor by which SPICE overestimates the actual truncation error. Default=7.0.
VNTOL	Sets the absolute voltage tolerance of the program. Set $VNTOL=RELTOL*$ (lowest voltage magnitude in the circuit). Default=1 microvolt.
BOOLL	Sets the low output level of a boolean expression. Default=0.0V.

BOOLH	Sets the high output level of a boolean expression. Default=4.5V.
BOOLT	Sets the input threshold level of a boolean expression. Default=1.5V.
BADMOS3	Use the older version of the MOS3 model with the “kappa” discontinuity. Default=NO (don’t use the older version).
KEEPOPINFO	Retain the operating point information when an AC Analysis is run. <i>This is particularly useful if the circuit is large and you do not want to run a (redundant) Operating Point Analysis.</i> Default=NO (run OP each time).
TRYTOCOMPACT	Applicable to the LTRA model. When specified, the simulator tries to condense LTRA transmission line’s past history of input voltages and currents. Default=NO (don’t compact).
NOOPITER	Skip directly to GMIN stepping algorithm. Default=NO (don’t skip).
GMINSTEP	Sets the number of steps in the GMIN stepping algorithm. When set to 0, GMIN stepping is disabled, making source stepping the simulator’s default DC (operating point) convergence algorithm. Default=10 steps.
SRCSTEP	Sets the number of steps in the source stepping algorithm for DC (operating point) convergence. Default=10 steps.
ACCT	Causes accounting and run-time statistics to be displayed. Default=NO (no display).
LIST	Displays a comprehensive list of all elements in the circuit with connectivity and values. Default=NO (no list).
OPTS	Displays a list of all standard SPICE3 Option parameter settings. Default=NO (no list).
BYPASS	Enables the device bypass scheme for non-linear model evaluation. Default=1 (on).

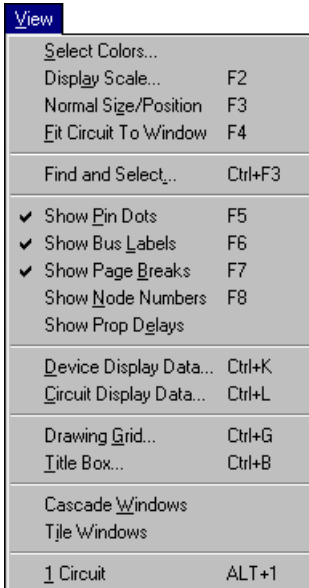
MINBREAK	Sets the minimum time between breakpoints. Default=0 seconds (sets the time automatically).
MAXOPALTER	Sets the maximum number of analog/event alternations for DC (operating point) convergence. Default=0.
MAXEVTITER	Sets the maximum number of event iterations for DC (operating point) convergence. Default=0.
NOOPALTER	Enables DC (operating point) alternations. Default=NO.
RAMPTIME	Controls turn-on time of independent sources and capacitor and inductor initial conditions from zero to their final value during the time period specified. Default=0.0 seconds.
CONVLIMIT	Disable convergence algorithm used in some built-in component models. Default=NO.
CONVSTEP	Sets the limit of the relative step size in solving for the DC operating point convergence for code model inputs. Default=0.25.
CONVABSSTEP	Sets the limit of the absolute step size in solving for the DC operating point convergence for code model inputs. Default=0.1.
AUTOPARTIAL	Enables the automatic computation of partial derivatives for XSpice code modules. Default=NO.
PROPMNS	Scale factor used to determine minimum propagation delay when actual value is not specified in simcode model. Default=0.5 (50% of typical propagation delay).
PROPMXS	Scale factor used to determine maximum propagation delay when actual value is not specified in simcode model. Default=1.5 (150% of typical propagation delay).
TRANMNS	Scale factor used to determine minimum transition time when actual value is not specified in simcode model. Default=0.5 (50% of typical transition time).

TRANMXS	Scale factor used to determine maximum transition time when actual value is not specified in simcode model. Default=1.5 (150% of typical transition time).
LOADMNS	Scale factor used to determine minimum input loading (maximum input resistance) when actual value is not specified in simcode model. Default=1.5 (150% of typical input resistance).
LOADMXS	Scale factor used to determine maximum input loading (minimum input resistance) when actual value is not specified in simcode model. Default=0.5 (50% of typical input resistance).
DRIVEMNS	Scale factor used to determine minimum output drive capacity (maximum output resistance) when actual value is not specified in simcode model. Default=1.5 (150% of typical output resistance).
DRIVEMXS	Scale factor used to determine maximum output drive capacity (minimum output resistance) when actual value is not specified in simcode model. Default=0.5 (50% of typical output resistance).
CURRENTMNS	Scale factor used to determine minimum supply current (maximum internal resistance) when actual value is not specified in simcode model. Default=1.5 (150% of typical internal resistance).
CURRENTMXS	Scale factor used to determine maximum supply current (minimum internal resistance) when actual value is not specified in simcode model. Default=0.5 (50% of typical internal resistance).
TPMNTYMX	Temporary global override for propagation delay index on simcode devices. (0=default, 1=min, 2=typ, 3=max). Default=0.
TTMNTYMX	Temporary global override for transition time index on simcode devices. (0=default, 1=min, 2=typ, 3=max). Default=0.

LDMNTYMX	Temporary global override for input loading index on simcode devices. (0=default, 1=min, 2=typ, 3=max). Default=0.
DRVMNTYMX	Temporary global override for output drive capacity index on simcode devices. (0=default, 1=min, 2=typ, 3=max). Default=0.
IMNTYMX	Temporary global override for supply current index on simcode devices. (0=default, 1=min, 2=typ, 3=max). Default=0.
SIMWARN	A nonzero value indicates that simcode warning messages can be displayed during run time. Simcode warnings may include information concerning timing violations (tsetup, thold, trec, tw, etc.) or indicate supply voltage dropping below device specifications. Default=0.
RSHUNT	Value in ohms of resistors added between each circuit node and ground, helping to eliminate problems such as “singular matrix” errors. In general, the value of RSHUNT should be set to a very high resistance (1e+12). Default=0 (no shunt resistors).
ADCSTEP	The minimum step size required to register an event on the input of the internal A/D converters. Default=0.01 volts.

Chapter 11

View Menu



The View menu contains commands which enable you to control various display options and windows.

Many of the items in the View menu are an on/off type option. If a check mark is shown beside an item, then the item is currently selected; if a check mark is not shown beside an item, it is not currently selected. Just highlight the menu item and click the left mouse button, and it will be either selected (checked) or deselected (not checked).

Select Colors...

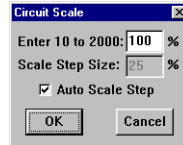
This feature allows the user to select the color associated with several functions (i.e. low level, high level) and items (i.e. Logic Display color, Hex and ASCII Key Cap color). The color of individual LEDs can be changed, for example, by first selecting one or more LEDs, choosing the “Select Colors...” command and changing the SELECTED LEDs color. The selected LED color will be the default color for all new LEDs. Logic Display colors can be changed by double-clicking on them. You can also individually select the color of 7-segment displays, analog waveforms, etc.



Choose “All Items” to set everything (except the background) to the same color. Press the Defaults button to restore everything to its original color. Press the Save Selections button to save the selected colors even after quitting CircuitLogix. Note: The color selections are stored in a single file, not with each individual circuit.

Display Scale...

This feature allows the user to select the scale at which a circuit will be displayed. It also allows user to select the Scale Step of the Zoom Tool.



Normal Size/Position

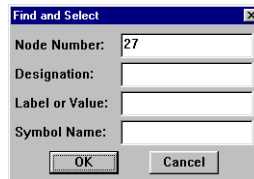
This feature changes the circuit display scale to 100% and moves the horizontal and vertical scroll bars to their home positions.

Fit Circuit To Window

This feature will reduce or enlarge the circuit (by changing the Display Scale and scroll bar positions) so that the entire circuit can be displayed within the circuit window.

Find and Select...

This command will display the following dialog box, allowing you to search through your schematic and identify specific devices or nodes.



For example, if a SPICE message indicates a problem at node 27 and your schematic is rather complex, this command can help you find the specified node. All wires connected to that node will be selected. You can also search for multiple devices. For example, to select all the transistors, enter "Q*" in the Designation field. All characters after the "Q" will be ignored in the search.

Show Pin Dots

This option allows you to control how connections between wires and devices are drawn. With “Show Pin Dots” checked, connections between wires and devices will be shown by a small dot where the connection occurs. If “Show Pin Dots” is not checked, the connecting dots will not be drawn. Dots showing where wires connect to other wires are always displayed.

Show Bus Labels

This option displays bus labels (numbers) by putting a small, numbered box at the end of each bus wire, and also shows the number of each bus connection wire next to the corresponding bus connection.

Show Page Breaks

This option displays the page divisions of a multi-page circuit when it is sent to the printer. The Show Page Breaks option is also useful when positioning a circuit on a single page. The page breaks you see on your screen are based on the printer selection, paper size and scaling factor.

Show Node Numbers

This option displays the node number for each node in your circuit. The number is shown at the center of the longest wire segment of each node. The node numbers are determined by CircuitLogix and are used for simulation purposes. They correspond to the variable names used in the analog analyses’ graphs, and can be used as a reference when viewing the graphs.

Show Prop Delays

This option displays the propagation delay for each device. With “Show Prop Delays” checked, the delay for each device will be shown within a rounded rectangle at the center of each device. If “Show Prop Delays” is not checked all devices will be drawn in normal form without the delay value being visible. Some devices such as Pulsers, LEDs and Macros do not have a delay, so no value is shown for these devices.

To change the propagation delay of one or more devices

select the “Set Prop Delay(s)...” command from the Edit menu.

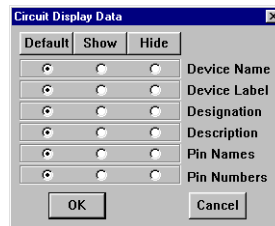
Device Display Data...

This dialog box allows you to quickly change the display settings of all selected devices. If the display item is currently visible on some of the selected devices, but not on others, the check box will be gray. Refer to *Netlist Information* in the Schematic Capture chapter for more information.



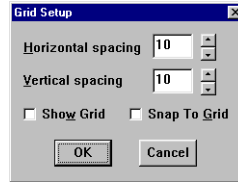
Circuit Display Data...

This dialog box allows you to temporarily override the display settings of every device in the entire circuit. The buttons at the top of each column will quickly change the settings for the entire column, or you can change individual items by clicking on the radio buttons. The Default setting tells CircuitLogix to use the Visible settings of each individual device. Refer to *Netlist Information* in the Schematic Capture chapter.



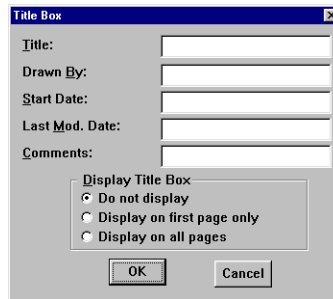
Drawing Grid...

Select “Drawing Grid...” to turn the alignment grid of the circuit window on or off. The grid is useful as an aid in precisely aligning objects. Snap-To-Grid allows new devices (devices not already in a circuit) to be placed according to the specified grid. It also allows old devices (devices already in the circuit) to be moved according to the selected grid, relative to their original position. *Snap to Grid, however, does not guarantee alignment of component pins.*



Title Box...

Choose the "Title Box..." command to add a title box to the lower right corner of the page. When you select the Title Box feature, the following dialog box will appear:



By selecting the desired check box you can specify if you want a Title Box to appear on every page, only the first, or not at all.

