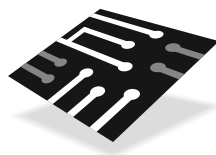


# multiSIM 2001

Schematic Capture, Simulation & Programmable Logic

---

*Educator User Guide*



**Electronics**  
WORKBENCH

DESIGN SOLUTIONS FOR EVERY DESKTOP

multiSIM™ and Electronics Workbench™ copyright © 1989, 1992-2000 Interactive Image Technologies Ltd. All rights reserved.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Portions of this product are provided under license from:

- Green Mountain Computing Systems
- Metamor, Inc.

ISBN 1-55169-088-8 Rev. 2

© 2000 Interactive Image Technologies Ltd. All rights reserved. Published December 2000.  
Printed in Canada.



# Preface

Congratulations on choosing Multisim from Electronics Workbench. We are confident that it will deliver years of increased productivity and superior designs.

Electronics Workbench is the world's leading supplier of circuit design tools. Our products are used by more customers than those of any other EDA vendor, so we are sure you will be pleased with the value delivered by Multisim, and by any other Electronics Workbench products you may select.

## Documentation Conventions

When Multisim manuals refer to a toolbar button, an image of the button appears in the left column.

The manuals show circuits in black and white, although Multisim is configured to use color by default. (You can change the color scheme.)



When you see the icon in the left column, the functionality described is only available in certain version of Multisim, or to users who have purchased add-in modules.

Multisim manuals use the convention **Menu/Item** to indicate menu commands. For example, “**File/Open**” means choose the **Open** command from the **File** menu.

Multisim manuals use the convention of an arrow (➤) to indicate the start of procedural information.

Multisim manuals use the construction CTRL-KEY and ALT-KEY to indicate when you need to hold down the “Ctrl” or “Alt” key on your keyboard and press another key.

## The Multisim Documentation Set

Multisim documentation consists of a *Getting Started and Tutorial* manual, this *User Guide*, and on-line help. All Multisim users receive PDF versions of the *Getting Started and Tutorial* manual and the *User Guide*. Depending on your version of Multisim, you may also receive a printed version of the manuals.

## Getting Started and Tutorial

The *Getting Started and Tutorial* manual introduces you to the Multisim interface. It also offers an introductory tutorial that takes you through the stages of circuit design, simulation, analysis and reporting.

## User Guide

The *User Guide* describes Multisim and its many functions in detail. The manual is organized based on the stages of circuit design, and explains all aspects of using Multisim, in detail.

## On-Line Help

Multisim offers a full help file system to support your use of the product. Choose **Help/Multisim Manual** to display the help file that explains the Multisim program in detail, or choose **Help/Multisim Reference** to display the help file that contains reference material (from the appendices) such as details on all the components families provided with Multisim. Both are standard Windows help files, offering a table of contents and index.

In addition, you can display context-sensitive help by pressing F1 from any command or window, or by clicking the **Help** button on any screen that offers it.

## Adobe PDF Files

Both the *Getting Started and Tutorial* manual and the full *User Guide*, including appendices, are provided on the Multisim CD as Adobe PDF files and are accessible from the Multisim program folder on the Windows Start menu. To open PDF files, you will need Adobe's free Acrobat Reader program, available for download at [www.adobe.com](http://www.adobe.com).

## License Agreement

Please read the license agreement included in the *Multisim Getting Started and Tutorial Manual* carefully before installing and using the software contained in this package. By installing and using the software, you are agreeing to be bound by the terms of this license. If you do not agree to the terms of this license, simply return the unused software within thirty days to the place where you obtained it and your money will be refunded.

# Chapter 1

## Introduction

### 1.1 About this Chapter

This chapter briefly introduces you to this manual and to Multisim 2001 itself. It also provides a summary of Multisim's features and in which edition they are available.

### 1.2 About this Manual

This manual is written for all Multisim 2001 users. It explains, in detail, all aspects of the Multisim 2001 product. The manual contains both:

- chapters, which explain features and functions, and which are organized based on the Design Bar buttons
- appendices, which contain reference-type information.

Depending on your Multisim 2001 edition, this manual, or just its appendices, may be available only on-line, not in print.



This manual describes a number of functions that are available only in some edition of Multisim 2001, or to users who have purchased optional modules. Such functions are identified by the icon shown in the column to the left. To order optional modules, contact Electronics Workbench. For a list of features in each product, see “1.4 Multisim 2001 Educational Editions” on page 1-2.

This manual assumes that you are familiar with Windows applications and know how, for example, to choose a menu from a command, use the mouse to select an item, and enable/disable an option box. If you are new to Windows, see your Windows documentation for help.