Cascade Windows

This command will arrange all open windows in a cascaded (stacked) order.

Tile Windows

This command will arrange all open windows in a tiled (adjacent) order.

Windows

Each window that is open (or can be opened with the available simulation data) is listed here in the menu. Selecting a window from this menu will open it and bring it to the front.

Chapter 12

Device Library

The device libraries are stored in two files: DEVICE.LIB and USER.LIB. All macro devices created by the user are stored in USER.LIB.

Devices can be selected from the libraries in various ways. Refer to *Placing Devices* in the Schematic Capture chapter. Once you have selected a device, it will follow the mouse to any position within the drawing window. If the portion of the work area where you need to position the device is not currently visible, move the device to the border of the window and the circuit will automatically scroll.

To place the device, click the mouse and the device will be placed at its present position. To cancel placement of the device press any key or double-click the mouse. Devices can be mirrored and/or rotated prior to placing them. To rotate a device, press the R key or click the right mouse button. To mirror a device, press the M key.

The “Repeat On” option in the Options menu determines whether the same device can be placed in the work area multiple times, or if you must select each device separately to place it. Refer to *Repeat On* in the Options Menu chapter for further details.

The devices and instruments are described in this chapter. They are classified into the following categories:

Analog	Can be used with the analog simulator.
Digital	Can be used with the digital simulator.
Mixed-mode	Can be used with either the analog or digital simulator.
Schematic	These devices are intended for drawing schematics and are not functional.

To assist you in finding the items in the library, each of the following device descriptions include the location in the parts browser. The location is displayed in the following format:

[Major Device Class/Minor Device Class] (Default Hotkey)

CircuitLogix is by no means limited to the devices provided in the device library. New devices can be created as macros or by importing SPICE-compatible subcircuits. User defined devices are permanently stored in the library files and can be used just like any other device. Refer to the Device Editor chapter for a complete discussion about creating new devices.

+V: Mixed-mode [Analog/Power] (1)

In analog simulation mode, this device provides a fully programmable DC power supply. It can be programmed for either a positive or negative voltage. It always uses the ground node as a reference. In digital simulation mode it provides a fixed High state. See also: Independent Sources. Example circuits: LOCK.CKT, ANALOG.CKT.



.IC: Analog [Analog/SPICE Controls] (I)

Wire an .IC device to any node that needs .IC (Initial Conditions) programmed. Double-click on the device and set the Label-Value field to the desired initial value. See also: Initial Conditions. Example circuit: 555.CKT.



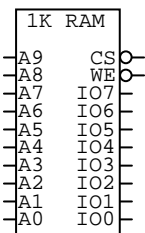
.NODESET: Analog [Analog/SPICE Controls] (N)

Wire a .NODESET device to any node that needs Nodeset programmed. Double-click on the device and set the Label-Value field to the desired Nodeset value. See also: Initial Conditions. Example circuit: BISTABLE.CKT.



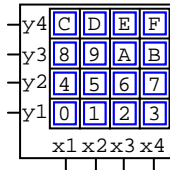
1K RAM: Mixed-mode [Digital/RAM-PROM]

This is a 1024 x 8 Random Access Memory. Multiple RAM chips may be used in a single circuit or macro. However, RAM data is not saved with the circuit or macro. Both the CS and WE pins must be pulled low to write into the RAM. To read from the RAM, keep the WE pin high and pull the CS pin low. For debugging purposes, you can edit the contents of the RAM. Refer to *Edit PROM/RAM* in the Edit Menu chapter.



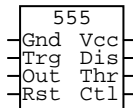
4x4 Switch: Mixed-mode [Switches/Matrix]

This is a 4x4 switch matrix. Clicking on one of the buttons closes a switch which connects one of the horizontal pins to one of the vertical pins. The switch remains closed until another button is clicked. If you try to activate the switch while running an analog simulation, CircuitLogix will ask if you want to rerun the simulation with the switch in the new position. Example circuit: 4X4.CKT.



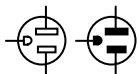
555 Timer: Analog [Linear ICs/Timers]

This device includes the SPICE data for simulating a 555 Timer. Timers may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. When creating astable circuits, initial conditions will be required on the timing capacitor in order for SPICE to converge on a solution. For monostable operation, the component model subcircuit should be used rather than the macromodel subcircuit. See also: Subcircuits, Initial Conditions. Example circuit: 555.CKT.



AC Outlet, AC Plug: Analog/Schematic [Connectors/Misc]

These devices are included for schematic purposes, however, the AC Outlet may be included during Analog simulations. Each contact is treated as having a very high shunt resistance ($1E+12\Omega$) to ground.



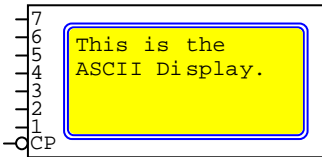
Antenna: Analog [Schematic Symbols/Antennas]

This device includes an internal 50Ω resistor to ground. The resistance can be changed by editing its Label-Value field.



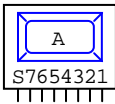
ASCII Display: Mixed-Mode [Displays/Digital Only] (a)

This represents a 16X4 LCD display intended mainly for use in Digital simulation mode. Up to 4 lines of 16 ASCII characters per line can be displayed on this device. The character, determined by the binary code applied to pins 1-7, is sent to the display when the level on the /CP pin changes from high to low. NOTE: <RETURN> will start a new line, <BACKSPACE> will delete a previously entered character, CTRL+G (bell) will send a beep to the PC's speaker, and CTRL+L (form-feed) will clear the display. The background color of the display can be changed. The text is always the Device Text color. This device can be used in Analog simulation mode as well, but no characters will be displayed. Example circuit: DISPLAY.CKT.



ASCII Key: Digital [Digital/Input Device] (A)

The ASCII Key operates like the Hex Key with one addition: A strobe pulse (low, high, low level) is generated and appears on the pin labeled "S". The binary code for any ASCII character can be generated by this device. This device simulates a keyboard. Example circuit: DISPLAY.CKT.



Battery: Mixed-mode [Analog/Power] (b)

In analog simulation mode, this device includes the SPICE data for simulating a DC voltage source. The voltage is specified in the Label/Value field. In digital simulation mode it provides a fixed High state on the "+" pin and a fixed Low state on the "-" pin. See also: Independent Sources. Example circuit: CEAMP.CKT.



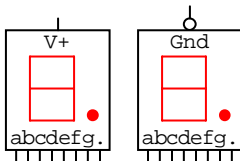
Buffer/Amp: Analog [Linear ICs/Buffers-Amps]

This device includes the SPICE data for simulating buffers and amplifiers. Buffers/amps may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



CA 7-Seg, CC 7-Seg: Mixed-mode [Displays/7-Segment LEDs]

These are Common-anode and Common-cathode 7-Segment LED Displays intended mainly for use in Digital simulation mode. They will display each of 7 segments plus the decimal point, corresponding to the input pins as they are pulled low. Different colors can be selected for each display. If the Prop Delay for this device is set to greater than 1, the display will remain visible even when power is removed. This allows the display to be multiplexed without causing a flash each time the display is addressed. This allows the display to better simulate real-time operation. In a real circuit the flashes are not usually noticeable because of the high repetition rate of the multiplexing circuit. These devices can be used in Analog simulation mode as well, but the segments will not light up. Example circuit: DISPLAY.CKT.



Capacitor, Polar Cap, Var Capacitor: Analog [Passive Components/Capacitors] (c, C)

These devices include the SPICE data for simulating a capacitor. The value of the capacitor is specified in the Label/Value field. It is drawn as a “variable” capacitor for schematic purposes only. See also: Capacitors, Initial Conditions. Example circuit: 555.CKT.



SemiCapacitor: Analog [Passive Components/Capacitors]

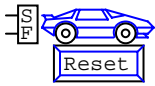
This device includes the SPICE data for simulating a semiconductor capacitor. Semiconductor capacitors may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits, Semiconductor Capacitors.



Car: Mixed-mode [Digital/Output Device]

This is an animated device with a Start input, a Finish output and a Reset button. It is intended mainly for use in Digital simulation mode. Multiple cars may be placed in the circuit. The number of simulation ticks it takes for a car to reach the finish is random for each car. A new random number is generated each time the Reset button *in the Toolbar* is pressed. The reset button on the car returns the car to its starting position. Double-click on the car to program the travel distance (up to

1000 units). Different colors can be selected for each car. This device can also be used in Analog simulation mode, but there will be no animation and the Finish output will never go true. Example circuit: CARS.CKT.



Caution: Schematic [Schematic Symbols/Misc]
This device is included for schematic purposes.



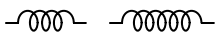
CDA-5: Analog [Linear ICs/Buffers-Amps]
This device includes the SPICE data for simulating current differencing amplifiers. Amplifiers may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



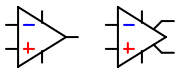
Coax, TwinLead, TwistedPair: Schematic [Schematic Symbols/Cables]
These devices are included for schematic purposes. SPICE data may be added by the user to simulate the lossless, lossy or URC transmission lines.



Coil 3T, Coil 5T: Analog [Passive Components/Inductors]
These devices include the SPICE data for simulating an inductor. The inductance of the coil is specified in the Label/Value field. See also: Inductors. Example circuit: RESONANT.CKT.



Comparator5, Comparator6: Analog [Linear ICs/Comparators]
These devices include the SPICE data for simulating 5-pin and 6-pin voltage comparators. Comparators may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



Connector: Mixed-mode [Connectors/Misc]
This device has an internal wire that connects the two sides, so it behaves just like a wire during simulation.



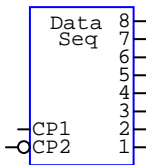
Crystal: Analog [Crystals/Standard]

This device includes the SPICE data for simulating a crystal. Crystals may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: XTAL_OSC.CKT.



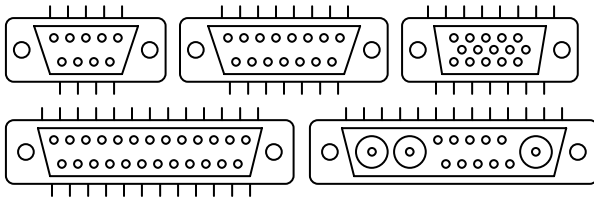
Data Seq: Mixed-mode [Digital/Instrument]

The Data Sequencer allows the user to specify up to 32k bytes which can be output in a defined sequence. Multiple Data Sequencers may be individually programmed. Refer to the *Data Sequencer* section in the Digital Logic Simulation chapter to program this device. Example circuit: BUSWIRE.CKT.



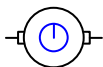
DB-9, DB-15, DB-15HD, DB-25, 13W3: Schematic [Connectors/DB Type, Misc]

These devices are included for schematic purposes. Bus wires may be added internally to the macro to connect two of them together for simulation. Be sure to use a unique bus wire number for each pair of connectors. Example circuit: DB9.CKT.



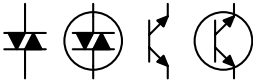
DC Motor: Mixed-mode [Motors/DC]

In digital simulation mode, the DC Motor is an animated device. The armature will rotate clockwise when there is a high on the positive terminal and a low on the negative terminal. It will rotate counter clockwise when there is a high on the negative terminal and a low on the positive terminal. In analog mode, the DC Motor is not animated, but it acts as an inductor and a resistor in series. Example circuit: LADDER.CKT.



Diac:A, Diac:B, Diac:C, Diac:D: Schematic [Schematic Symbols/Diacs]

This device is included for schematic purposes.



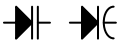
Diode: Analog [Active Components/Diodes] (d)

This device includes the SPICE data for simulating a junction diode. Diodes may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Diodes, Models, Initial Conditions. Example circuit: ASTABLE.CKT.



Varactor, Varactor:A: Analog [Active Components/Diodes]

This device includes the SPICE data for simulating a variable capacitance diode. Diodes may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Diodes, Models, Initial Conditions.



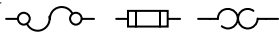
Zener Diode: Analog [Active Components/Diodes] (D)

This device includes the SPICE data for simulating a zener diode. Diodes may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Diodes, Models, Initial Conditions. Example circuit: PS1.CKT.



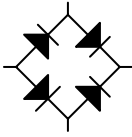
Fuse, Fuse:A, Thermal Fuse: Analog [Fuses/Electronic] (f) [Fuses/Thermal]

These devices include the SPICE data for simulating a current fuse. Fuses may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



FW Bridge: Analog [Active Components/Diodes]

This device includes the SPICE data for simulating a full-wave bridge rectifier. Bridges may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: PS1.CKT.



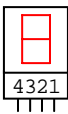
Ground: Mixed-mode [Analog/Power], [Digital/Power] (0 (zero))

In analog simulation mode, this device provides a ground reference node for the circuit. **Every analog circuit must have a ground reference.** In digital simulation mode it provides a fixed Low state.



Hex Display: Mixed-mode [Displays/Digital Only] (h)

This is a Hexadecimal 7-Segment Display with a built in decoder. It is intended mainly for use in Digital simulation mode. It will display a hexadecimal number (0-9 and A-F) based on the binary code applied to the pins 1-4. Different colors can be selected for each display. This device can also be used in Analog simulation mode, but the segments will not light up. Example circuit: SIM.CKT.



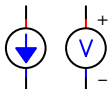
Hex Key: Digital [Digital/Input Device] (H)

This is a hexadecimal key. It will generate the binary code associated with the hexadecimal number displayed on the key. After the hex key has been selected (single click on the hex number shown in the middle of the device), the hex number displayed can be changed in two ways: 1) click on the number displayed to increment it, or 2) press a hexadecimal key (0-9 and A-F) on the keyboard. Multiple Keys may be used in a circuit; only the selected key will accept input from the keyboard. Example circuit: LOCK.CKT.



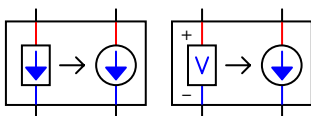
I Source, V Source: Mixed-mode [Analog/Power] (i, v)

In analog simulation mode, these are independent DC current and voltage sources. Enter the current or voltage in the Label-Value field. In digital simulation mode, they provide a fixed High state on the “+” terminal and a fixed Low state on the “-” terminal (the arrow points away from the “+” terminal, toward the “-” terminal on the I Source). See also: Independent Sources. Example circuit: 741.CKT.



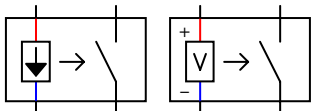
I->I Source, V->I Source: Analog [Analog/Power]

These are linear current-controlled and voltage controlled current sources. The current on the output (right-hand side) is controlled by the current or voltage on the input (left-hand side). See also: Current-Controlled Current Sources, Voltage-Controlled Current Sources. Example circuit: 741.CKT.



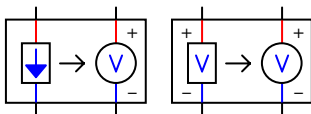
I->Switch, V->Switch: Analog [Switches/Controlled]

These devices include the SPICE data for simulating a current-controlled and voltage-controlled switch. Specific models may be added by the user. See also: Current Controlled Switches, Voltage Controlled Switches, Models. Example circuit: SWITCHES.CKT.



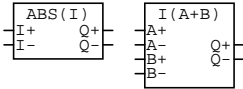
I->V Source, V->V Source: Analog [Analog/Power]

These are linear current-controlled and voltage-controlled voltage sources. The voltage on the output (right-hand side) is controlled by the current or voltage on the input (left-hand side). See also: Current-Controlled Voltage Sources, Voltage-Controlled Voltage Sources. Example circuit: 741.CKT.



I-Math1, I-Math2: Analog [Math Functions/Current]

These devices provide direct access to SPICE's nonlinear dependent source math functions. They are set up in subcircuit format to provide easy selection of the math functions. I-Math1 provides access to the single-variable current functions (abs, cos, sqrt, etc.) I-Math2 provides access to the dual-variable current operations (+, -, *, / and ^). See also: V-Math, Nonlinear Sources.



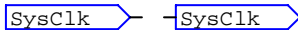
Inductor, Var Inductor: Analog [Passive Components/Inductors] (I, L)

This device includes the SPICE data for simulating an inductor. The inductance of the coil is specified in the Label/Value field. It is drawn as a “variable” inductor for schematic purposes only. See also: Inductors.



Input, Output: Mixed-mode [Connectors/Misc]

These are schematic symbols of page connectors in a circuit. All “Input” and “Output” devices that have the same name will operate as though they were connected together. Double-click on the device to edit the name. Example circuit: INOUTPUT.CKT.



Lamp: Mixed-mode [Displays/Incandescent]

This device includes the SPICE data for simulating a resistor in analog mode. The value of the resistor is specified in the Label/Value field. In digital simulation mode, the lamp will light when one terminal is high and the other terminal is low. Each lamp may be a different color.



LED: Mixed-mode [Displays/Diode]

This is a Light Emitting Diode. In digital mode, it will “light” when a low level is applied to its cathode and a high level is applied to its anode. Different colors can be selected for each LED. This device also includes the SPICE data for simulating an LED. LEDs may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: OPTO.CKT.



Logic Display: Mixed-mode [Displays/Digital Only]

This is a simple logic state indicator intended for use mainly in Digital simulation mode. It has only one pin and indicates whether the state is low (display off) or high (display on). Double-click on the Logic Display to change its color. This device can also be used in Analog simulation mode, but it does not light. Example circuit: LED.CKT.



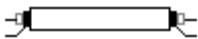
Logic Switch: Mixed-mode [Switches/Digital]

This is a switch which provides either a Low or a High logic level when in digital simulation mode and two programmable voltage levels while in analog simulation mode. When in analog mode, the default voltages levels are 0v and 5v. To change the voltage levels, select the switch by dragging a selection rectangle around it, double-click on it, and then enter a SPICE comment of the following form into the SPICE Data field: *0=1v 1=10v. This comment will make the switch output 1v for a low level and 10v for a high level. While digital simulation is running a single click on the switch will cause its output to immediately change states. If you click on the switch while running an analog simulation, CircuitLogix will ask if you want to rerun the simulation with the switch in the new position. To move a switch to a new location you must first drag a selection rectangle over the switch to select it. Example circuit: SIM.CKT.



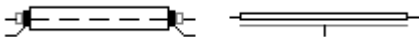
LossLessLine: Analog [Transmission Lines/SPICE Simulation]

This device includes the SPICE data for simulating a lossless transmission line. Characteristic impedance is specified in the Label/Value field. Time delay or frequency/normalized length are specified in the Spice Data field. See also: Lossless Transmission Lines. Example circuit: LLTRAN.CKT.



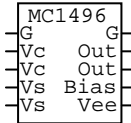
LossyLine, URC-Line: Analog [Transmission Lines/SPICE Simulation]

These devices include the SPICE data for simulating lossy and uniform distributed RC transmission lines. Specific models may be added by the user. See also: Lossy Transmission Lines, URC Lines, Models.



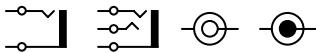
MC1496: Analog [Linear ICs/Modulators]

This device includes the SPICE data for simulating an MC1496 Balanced Modulator/Demodulator. Devices may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: AMMOD.CKT.



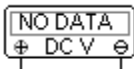
Mono Jack, Stereo Jack, Phone Jack, Phone Plug: Schematic [Connectors/Misc]

These devices are included for schematic purposes. Bus wires may be added internally to the macro to connect two of them together for simulation. Be sure to use a unique bus wire number for each pair of connectors.



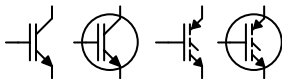
Multimeter: Analog [Analog/Instruments]

Multimeters can be wired directly into the circuit to measure resistance, voltage or current. DC voltage and current can only be measured if Operating Point Analysis is enabled. To measure DC AVG or AC RMS voltage or current, Transient Analysis must be enabled and must simulate enough cycles of transient data to make the measurements meaningful. Multiple multimeters may be individually programmed. Refer to the section “Multimeter” in the “Analog Simulation” chapter to program this device. Example circuit: SWITCHES.CKT.



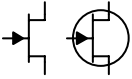
N-IGBT, N-IGBT:A, N-IGBT:B, N-IGBT:C: Analog [Active Components/IGBTs]

These devices include the SPICE data for simulating N-channel Insulated Gate Bipolar Transistors. Transistors may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Subcircuits.



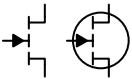
N-JFET, N-JFET:A: Analog [Active Components/JFETs] (j)

This device includes the SPICE data for simulating an N-channel junction field-effect transistor. JFETs may be selected from a list of available models (i.e. 2N4393) or new models may be added by the user. Double-click on the device to select the desired model. See also: JFETs, Models, Initial Conditions. Example circuit: CSJFAMP.CKT.



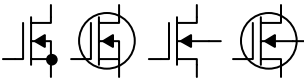
N-MESFET, N-MESFET:A: Analog [Active Components/MESFETs] (z)

This device includes the SPICE data for simulating an N-channel MESFET (GaAsFET). MESFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MESFETs, Models.



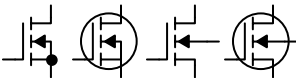
N-DMOS 3T, N-DMOS 3T:A, N-DMOS 4T, N-DMOS 4T:A: Analog [Active Components/MOSFETs Depl]

These devices include the SPICE data for simulating 3-terminal and 4-terminal N-channel Depletion Mode MOSFETs. MOSFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MOSFETs, Models, Initial Conditions.



N-EMOS 3T, N-EMOS 3T:A, N-EMOS 4T, N-EMOS 4T:A: Analog [Active Components/MOSFETs Enh] (m)

These devices include the SPICE data for simulating 3-terminal and 4-terminal N-channel Enhancement Mode MOSFETs. MOSFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MOSFETs, Models, Initial Conditions.



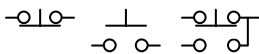
N-UJT: Analog [Active Components/Unijunction]

This device includes the SPICE data for simulating an N-channel unijunction transistor. Transistors may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



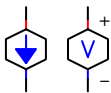
NC Push-button, NO Push-button, SPDT PB: Mixed-mode [Switches/Push Button]

These are Normally-Closed, Normally-Open and Single-Pole Double-Throw Push-Button switches. While digital simulation is running, the switch may be activated by clicking on it with the left mouse button and will remain in the activated position as long as the mouse button is held down. Multiple switches *of the same type* may be activated simultaneously if they have the same label in the Label-Value field. These switches cannot be activated while running an analog simulation, but simply act as a short or an open. Example circuit: CARS.CKT.



NLI Source, NLV Source: Analog [Analog/Power]

These are nonlinear current and voltage sources. See also: Dependent Sources. Example circuit: 741.CKT.



NPN Darling1, NPN Darling2, NPN Darling3: Analog [Active Components/Darlington]

These devices include the SPICE data for simulating an NPN Darlington Transistor. Transistors may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



NPN Trans:B, NPN Trans:C: Analog [Active Components/BJTs] (q)

These devices include the SPICE data for simulating an NPN bipolar junction transistor. Transistors may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: BJTs, Models, Initial Conditions. Example circuit: CEAMP.CKT.