## 1.3 What is Multisim 2001?

Multisim 2001 is the latest generation of the world's most popular electronic design and education software from Electronics Workbench. It is a complete system design tool that offers a large component database, schematic entry, full analog/digital SPICE simulation, VHDL/Verilog design entry/simulation, FPGA/CPLD synthesis, RF capabilities, postprocessing features and seamless transfer to PCB layout packages such as Ultiboard, also from Electronics Workbench. It offers a single, easy-to-use graphical interface for all your design and analysis needs.

## 1.4 Multisim 2001 Educational Editions

Multisim 2001 is available in three editions for the educational community: Education, Lab, and Student.

The Education edition is for use by professors, teachers, instructors, tutors etc., and is typically used for the creation of demonstrations, examples, assignments, or tests.

The Lab edition is for use by students at school in a multi-station laboratory environment. It is available either in networked or standalone configurations.

The Student edition is for use by students for home study on their personal computer.

# Chapter 2

## User Interface

### 2.1 About this Chapter

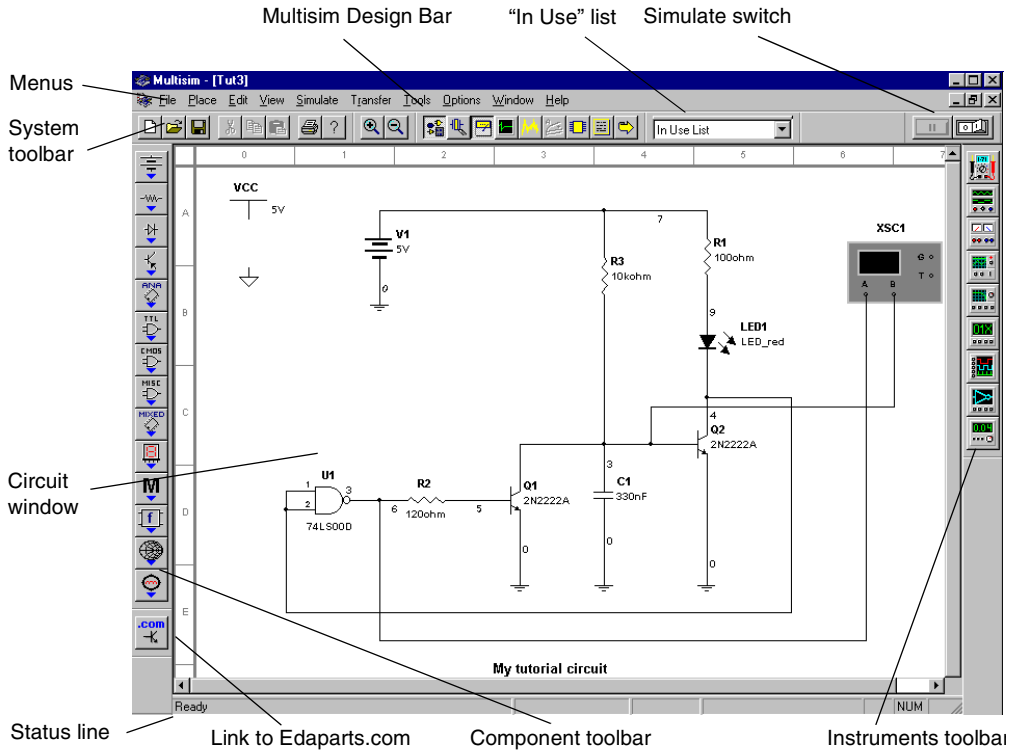
This chapter explains the basic operation of the Multisim 2001 user interface, and briefly describes all available Multisim 2001 commands.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

## 2.2 Introduction to the Multisim Interface

Multisim's user interface consists of the following basic elements:



**Note** Your circuit window may, by default, have a black background; however, for the purposes of this document, we show a white background. To change the background color, see “2.4.3 The Circuit Tab” on page 2-6.

**Menus** are, as in all Windows applications, where you find commands for all functions.

The **system toolbar** contains buttons for commonly-performed functions, as described in “2.5 System Toolbar and Zoom Toolbar Buttons” on page 2-13.

The **instruments toolbar** contains buttons for each instrument, as described in “6.2 Introduction to the Multisim Instruments” on page 6-1.

The link to **Edaparts.com** launches your Web browser loaded with the Electronic Workbench Edaparts.com Web site. You can then navigate the site to download parts, as described in “4.4 Using Edaparts.com” on page 4-24.

The **zoom toolbar** allows you to zoom in and out on the active circuit.

The **Multisim Design Bar** is an integral part of Multisim, and is explained in more detail in “2.3 Introduction to the Design Bar” on page 2-3.

The **“In Use” list** lists all the components used in the active circuit, for easy re-use.

The **component toolbar** contains Parts Bin buttons that let you open Parts Bins (which, in turn, contain buttons for each family of components), as described in “3.5.1 Choosing a Component and Using the Browser Screen” on page 3-6.

The **circuit window** is where you build your circuit designs.

The **status line** displays useful information about the current operation and a description of the item the cursor is currently pointing to.