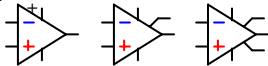
Op-Amp3: Analog [Linear ICs/OPAMPs]

This device is primarily intended for schematic purposes, but it includes SPICE data for simulating an ideal operational amplifier. See also: Voltage-Controlled Voltage Sources.



Op-Amp5, Op-Amp6, Op-Amp7: Analog [Linear ICs/OPAMPs] (o)

These devices include the SPICE data for simulating operational amplifiers. Op amps may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: ANALOG.CKT.



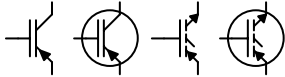
OptoIsolator: Mixed-mode [Optical Devices/Coupler]

This is an NPN optical isolator. In digital mode, the LED in this device operates exactly like a regular LED. When the LED is “on”, the logic level applied to the NPN transistor’s emitter will appear on its collector. When the LED is “off”, a 3-state level will appear on the transistor’s collector. In analog mode, the LED does not light, but the SPICE subcircuit information will be used for simulation. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuits: OPTO.CKT, STEPPER.CKT.



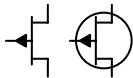
P-IGBT, P-IGBT:A, P-IGBT:B, P-IGBT:C: Analog [Active Components/IGBTs]

These devices include the SPICE data for simulating P-channel Insulated Gate Bipolar Transistors. Transistors may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Subcircuits.



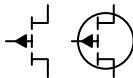
P-JFET, P-JFET:A: Analog [Active Components/JFETs] (J)

This device includes the SPICE data for simulating a P-channel junction field-effect transistor. JFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: JFETs, Models, Initial Conditions.



P-MESFET, P-MESFET:A: Analog [Active Components/MESFETs] (Z)

These devices include the SPICE data for simulating a P-channel MESFET (GaAsFET). MESFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MESFETs, Models.



P-DMOS 3T, P-DMOS 3T:A, P-DMOS 4T, P-DMOS 4T:A: Analog
[Active Components/MOSFETs Depl]

These devices include the SPICE data for simulating 3-terminal and 4-terminal P-channel Depletion Mode MOSFETs. MOSFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MOSFETs, Models, Initial Conditions.



P-EMOS 3T, P-EMOS 3T:A, P-EMOS 4T, P-EMOS 4T:A: Analog

[Active Components/MOSFETs Enh] (M)

These devices include the SPICE data for simulating 3-terminal and 4-terminal P-channel Enhancement Mode MOSFETs. MOSFETs may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: MOSFETs, Models, Initial Conditions.



Pentode, Pentode:H: Schematic [Active Components/Vacuum Tubes]

These devices are included for schematic purposes.

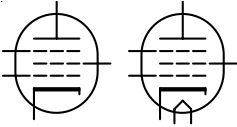


Photo Diode: Schematic [Optical Devices/Sensors]

This device is included for schematic purposes.



Photo NPN: Schematic [Optical Devices/Sensors]

This device is included for schematic purposes.



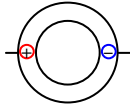
Photo Resist: Analog [Optical Devices/Sensors]

This device includes the SPICE data for simulating a standard resistor. The value of the resistor is specified in the Label/Value field. See also: Resistor.



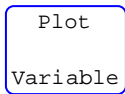
Piezo Buzzer: Mixed-mode [Transducers/Sound Device]

In digital simulation mode, the piezo buzzer will send a continuous stream of beeps to the PC's speaker when a low level is applied to its “-” terminal and a high level is applied to its “+” terminal. It contains SPICE data for use in analog mode as a capacitor. See also: Capacitors. Example circuit: SOUND.CKT.



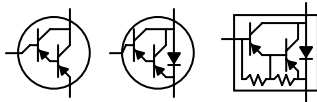
Plot Var: Analog [Analog/SPICE Controls]

This device provides a list of all of the plot variables for which data has been collected for the circuit. Place this device anywhere in the drawing window; no wires are needed. Click on the device with the Probe Tool, then select the variable you wish to plot from the list. To view multiple waveforms, SHIFT-click on the Plot Var device.



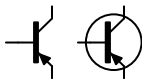
PNP Darling1, PNP Darling2, PNP Darling3: Analog [Active Components/Darlington]

This device includes the SPICE data for simulating an PNP Darlington Transistor. Transistors may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



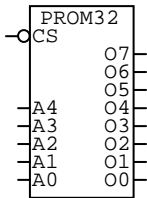
PNP Trans:B, PNP Trans:C: Analog [Active Components/BJTs] (Q)

These devices include the SPICE data for simulating an PNP bipolar junction transistor. Transistors may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: BJTs, Models, Initial Conditions. Example circuit: PUSHPULL.CKT.



PROM32: Mixed-mode [Digital/RAM-PROM]

This is a 32 x 8 PROM. Multiple PROMs may be individually programmed. Refer to *Edit PROM/RAM* in the Edit Menu chapter to program this device. PROM data is saved with the circuit and in macros. Example circuit: STEPPER.CKT.



Pulser: Digital [Digital/Instrument] (p)

The Pulser is a programmable pulse generator. Multiple Pulsers may be individually programmed. Refer to *The Pulser* in the Digital Logic Simulation chapter to program this device. Example circuit: SIM.CKT.



PUT: Analog [Active Components/Unijunction]

This device includes the SPICE data for simulating a programmable unijunction transistor. PUTs may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



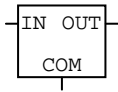
Reference3: Analog [Linear ICs/References]

This device includes the SPICE data for simulating a 3-pin programmable reference. References may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.



Regulator3: Analog [Linear ICs/Regulators]

This device includes the SPICE data for simulating a 3-pin voltage regulator. Regulators may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits.

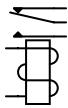


Relays

Relays are available in two forms. The SPDT Relay is a complete unit, containing both the coil and the contacts in a single device. Other relay devices are available in separate sections, allowing you to create your own relays with multiple pairs of contacts. By assigning the same label in the Label-Value field of the coil and the contacts, the coil can be assigned to switch any number of contacts.

SPDT Relay: Mixed-mode [Relays/Complete]

This device includes the SPICE data for simulating a Single-Pole Double-Throw relay. Relays may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. In digital mode, it switches the contact connections when a low level is applied to one terminal of the coil and a high level is applied to the other terminal. See also: Subcircuits. Example circuit: ALARM.CKT, ANRELAY.CKT.

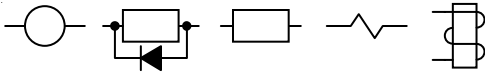


Circle, Polar Coil, Rectangle, Solenoid, Standard: Mixed-mode [Relays/Coil] (k)

These relay coils can be used in conjunction with any of the relay contacts. In order to control contacts with a given coil, set the Description field to the same string for both the coil and all contacts which the coil should control.

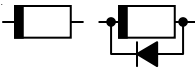
In **digital** simulation mode, the coils switch when a low level is applied to one terminal of the coil and a high level is applied to the other terminal. The pickup and release times for a coil can be individually programmed. The pickup time is the time delay following activation of the coil until the contacts close while the release time is the time following deactivation of the coil until the contacts open. To change the pickup and release times, double-click on a coil and then enter a SPICE comment of the following form into the SPICE Data field: *p=2 r=3. This comment will set the digital simulation mode pickup time to be 2 ticks and the release time to be 3 ticks.

In **analog** simulation mode the coil may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. Note: No SPICE data is included for the Latch Coil or the Polar Latch. Example circuit: LADDER.CKT.



Latch Coil, Polar Latch: Digital [Relays/Coil] (k)

These relay coils are similar to those described above with the following exceptions: 1) they do not contain SPICE simulation data for use in analog simulation and 2) being latched coils, they latch the contacts into the opposite position with a single pulse.



SPDT:A, SPDT:B, SPDT:C, SPST-NC, SPST-NO: Mixed-mode [Relays/Contacts] (K)

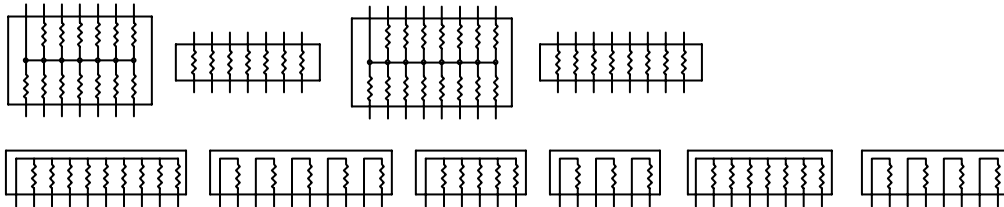
These relay contacts can be used in conjunction with any of the relay coils. In order to control contacts with a given coil, set the Description field to the same string for both the coil and all contacts which the coil should control. In analog simulation mode, the contacts may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. Example circuit: LADDER.CKT.



RDIP14, RDIP14:A, RDIP16, RDIP16:A: Analog [Passive Components/Resistors]

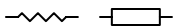
RSIP10, RSIP10:A, RSIP6, RSIP6:A, RSIP8, RSIP8:A

These devices include the SPICE data for simulating resistor packs. The value of the resistors is specified in the Label/Value field. See also: Resistors.



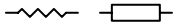
Resistor, Resistor:A: Mixed-mode [Passive Components/Resistors] (r)

These devices include the SPICE data for simulating a resistor. The value of the resistor is specified in the Label/Value field. In digital simulation mode, when connected directly to a +V or a Ground, it acts like a standard pull-up or pull-down resistor. Otherwise, it acts as an open in digital simulation mode. See also: Resistors. Example circuits: ROCKET.CKT, ANALOG.CKT.



SemiResistor, SemiResistorA: Analog [Passive Components/Resistors]

This device includes the SPICE data for simulating a semiconductor resistor. Semiconductor resistors may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits, Semiconductor Resistors.

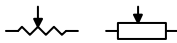


Var Resistor, Var ResistorA: Analog [Passive Components/Resistors] (R)

These devices include the SPICE data for simulating two fixed resistors by setting the following defaults:

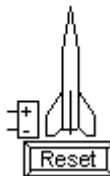
Label/Value:	10k
Spice Data:	%DA %1 %2 4k
	%DB %2 %3 6k

where the total resistance is $4k + 6k = 10k$. These values may be adjusted as required by the user. See also: Resistors.



Rocket: Mixed-mode [Digital/Output Devices]

This is an animated device similar to the Car, but with + and - input pins. When a high state is placed on the + input and a low state is placed on the - input, the rocket will fire. The reset button on the rocket returns the rocket to its starting position. Double-click on the rocket to program the travel distance (up to 1000 units). Different colors can be selected for each rocket. This device can also be used in Analog simulation mode, but is not animated. In this mode, a $1k\Omega$ resistor is placed across the + and - input pins. Example circuit: ROCKET.CKT.



Schottky: Analog [Active Components/Diodes]

This device includes the SPICE data for simulating a schottky diode. Diodes may be selected from a list of available models or new models may be added by the user. Double-click on the device to select the desired model. See also: Diodes, Models, Initial Conditions.



SCOPE: Digital [Digital/Instrument] (t)

A “SCOPE” is actually a tool which allows you to look at simulation waveforms as they are charted in the Waveforms window. Connect a scope (each scope must have a unique name) at each point in the circuit where you wish to see the states charted. Double-click on the device to edit the name. SCOPES are also used to identify the connecting nodes when creating a subcircuit drawing for use with the “Create SPICE subcircuit (.SUB)” script. Example circuit: SCOPE.CKT.



SCR: Analog [Active Components/SCRs]

This device includes the SPICE data for simulating an SCR (thyristor). SCRs may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: SCR.CKT.



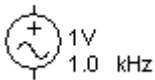
Shockley: Schematic [Schematic Symbols/Diodes]

This device is included for schematic purposes.



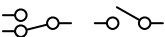
Signal Gen: Analog [Analog/Instruments] (g)

This is a fully programmable, multifunction analog signal generator which offers a variety of output waveforms. Multiple generators may be individually programmed. The minimum and maximum amplitudes of the waveform are stored in the Label-Value field and by default are displayed above the device symbol. Refer to the section *Multifunction Signal Generator* in the Analog Simulation chapter to program this device. Example circuit: ANALOG.CKT.



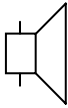
SPDT Switch, SPST Switch: Mixed-mode [Switches/Toggle] (S)

These are functional Single-Pole Double-Throw and Single-Pole Single-Throw switches. Multiple switches *of the same type* may be activated simultaneously (creating double-pole or triple-pole switches, etc.) if they have the same label in the Label-Value field. If you try to activate the switch while running an analog simulation, CircuitLogix will ask if you want to rerun the simulation with the switch in the new position. Example circuit: ALARM.CKT.



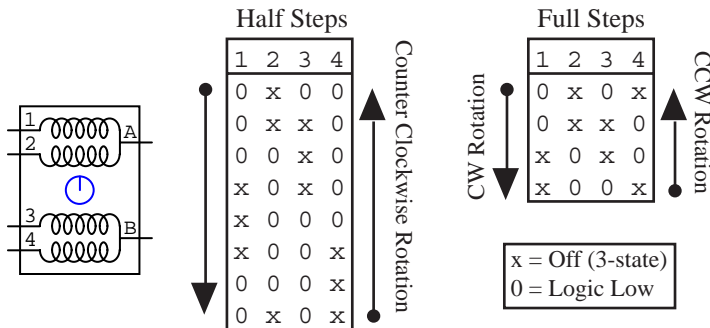
Speaker: Mixed-mode [Transducers/Sound Device]

In digital simulation mode, the speaker will send a single beep to the PC's speaker when a low level is applied to one of its terminals and a high level is applied to its other terminal. It will also send a single beep each time the high and low level are reversed or removed and then applied. It contains SPICE data for use in analog mode as a resistor. See also: Resistors. Example circuit: SOUND.CKT.



Stepper: Mixed-mode [Digital/Output Device]

This device simulates an eight position stepper motor. It is intended mainly for Digital simulation mode. It can be connected in unipolar or bipolar mode and can be driven in full or half steps. The following tables show how the motor is driven in unipolar mode with the A and B terminals connected to a logic high. In analog simulation mode, it is not animated, but treated as inductors and resistors in series. Example circuit: STEPPER.CKT.



Stoptlight: Mixed-mode [Digital/Output Device]

This device has 3 lights—red, yellow and green—with one input for each light. It is intended mainly for use in Digital simulation mode. The light will be on when its associated pin is in the high state and off when its pin is in the low state. In analog simulation mode, the lights do not light. Example circuit: DISPLAY.CKT.



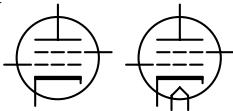
Terminal: Analog [Connectors/Misc]

This is a schematic symbol of a generic connector terminal point. It can be used in Analog simulations to connect digital simcode devices to a power bus. To do so, enter the name of the bus in the Terminal's bus data field.



Tetrode, Tetrode:H: Schematic [Active Components/Vacuum Tubes]

This device includes the SPICE data for simulating a vacuum tube tetrode. Tetrodes may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: VTPWRAMP.CKT.



Transformers

CircuitLogix provides two different types of transformer simulation. The first method which most closely represents the functionality of actual transformers uses subcircuits, consisting of a voltage-controlled voltage supply, a current-controlled current supply, winding resistances and leakage and magnetization inductors. The turns ratios for these devices is determined by the voltage and current gains of the supplies. The Trans1, Trans2 and Trans3 devices use this method.

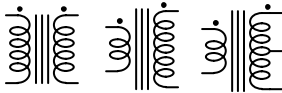
The second method uses coupled (mutual) inductors. This method is described in detail in the Analog Simulation chapter. The drawback to this method is that the impedance of the secondary winding is not reflected back into the primary. The Transformer and CTTransformer devices use this method. See Coupled Inductors for more information.

Note: For SPICE to operate properly, all nodes in a circuit require a DC path to ground. In circuits that use transformers, *both* sides of the transformer need a DC path to ground. This can be accomplished in various ways:

1. Ground can be connected directly to both sides of the transformer (see example circuit PS1.CKT).
2. Ground can be connected indirectly to both sides of the transformer through a resistor (see example circuit PS2.CKT).
3. Enable the RSHUNT option in the Analog Options dialog box which is accessed from the Options menu.

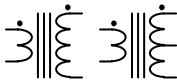
Trans1, Trans2, Trans3: Analog [Transformers/Subcircuit]

These devices include the SPICE data for simulating a transformer. Transformers may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Transformers, Subcircuits. Example circuits: PS1.CKT, PS2.CKT.



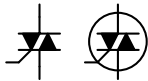
Transformer, CTTransformer: Analog [Transformers/Coupled Inductors]

These devices include the SPICE data for simulating a pair and a trio of coupled inductors. See also: Coupled Inductors, Transformers. Example circuit: VTPWRAMP.CKT.



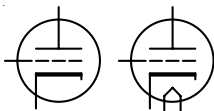
Triac:A, Triac:B: Analog [Active Components/Triacs]

This device includes the SPICE data for simulating a triac (thyristor). Triacs may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: TRIAC.CKT.



Triode, Triode:H: Schematic [Active Components/Vacuum Tubes]

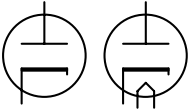
This device includes the SPICE data for simulating a vacuum tube triode. Triodes may be selected from a list of available subcircuits or new subcircuits may be added by the user. Double-click on the device to select the desired subcircuit. See also: Subcircuits. Example circuit: RIAAAMP.CKT.



Tunnel: Schematic [Schematic Symbols/Diodes]
 This device is included for schematic purposes.

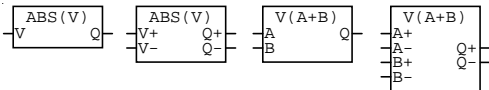


Vac Diode, Vac Diode:H: Schematic [Active Components/Vacuum Tubes]
 These devices are included for schematic purposes.



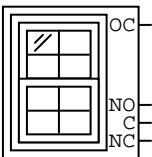
V-Math1, V-Math1 Ref, V-Math2, V-Math2 Ref: Analog [Math Functions/Voltage]

These devices provide direct access to SPICE's nonlinear dependent source math functions. They are set up in subcircuit format to provide easy selection of the math functions. V-Math1 and V-Math1 Ref provides access to the single-variable voltage functions (abs, cos, sqrt, etc.) V-Math2 and V-Math2 Ref provides access to the dual-variable voltage operations (+, -, *, / and ^). V-Math1 and V-Math2 require only a single input for each variable which is then referenced to ground. V-Math1 Ref and V-Math 2 Ref require both + and - inputs for each variable and provide both a + and - output. See also: I-Math, Nonlinear Sources. Example circuit: MATH1.CKT.



Window: Digital [Digital/I/O Device]

This is an animated device which simulates the opening and closing of a window. It has a single input (OC) which opens the window when high and closes it when low, or, it can be opened or closed by clicking on the window pane with the mouse. It also contains a single-pole double-throw switch that is activated when the window is opened or closed. Example circuit: ALARM.CKT.



Basic Logic Devices: Mixed-mode [Digital/*the minor class varies*]

Basic logic devices include a variety of common gates and flip-flops. Most of the gates are provided with their DeMorgan equivalents. Example circuit: 4X4.CKT.

Buffers:	The output will follow the input.
Tristate Buffers:	If the output is enabled, the output will follow the input.
Inverters:	If the input is high, the output will be low.
AND gates:	If all inputs are high, the output will be high. 2, 3 and 4 inputs.
OR gates:	If any input is high, the output will be high. 2, 3 and 4 inputs.
NAND gates:	If all inputs are high, the output will be low. 2, 3, 4 and 8 inputs.
NOR gates:	If any input is high, the output will be low. 2, 3, 4 and 8 inputs.
XOR gates:	If only one input is high, the output will be high. 2 inputs.
XNOR gates:	If only one input is high, the output will be low. 2 inputs.
D Flip-flops:	The output follows the input when clock occurs.
JK Flip-flops:	The output based on the J and K inputs when the clock occurs.
SR Flip-flops:	The output is high when set, low when reset.

Digital ICs: Mixed-mode [Digital ICs by Number/*the minor class varies*]

CircuitLogix provides a comprehensive library of commonly used Digital integrated circuits. Pin names are based on the HE4000 and 74LS00 families as defined by Philips Semiconductors, an international manufacturer of integrated circuits. The device number listed below indicates the logical function of the device. In digital simulation mode, the logic function of all families (74xx, 74LSxx, 74Sxx, 74Fxx, 74HCxx, etc.) is the same. In analog simulation mode, the characteristics of each family is unique. Characteristics of each part can be adjusted only within the min, max and typical databook values.

Note: *All of these devices are mixed-mode. O.C. = Open Collector outputs.*

4000	Dual 3-Input NOR Gate and Inverter
4001	Quad 2-Input NOR Gate
4002	Dual 4-Input NOR Gate
4006	18-Stage Static Shift Register
4008	4-Bit Binary Full Adder
4011	Quad 2-Input NAND Gate
4012	Dual 4-Input NAND Gate
4013	Dual D Flip Flop
4014	8-Bit Static Shift Register
4015	Dual 4-bit Static Shift Register
4017	5-Stage Johnson Counter
4018	Presetable Divide-by-n Counter
4019	Quad 2-Input Multiplexer
4020	14-Stage Binary Counter
4021	8-Bit Static Shift Register
4022	4-Stage Divide-by-8 Johnson Counter

4023 Triple 3-Input NAND Gate
 4024 7-Stage Binary Counter
 4025 Triple 3-Input NOR Gate
 4027 Dual JK Flip Flop
 4028 1-of-10 Decoder
 4029 Synchronous Up/Down Binary/Decade Counter
 4030 Quad Exclusive-OR Gate
 4031 64-Stage Static Shift Register
 4035 4-Bit Universal Shift Register
 4040 12-Stage Binary Counter
 4041 Quad True/Complement Buffer
 4042 Quad D-Latch
 4043 Quad R/S Latch (3-State)
 4044 Quad R/S Latch (3-State)
 4049 Hex Inverter
 4050 Hex Buffer
 4068 8-Input NAND Gate
 4069 Hex Inverter
 4070 Quad Exclusive-OR Gate
 4071 Quad 2-Input OR Gate
 4072 Dual 4-Input OR Gate
 4073 Triple 3-Input AND Gate
 4075 Triple 3-Input OR Gate
 4076 Quad D Register (3-State)
 4077 Quad Exclusive-NOR Gate
 4078 8-Input NOR Gate
 4081 Quad 2-Input AND Gate
 4082 Dual 4-Input AND Gate
 4085 Dual 2-Wide 2-Input AND-OR-Invert Gate
 4086 4-Wide 2-Input AND-OR-Invert Gate
 4093 Quad 2-Input NAND Schmitt Trigger
 4094 8-Stage Shift-and-Store Bus Register
 4104 Quad Low-to-High Voltage Translator (3-State)
 4502 Strobed Hex Inverter/Buffer
 4505 64-Bit, 1-Bit per Word Random Access Read/Write Memory
 4508 Dual 4-Bit Latch
 4510 BCD Up/Down Counter
 4511 BCD to 7-Segment Latch/Decoder/Driver
 4512 8-Input Multiplexer (3-State)
 4514 1-of-16 Decoder/Demultiplexer with Input Latches
 4515 1-of-16 Decoder/Demultiplexer with Input Latches
 4516 Binary Up/Down Counter
 4517 Dual 64-Bit Static Shift Register
 4518 Dual BCD Counter
 4519 Quad 2-Input Multiplexer
 4520 Dual 4-Bit Binary Counter
 4522 Programmable 4-Bit BCD Down Counter
 4526 Programmable 4-Bit Binary Down Counter

4531	13-Input Parity Checker/Generator
4532	8-Input Priority Encoder
4539	Dual 4-Input Multiplexer
4543	BCD to 7-Segment Latch/Decoder/Driver
4555	Dual 1-or-4 Decoder/Demultiplexer
4556	Dual 1-of-4 Decoder/Demultiplexer
4585	4-Bit Magnitude Comparator
4731	Quad 64-Bit Static Shift Register
7400	Quad 2-Input NAND Gate
7401	Quad 2-Input NAND Gate (O.C.)
7402	Quad 2-Input NOR Gate
7403	Quad 2-Input NAND Gate (O.C.)
7404	Hex Inverter
7405	Hex Inverter (O.C.)
7406	Hex Inverter Buffer/Driver (O.C.)
7407	Hex Inverter/Driver (O.C.)
7408	Quad 2-Input AND Gate
7409	Quad 2-Input AND Gate (O.C.)
7410	Triple 3-Input NAND Gate
7411	Triple 3-Input AND Gate
7412	Triple 3-Input NAND Gate (O.C.)
7413	Dual 4-Input NAND Schmitt Trigger
7414	Hex Inverter Schmitt Trigger
7415	Triple 3-Input AND Gate (O.C.)
7416	Hex Inverter Buffer/Driver (O.C.)
7417	Hex Buffer/Driver (O.C.)
7420	Dual 4-Input NAND Gate
7421	Dual 4-Input AND Gate
7422	Dual 4-Input NAND Gate (O.C.)
7425	Dual 4-Input NOR Gate with Strobe
7426	Quad 2-Input NAND Gate (O.C.)
7427	Triple 3-Input NOR Gate
7428	Quad 2-Input NOR Buffer
7430	8-Input NAND Gate
7432	Quad 2-Input OR Gate
7433	Quad 2-Input NOR Buffer (O.C.)
7437	Quad 2-Input NAND Buffer
7438	Quad 2-Input NAND Buffer (O.C.)
7439	Quad 2-Input NAND Buffer (O.C.)
7440	Dual 4-Input NAND Buffer
7442	BCD-to-Decimal Decoder (1-of-10)
7445	BCD-to-Decimal Decoder/Driver (O.C.)
7447	BCD-to-Seven-Segment Decoder/Driver (O.C.)
7448	BCD-to-Seven-Segment Decoder/Driver w/Pullups
7473	Dual J-K Flip-Flop
7474	Dual D Flip-Flop
7475	Quad Bistable Latch

7476 Dual J-K Flip-Flop
 7483 4-Bit Full Adder
 7485 4-Bit Magnitude Comparator
 7486 Quad 2-Input Exclusive-OR Gate
 7490 Decade Counter
 7492 Divide-by-Twelve Counter
 7493 4-Bit Binary Ripple Counter
 7495 4-Bit Shift Register
 7496 5-Bit Shift Register
 74107 Dual J-K Flip-Flop
 74109 Dual J-K Positive Edge-Trigger Flip-Flop
 74112 Dual J-K Edge-Triggered Flip-Flop
 74113 Dual J-K Edge-Triggered Flip-Flop
 74125 Quad Buffer (3-State)
 74126 Quad Buffer (3-State)
 74133 13-Input NAND Gate
 74136 Quad 2-Input Exclusive-OR Gate (O.C.)
 74138 1-of-8 Decoder/Demultiplexer
 74139 Dual 1-of-4 Decoder/Demultiplexer
 74147 10-Line to 4-Line Priority Encoder
 74148 8-Input Priority Encoder
 74151 8-Input Multiplexer
 74153 Dual 4-Line to 1-Line Multiplexer
 74154 1-of-16 Decoder/Demultiplexer
 74155 Dual 2-Line to 4-Line Decoder/Demultiplexer
 74157 Quad 2-Input Data Selector/Multiplexer
 74158 Quad 2-Input Data Selector/Multiplexer (Inverting)
 74160A BCD Decade Counter
 74161A 4-Bit Binary Counter
 74162A BCD Decade Counter
 74163A 4-Bit Binary Counter
 74164 8-Bit Serial-In Parallel-Out Shift Register
 74165 8-Bit Serial/Parallel-In, Serial-Out Shift Register
 74166 8-Bit Serial/Parallel-In, Serial-Out Shift Register
 74168 Synchronous BCD Decade Up/Down Counter
 74169 Synchronous 4-Bit Binary Up/Down Counter
 74173 Quad D Flip-Flop (3-State)
 74174 Hex D Flip-Flop
 74175 Quad D Flip-Flop
 74181 4-Bit Arithmetic Logic Unit
 7482 Lookahead Carry Generator
 74190 Presettable BCD/Decade Up/Down Counter
 74191 Presettable 4-Bit Binary Up/Down Counter
 74192 Presettable BCD/Decade Up/Down Counter
 74193 Presettable 4-Bit Binary Up/Down Counter
 74194 4-Bit Bidirectional Universal Shift Register
 74195 4-Bit Parallel Access Shift Register
 74197 Presettable 4-Bit Binary Ripple Counter

74199 8-Bit Parallel Access Shift Register
74240 Octal Inverter Buffer (3-State)
74241 Octal Buffer (3-State)
74242 Quad Inverting Transceiver (3-State)
74243 Quad Bus Transceiver (3-State)
74244 Octal Buffer (3-State)
74245 Octal Bus Transceiver (3-State)
74247 BCD-to-Seven-Segment Decoder/Driver (O.C.)
74248 BCD-to-Seven-Segment Decoder/Driver w/Pullups
74251 8-Input Multiplexer (3-State)
74253 Dual 4-Input Multiplexer (3-State)
74256 Dual 4-Bit Addressable Latch
74257 Quad 2-Line to 1-Line Data Selector/Multiplexer (3-State)
74258 Quad 2-Line to 1-Line Data Selector/Multiplexer (3-State)
74259 8-Bit Addressable Latch
74266 Quad 2-Input Exclusive-NOR Gate (O.C.)
74273 Octal D Flip-Flop
74280 9-Bit Odd/Even Parity Generator/Checker
74283 4-Bit Full Adder with Fast Carry
74290 BCD Decade Counter
74293 4-Bit Binary Ripple Counter
74298 Quad 2-Port Register
74352 Dual 4-Line to 1-Line Multiplexer
74353 Dual 4-Input Multiplexer (3-State)
74373 Octal Transparent Latch (3-State)
74374 Octal D Flip-Flop (3-State)
74375 Quad Bistable Latch
74377 Octal D Flip-Flop with Clock Enable
74378 Hex D Flip-Flop with Clock Enable
74568 BCD Decade Up/Down Synchronous Counter (3-State)
74569 4-Bit Binary Up/Down Synchronous Counter (3-State)

Index

Symbols

%

%="path\filename.ext" 2-21

%B 2-21

%D 2-20, 2-21

%L 2-19

%M 2-20

%N 2-19

%P 2-21

%pinorder 2-21

%S 2-20

%V 2-19

%X 2-21

+V 2-18, 5-17, 12-4

.IC 5-46, 12-4, A-2

.INCLUDE 2-21

.NODESET 5-45, A-2

.NS 12-4

.RAWFile 5-30, 5-31

13W3 12-9

1KRAM 12-4

4x4 Switch 12-5

4xxx 12-32

555 Timer 12-5

74xxx 12-32

A

ABSTOL 5-32, A-3

AC Analysis 5-28

AC Outlet 12-5

Access Faults 6-6

ACCT 5-35

ADCSTEP 5-37

Adding Shapes (Symbol Editor) 3-6

ADDSHUNTS A-1

Aliases 3-29

Always Set Defaults 5-30

Ammeter 5-38, 12-15

Analog

Analog Analyses 5-27

Analog Check Box 5-4

Analog Options 5-30, 5-55

Analog Options (Fault Simulation) 6-9

Analog Simulation 5-3

Analysis Selection (Fault Simulation) 6-9

Digital/Analog Mode 1-2, 10-4

Analyses

AC Analysis 5-28

Analog Options 5-30

Analysis Example 5-9

Analysis Probe 8-10

Analysis Selection (Fault Simulation) 6-9

DC Analysis 5-27

Operating Point Analysis 5-27

Run-Time Test Points 5-18

Set Defaults 5-29

SPICE for Windows 5-20

Test Points 5-18

Transient Analysis 5-28

AND gates 12-32

Antenna 12-6

Arc 3-9

Arrow Tool 1-2, 2-6

Arrow/Wire 7-10, 10-4

Ascii Display 12-6

Ascii Key 12-6

ASCIIOUTPUT 5-21, 5-30

Auto Designation 7-12

Auto Designation Prefix 2-16

Auto Refresh 7-12

Auto Routing 2-11

AUTOPARTIAL 5-35

B

BADMOS3 5-34

Battery 12-6

Behavioral Models 3-14

Bill of Materials *See* Parts Lists

Binary Output 5-30

Bitmap 2-24

BJTs 5-58

Black Box 6-11, 9-5

Index

Bode plotter 5-28
Boolean Expressions 5-34, 5-55
BOOLH 5-34, 5-55
BOOLL 5-34, 5-55
BOOLT 5-34, 5-55
Breakpoints 4-12, 4-13
Bridge 12-11
Browse 2-3, 2-8
Buffer 12-32
Buffer/Amp 12-7
Bus Data 2-18, 2-21, 5-17
Bus Labels 7-12, 11-5
Bus Wires 2-13, 8-7
BYPASS 5-35
Bypass Capacitors 5-17

C

CA7-Seg 12-7
Capacitors 5-48, 12-7
Car 12-8
Cascade Windows 11-7
CC7-Seg 12-7
CDA-5 12-8
CHGTOL 5-32
Circuit Display Data 7-12, 11-6
Circuit Faults 6-6, 6-7, 7-12
Circuit Info 10-3
Class Selected Device 9-6
Close 7-3
CMOS Devices 12-32
Coax 12-8
Coil 3T 12-8
Coil 5T 12-8
Coils 12-13
Color 4-4, 7-12, 11-3
Comparator5 12-8
Comparator6 12-8
Component Models 3-24, 12-5
Connection Area 7-11
Connector 12-9
Contacts 12-24
CONVABSSTEP 5-35
Convergence A-1

Convert ASCII Library 9-8
CONVLIMIT 5-35
CONVSTEP 5-35
Copy 6-8, 8-3
Copy Circuit 8-4
Copy Protection 1-11
Copy Waveforms 8-4
Copyright 1-12
Coupled Inductors 5-50, 12-29
Crystal 12-9
CTTransformer 12-30
Current Controlled Switches 5-50, 12-12
Current Sources 5-51, 12-12
Current-Controlled Current Sources 5-53, 12-12
Current-Controlled Voltage Sources 5-54, 12-13
CURRENTMNS 5-36
CURRENTMXS 5-37
Cursor Tools 10-4
Cursors 5-24
Curve Tracer 5-27
Cut 6-8, 8-3

D

D Flip-flops 12-32
Data Sequencer 4-9, 5-45, 8-9, 12-9
DB-15 12-9
DB-15HD 12-9
DB-25 12-9
DB-9 12-9
DC Analysis 5-27
Warnings A-2
DC Motor 12-10
DEFAD 5-32
DEFAS 5-32
Default Font 7-10
Default Transient Analysis 7-11
Defaults 7-9
Define New Macro 9-3
DEFL 5-32

Index

- DEFW 5-32
 - Delays 4-8, 11-5
 - Delete Items 8-3
 - Delete Macro 9-6
 - Delete Tool 1-2, 2-6, 6-7
 - DeMorgan Gates 12-32
 - Dependent Sources 5-54
 - Description 2-16, 2-21
 - Designation 2-15, 2-20, 8-6, 10-3, 11-6
 - Designation Prefix 2-16
 - Device Current 3-15
 - Device Data (Edit) 8-7
 - Device Data Display (Fault Simulation) 6-8
 - Device Display Data 11-6
 - Device Faults 6-3, 6-4, 6-6
 - Device Label 11-6
 - Device Library 2-8
 - Device Library (Fault Simulation) 6-9
 - Device Name 2-19, 11-6
 - Device Replacement (Fault Simulation) 6-8
 - Device Selection 2-3, 2-8, 3-14
 - Device Symbol 2-3, 2-9
 - DEVICE.LIB 12-3
 - Device/Plot Font 7-10
 - DGND 5-30
 - Diac 12-10
 - Digital
 - Digital Logic Simulation 4-3
 - Digital Options 4-12
 - Digital Options (Fault Simulation) 6-9
 - Digital Simcode Devices 1-6, 3-14, 5-61
 - Digital/Analog Button 1-2, 4-6, 5-15, 10-4
 - Edit Digital Parameters 8-7
 - Digital Oscilloscope 5-28
 - Diodes 5-57, 12-10, 12-26, 12-31
 - DIPs 3-7
 - Directories and Files 1-8, 7-11
 - Disable Circuit Options 6-7
 - Disclaimer 1-13
 - Display Hints 6-6, 6-8, 6-9
 - Display Scale 11-4
 - Display Title Box 7-12
 - Display Toolbar 7-10
 - Display Variable Names 5-23, 7-10, 10-4
 - Displaying Waveforms 5-23
 - Dots 11-5
 - Double-pole Switches 12-27
 - DPDT Switch 12-27
 - Drawing Grid 7-12, 11-6
 - Drawing Shapes (Symbol Editor) 3-5
 - Drive Capacity 5-36, 5-37
 - DRIVEMNS 5-36
 - DRIVEMXS 5-36
 - DRVMNTYMX 5-37
 - Duplicate 8-4
 - DVCC 5-30
 - DVDD 5-30
- ## E
- Edit Buffer 3-7
 - Edit Bus Connection 8-7
 - Edit Bus Wire Number 8-7
 - Edit Data Sequencer 8-9
 - Edit Device Data 8-7
 - Edit Digital Params 8-7
 - Edit Input/Output 8-9
 - Edit Macro 9-3
 - Edit Menu
 - Copy 8-3
 - Copy Circuit 8-4
 - Copy Waveforms 8-4
 - Cut 8-3
 - Delete Items 8-3
 - Duplicate 8-4
 - Edit Selected Item 8-6
 - Edit Bus Connection 8-7
 - Edit Bus Wire Number 8-7
 - Edit Data Sequencer 8-9
 - Edit Device Data 8-7
 - Edit Digital Params 8-7

Index

- Edit Input/Output 8-9
- Edit Multimeter 8-9
- Edit PROM/RAM 8-8
- Edit Pulser 8-8
- Edit Run-Time Test Point 8-8
- Edit Scope/Probe Name 8-10
- Edit Signal Generator 8-9
- Edit/Select Spice Model 8-8
- Mirror 8-5
- Move 8-3
- Paste 8-3
- Place Selected Label(s) 8-5
- Refresh Screen 8-5
- Rotate 90 8-5
- Select All 8-4
- Select Font 8-10
- Set Auto Designation 8-6
- Set Prop Delay(s) 8-5
- Straighten Selected Wire(s) 8-5
- Undo 8-3
- Edit Multimeter 8-9
- Edit PROM/RAM 8-8
- Edit Pulser 8-8
- Edit Run-Time Test Point 8-8
- Edit Scope/Probe Name 8-10
- Edit Selected Item 8-6
- Edit Signal Generator 8-9
- Edit/Select Spice Model 8-8
- Element Definitions 3-8
- Element List 3-7
- Ellipses 3-9
- Equivalent Circuits 3-24
- Errors 5-22, 7-8, A-1
- Exclude From Parts 2-22
- Exclude From PCB 2-22
- Exit 7-12
- Expand Macro 9-4, 9-6
- Exponential 5-42
- Export 7-4, 7-11
 - Circuit 2-24, 7-5
 - Waveforms 7-5

F

- Fault Example 6-11
- Fault Password 6-6, 6-10
- Faults 2-22, 6-3
- FETs 5-58, 12-16
- File Menu
 - Close 7-3
 - Exit 7-12
 - Export Circuit 7-5
 - Export Options 7-4
 - Export Waveforms 7-5
 - Merge 7-4
 - New 7-3
 - Open 7-3
 - Preferences 7-9
 - Print Circuit 7-6
 - Print Setup 7-6
 - Print Waveforms 7-6
 - Revert 7-3
 - Save 7-4
 - Save As 7-4
 - Script Functions 7-6
- Fill Color 3-6
- Find and Select 11-4
- Fit Circuit To Window 11-4
- Flip-Flops 12-32
- FM Signal 5-41
- Fonts 8-10
- Frequency Measurement 5-12
- Frequently Asked Questions B-1
- Fuse 12-11
- FW Bridge 12-11

G

- GaAs FETs 5-60
- Gates 12-32
- Gear Integration 5-31, A-3
- Generic Models 3-29
- Getting Started
 - Analog Circuit Examples 5-4

Index

Digital Circuit Examples 4-4
Schematic Example 2-3
GMIN 5-32, A-2
GMINSTEP 5-34
Graphics Windows 5-22
Grid 7-12, 11-6
Ground 2-18, 5-17, 12-11
Group Selected Items 2-23, 8-10
Groups 2-23, 8-10, 9-3

H

Hardware Requirements 1-3
Help 1-9
Help Tool 1-2
Hex Display 12-11
Hex Key 12-12
HIGH 6-3, 6-5, 6-10
Hints 6-5, 6-8, 6-9
Hotkeys 2-9

I

ISource 12-12
I->ISource 5-53, 12-12
I->Switch 5-50, 12-12
I->VSource 5-54, 12-13
I-Math 12-13
IMNTYMX 5-37
Importing SPICE Netlists 7-8
Independent Sources 5-51
Inductors 5-49, 12-13
Initial Conditions 5-29, 5-45, 12-4, A-2
Input 8-9, 12-13
Input Loading 5-36, 5-37
Input Loading (Digital Simcode Devices) 3-15
Installing CircuitLogix 1-3
Integration Method 5-31
Introduction 1-3
Inverter 12-32
ITL1 5-32, A-2
ITL2 5-32, A-2

ITL3 5-32
ITL4 5-33, A-3
ITL5 5-33

J

JFETs 5-58
JK Flip-flops 12-32

K

KEEPPOPINFO 5-34

L

Label 2-15
Label-Value 2-15, 2-19, 11-6
Labeling the Circuit 2-14
Lamp 12-13
Latched Relay Coil 12-24
LDMNTYMX 5-37
LED 12-14
Library Menus 12-3, 12-4
License Agreement 1-12
Line Color 3-6
Linearized Waveforms 5-25
Lines 3-8
Linking Files 3-16, 9-7
LIST 5-35
Loading 5-36, 5-37
LOADMNS 5-36
LOADMXS 5-36
Locking Macros 9-5
Logic Display 12-14
Logic Gates 12-32
Logic Switch 12-14
Lossless Transmission Lines 5-56
LossLessLine 12-14
Lossy Transmission Lines 5-57
LossyLine 12-15
LOW 6-3, 6-5, 6-10

M

Macro Button 1-2, 3-4

Index

- Macro Copier 9-7
 - Macro Libraries 1-4
 - Macro Lock 6-11, 9-5
 - Macro Utilities 3-13, 9-5
 - Macro Utilities Dialog
 - Class Selected Device 9-6
 - Delete Macro 9-6
 - Expand Macro 9-6
 - Model Data 9-7
 - Save Macro 9-6
 - Macromodels 3-24, 12-5
 - Macros 2-24, 3-3
 - Creating 3-10
 - Expanding 9-4
 - Saving 9-4
 - Macros Menu 9-3
 - Convert ASCII Library... 9-8
 - Define New Macro 9-3
 - Edit Macro Name 9-3
 - Expand Macro 9-4
 - Macro Copier 9-7
 - Macro Lock 9-5
 - Macro Utilities 9-5
 - Save ASCII Library... 9-8
 - Save Macro 9-4
 - Major Device Class 2-3, 2-9, 3-14, 9-6
 - Manual Routing 2-12
 - Math Functions 12-13, 12-31
 - MAXEVTITER 5-35
 - MAXOPALTER 5-35
 - MC1496 12-15
 - Measurement Cursors 5-24
 - Merge 7-4
 - MESFETs 5-60
 - Metafile 2-24, 3-6
 - MINBREAK 5-35
 - Minor Device Class 2-3, 2-9, 3-14, 9-6
 - Mirror 1-2, 2-8, 2-9, 6-8, 8-5
 - Mixed-mode Simulation 5-3
 - Model Data 9-7
 - Model Linking Files 3-16, 9-7
 - Models 2-16, 3-16, 3-21, 6-8, 8-8
 - 8
 - Modulators 12-15
 - MonoJack 12-15
 - MOSFETs 5-59
 - Move 8-3
 - Multipart Packages 2-23, 8-10
 - Multi-user Installation 1-7
 - Multimeter 5-27, 5-38, 8-9, 12-15
 - Multipliers 2-20, A-1
 - Mutual Inductors 5-50, 12-29
- ## N
- N-DMOS3T 12-16
 - N-DMOS4T 12-16
 - N-EMOS3T 12-17
 - N-EMOS4T 12-17
 - N-IGBT 12-16
 - N-JFET 12-16
 - N-MESFET 12-16
 - N-UJT 12-17
 - Naming Tool 1-2, 2-7
 - NAND gates 12-32
 - NC PushButton 12-17
 - Netlists 2-14, 7-8
 - New 7-3
 - NLI Source 5-54, 12-17
 - NLV Source 5-54, 12-17
 - NO PushButton 12-17
 - Node Numbers 2-21, 6-9, 7-12, 11-5
 - NODESET 5-45, A-2
 - Nonlinear Sources 5-54, 12-17
 - NOOPALTER 5-35
 - NOOPITER 5-34
 - NOR gates 12-32
 - Normal Size/Position 11-4
 - NPN Darling1 12-18
 - NPN Darling2 12-18
 - NPN Darling3 12-18
 - NPN Trans 12-18
 - Numerical Integration 5-31

Index

O

Offsetting Waveforms 5-24
Ohmmeter 5-38, 12-15
Op-Amp3 12-18
Op-Amp5 12-18
Op-Amp6 12-18
Op-Amp7 12-18
OPEN 6-3, 6-5, 6-10, 7-3
Operating Point Analysis 5-27
Warnings A-1
Options Menu 10-3
Analog Analyses 10-5
Analog Options 10-5
Arrow/Wire 10-4
Auto Designation 10-3
Auto Refresh 10-3
Circuit Info 10-3
Cursor Tools 10-4
Digital Options 10-5
Digital/Analog Mode 10-4
Display Toolbar 10-4
Repeat On 10-3
Reset Simulation 10-5
Run Simulation 10-5
Scope Probe 10-4
Step Simulation 10-5
Trace On 10-5
Waveforms 10-5
Optoisolator 12-19
OPTS 5-35
OR gates 12-32
Oscilloscope 5-28
Output 8-9, 12-13
Output Drive Capacity 5-36, 5-37
Output Drive: 3-15

P

P-DMOS3T 12-20
P-DMOS4T 12-20
P-EMOS3T 12-20
P-EMOS4T 12-20
P-IGBT 12-19

P-JFET 12-19
P-MESFET 12-19
Package 2-16, 2-21, 3-19, 11-6
Pad Designation 3-7
Page Breaks 7-12, 11-5
Parameter Passing 3-29
Parameters 2-19
Partial Derivatives 5-35
Parts Button 1-2, 2-3, 2-8
Parts Lists 2-24, 7-7
Password 6-6, 6-10
Paste 6-8, 8-3
Pattern Editor 4-10
PCB Netlists 2-18, 2-24, 3-20, 5-17, 7-8
Pentode 12-20
Persistent Waveform 5-26
Phone Jack 12-15
Phone Plug 12-15
Photo Diode 12-20
Photo NPN 12-20
Photo Resist 12-21
Piece-Wise 5-44
Piezo Buzzer 12-21
Pin Dots 7-12, 11-5
Pin Names 2-22, 3-7, 3-12, 11-6
Pin Numbers 2-22, 2-23, 3-7, 3-19, 11-6
Pin Placement 3-10
PIVREL 5-33
PIVTOL 5-33
Place Selected Label(s) 8-5
Placing Devices 2-8
Plot Var 7-8, 12-21
PNPDarling1 12-21
PNPDarling2 12-21
PNPDarling3 12-21
PNPTrans 12-22
Polar Cap 12-7
POLY 5-55
Polygons 3-9
Polylines 3-8
Position 11-4
Power Devices A-3

Index

Preferences 1-7, 7-9
Prefix 2-16
Print
 Circuit 2-24, 7-6
 Waveforms 7-6
Print Setup 7-6
Printed Circuit Boards 7-8
Probe Tool 1-2, 4-4, 4-6, 4-11, 5-16, 10-4
Project Installation 1-7
PROM 8-8
PROM32 12-22
Prompt to Save 7-10
Propagation Delays 3-15, 4-8, 5-36, 5-37, 8-5, 11-5
PROMNS 5-36
PROPMXS 5-36
Pulse Waveform 5-43
Pulser 4-8, 8-8, 12-22
PUT 12-22
PWL 5-44

Q

Quick Reference - Toolbar 1-2

R

RAM 8-8, 12-4
RAMPTIME 5-35
RAWFile 5-30, 5-31
RDIP14 12-25
RDIP16 12-25
Recalling Waveforms 5-25
Rectangles 3-9
Reference 5-16
Reference3 12-23
RefreshScreen 8-5, 10-3
Regulator3 12-23
Relay Coil 12-24
Relay Contacts 12-24

Relays 12-23
RELTOL 5-33, A-2
Repeat On 2-10, 7-10, 10-3, 12-3
Replace Device 6-6, 6-8, 6-9
Replacement Status 6-6, 6-8
Required Hardware 1-3
Required User Background 1-11
Reset 1-2, 4-6, 5-15, 10-5
Reset (Waveforms) 5-26
Resistors 5-47, 12-25
Revert 7-3
RF Devices A-3
Rocket 12-26
Rotate 90 1-2, 2-8, 2-9, 6-8, 8-5
Round Rects 3-9
RSHUNT 5-37
RSIP10 12-25
RSIP6 12-25
RSIP8 12-25
Run 1-2, 4-6, 5-15, 10-5
Run-Time Test Points 5-10, 5-16, 5-18, 5-20, 7-8, 8-8

S

Save 2-23, 7-4
Save As 7-4
Save ASCII Library 9-8
Save Circuit 6-9
Save Macro 3-14, 9-4, 9-6
Sawtooth 5-43
Scale 11-4
Scaling Waveforms 5-23
Schematic Capture 2-3
SCOPE 4-4, 4-11, 8-10, 12-26
Scope Probe 4-7, 4-11, 10-4
SCR 12-26
Script Functions 2-24, 6-8, 7-6
Select All 8-4
Select Colors 4-4, 7-12, 11-3
Selecting Items 2-11
SemiCapacitor 12-7
Semiconductor Capacitors 5-48
Semiconductor Resistors 5-47

Index

- SemiResistor 12-25
 - Set Defaults 5-29
 - Set Prop Delay(s) 8-5
 - Setup Button 5-24
 - Shockley 12-26
 - SHORT 6-4, 6-5, 6-10
 - Show Bus Labels 7-12, 11-5
 - Show Node Numbers 6-9, 7-12, 11-5
 - Show Page Breaks 2-24, 7-12, 11-5
 - Show Pin Dots 7-12, 11-5
 - Show Prop Delays 7-12, 11-5
 - Shunt Resistors 5-37
 - Signal Generator 5-39, 5-51, 5-52, 6-8, 8-9, 12-27
 - Signal Selection 6-8
 - Simcode Devices 1-6, 3-14, 5-61
 - Simulation
 - Analog/Mixed Mode 5-3
 - Digital Logic Simulation 4-3
 - Errors 5-22
 - Warnings 5-22
 - SIMWARN 5-37
 - Sine Wave 5-40
 - Single Click Connect 7-10
 - Singular Matrix Error 5-37
 - Size 11-4
 - SmartWires 2-12
 - SPDT PB 12-17
 - SPDT Relay 12-23
 - SPDT Switch 12-27
 - Speaker 12-27
 - SPICE 5-3
 - SPICE Convergence A-1
 - SPICE Data 2-19
 - SPICE Errors 5-22, A-1
 - SPICE Models 3-21, 8-8
 - SPICE Multipliers 2-20, A-1
 - SPICE Netlist 7-7
 - SPICE Netlists 2-24
 - SPICE Options 5-31, 6-9, A-1
 - SPICE Prefix 2-16
 - SPICE Prefix Character(s) 3-18
 - SPICE Subcircuits 7-7
 - SPICE Suggested Reading A-3
 - SPICE Warnings 5-22, A-1
 - SPICE Window 5-20
 - SPST Switch 12-27
 - SR Flip-flops 12-32
 - SRCSTEP 5-34
 - Step 1-2, 4-6, 5-15, 10-5
 - Stepper Motor 12-28
 - Stepping Device Values 7-8
 - Stereo Jack 12-15
 - Stop 1-2, 5-15
 - Stoptlight 12-28
 - Storing Waveforms 5-25
 - Straighten Selected Wires 8-5
 - Subcircuit Linking Files 3-16, 9-7
 - Subcircuits 2-21, 3-16, 3-24, 5-60, 6-8, 7-7
 - Supply Current 5-36, 5-37
 - Supply Voltage 5-22, 5-37
 - Switch Matrix 12-5
 - Symbol Editor 3-5, 3-24, 3-28
 - Symbols 2-24
- ## T
- Technical Support 1-10
 - TEMP 5-33
 - Terminal 2-18, 12-28
 - Test Points 5-11, 5-16, 5-18, 5-21, 5-22, 5-31, 10-5
 - Tetrode 12-29
 - Text Tool 1-2, 2-6
 - THold 5-22, 5-37
 - Thyristor 12-26, 12-30
 - Tile Windows 11-7
 - Timestep Too Small 5-33
 - Timing Violations 5-22, 5-37
 - Title Box 7-12, 11-7
 - TNOM 5-33
 - Toolbar 1-2, 2-6, 4-5, 5-15, 7-10, 10-4
 - TPMNTYMX 5-37
 - Trace (Symbol Editor) 3-8
 - Trace Feature 1-2, 4-4, 4-7, 5-16, 6-9, 10-5

Index

TRANMNS 5-36
TRANMXS 5-36
Trans1 12-30
Trans2 12-30
Trans3 12-30
Transformers 5-50, 12-29, 12-30
Transient Analysis 5-28, 7-11
 Warnings A-2
Transistors 5-58, 12-16, 12-19, 12-21
Transition Time 3-15, 5-36, 5-37
Transmission Lines 5-56, 12-14
Trapezoidal Integration 5-31, A-3
TraxMaker 2-16, 2-18, 3-13, 3-20, 3-29, 5-17, 7-8
 Pin Numbers/Pad Designations 3-7
TRec 5-22, 5-37
Triac 12-30
Triangle 5-43
Triode 12-30
Tristate Buffers 12-32
Troubleshooting A-1
TRTOL 5-34
TRYTOCOMPACT 5-34
TSetup 5-22, 5-37
TTL Devices 12-32
TTMNTYMX 5-37
Tunnel 12-30
Tw 5-22, 5-37
TwinLead 12-8
TwistedPair 12-8

U

U.S. Government Restricted Provisions 1-14
UIC 5-29, A-2
Undo 8-3
Updating
 Macro Libraries 1-4
 Pre-4.0 Circuits and Macro Libraries 1-5
 Pre-5.0 Digital Circuits 1-6

 Previous Versions 1-4
URC Lines 5-57, 12-15
User-defined Schematic Symbols 2-24
USER.LIB 1-7, 3-10, 3-14, 9-3, 9-7, 12-3

V

VSource 12-12
V->I Source 5-52, 12-12
V->Switch 5-50, 12-12
V->V Source 5-53, 12-13
V-Math 12-31
Vac Diode 12-31
Vacuum Tubes 12-20, 12-29, 12-30, 12-31
Value 2-15, 11-6
Value Stepping 7-8
Var Capacitor 7-9, 12-7
Var Inductor 7-9, 12-13
Var Resistor 7-9, 12-25
Varactor 12-10
Variable Names 10-4
Vcc 5-17
View Menu 11-3
 Cascade Windows 11-7
 Circuit Display Data 11-6
 Device Display Data 11-6
 Display Scale 11-4
 Drawing Grid 11-6
 Find and Select 11-4
 Fit Circuit To Window 11-4
 Normal Size/Position 11-4
 Select Colors 11-3
 Show Bus Labels 11-5
 Show Node Numbers 11-5
 Show Page Breaks 11-5
 Show Pin Dots 11-5
 Show Prop Delays 11-5
 Tile Windows 11-7
 Title Box 11-7
 Windows 11-7

Index

Virtual Instruments

- Bode Plotter 5-22
- Curve Tracer 5-22
- Data Sequencer 4-9, 5-45
- Digital Multimeter 5-22, 5-38
- Digital Oscilloscope 5-22
- Logic Analyzer 4-11
- Multifunction Signal Generator 5-39
- VNTOL 5-34, A-3
- Voltage Controlled Switches 5-50, 12-12
- Voltage Sources 5-51, 12-12
- Voltage-Controlled Current Sources 5-52, 12-12
- Voltage-Controlled Voltage Sources 5-53, 12-13
- Voltmeter 5-38, 12-15

W

- Warning Messages 5-22, 5-37, A-1
- Warranty 1-14
- Waveforms 1-2, 4-4, 4-7, 4-11, 5-16, 8-4, 10-5
- Window (Device) 12-31
- Wire Tool 1-2, 2-6, 6-7
- Wiring the Circuit 2-11
- Word Generator 4-9, 5-45
- Wrong value 6-4

X

- XNOR gates 12-32
- XOR gates 12-32
- XSpice 3-14

Z

- Zener Diode 12-10
- Zoom Tool 1-2, 2-7, 11-4

Index