The **simulate switch** is an easy way to run, stop or pause the simulation of your circuit.

## 2.3 Introduction to the Design Bar

The Design Bar is a central component of Multisim, allowing you easy access to the sophisticated functions offered by the program. The Design Bar guides you through the logical steps of building, simulating, analyzing and, eventually, exporting your design. Although Design Bar functions are available from conventional menus, this manual assumes you are taking advantage of the ease of use offered by the Design Bar.

**Note** If the simplified version option is selected, the Design Bar is hidden. For details on the simplified version option, “13.4.1.2 Simplified Version” on page 13-7.



The Component design button is selected by default, since the first logical activity is to place components on the circuit window. For details on the functions associated with this button, see Chapter 4, “Components”.



The Component Editing button lets you modify the components in Multisim, or add components. For details on the functions associated with this button, see Chapter 5, “Component Editing”.



The Instrument design button is selected by default and lets you attach instruments to your circuit. For details on the functions associated with this button, see Chapter 6, “Instruments”.



The Simulate button runs/pauses/stops the simulation. The green sine wave line moves while simulation is running. For details on functions associated with simulation, see Chapter 7, “Simulation”.



The Analysis button lets you choose the analysis you want to perform on your circuit. For details on the functions associated with this button, see Chapter 8, “Analyses”.



The Postprocessor button lets you perform further operations on the results of your simulation. For details on the functions associated with this button, see Chapter 9, “Postprocessor”.



The VHDL/Verilog HDL button allows you to work with VHDL modeling (not available in all editions). For details on the functions associated with this button, see Chapter 10, “HDLs and Programmable Logic”.



The Reports button lets you print reports about your circuits (list of components, component details, instruments). For details on the functions associated with this button, see Chapter 11, “Reports”.



The Transfer button lets you communicate with and export to PCB layout programs, such as Ultiboard, also from Electronics Workbench. You can also export simulation results to programs such as MathCAD and Excel. For details on the functions associated with this button, see Chapter 12, “Transfer/Communication”.

## 2.4 Customizing the Interface

### 2.4.1 About Customizing the Interface

You can customize virtually any aspect of the Multisim interface, including the toolbars, the colors used in your circuit, page size, zoom level, auto-backup interval, symbol set (ANSI or DIN) and printer setup. Your customization settings are saved individually with each circuit file you use so you could, for example, have one color scheme for one circuit and another for a different circuit. You can also override the settings for individual instances (for example, change one particular component from red to orange) or for the entire circuit.

Most customization is done through **Options/Preferences**. This command allows you to either change the settings for the current circuit file only, or create default settings that affect both the current circuit and all circuits you subsequently create. For example, if your current circuit shows component labels and you save those settings as the default, when you choose **File/New** and create a new circuit, that circuit will be set to show component labels as well. Changing the default settings will **not** affect circuits you have already created and saved. However, if your current circuit shows component labels and you simply change it to hide such labels, the change affects only your current circuit.

Customization changes done through the pop-up menu, as described in the following sections, affect the current circuit only.



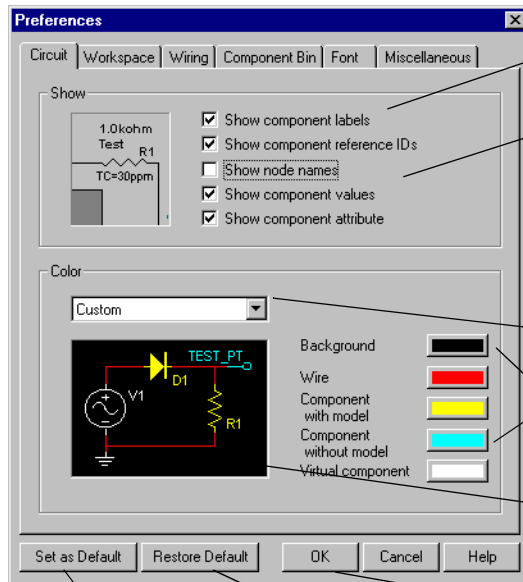
## 2.4.2 General Instructions for Customizing the Interface Using the Preferences Screen

This section explains general procedures for setting preferences. The following sections describe details of setting specific options.

- To set the preferences for circuits, choose **Options/Preferences**. The Preferences screen appears, offering you the following six tabs:
  - the Circuit tab, where you set the way your circuit and its components appear on the screen
  - the Workspace tab, where you set the appearance and behavior of the circuit window
  - the Wiring tab, where you set the wire width and autoroute options
  - the Component Bin tab, where you set the symbol set, behavior of the Component toolbar and its Parts Bins, and component placement
  - the Font tab, where you select the font, font size and style for text elements on the circuit
  - the Miscellaneous tab, where you set the auto-backup options, default path and folder, simulation level and PCB options.
- 1. Select the desired tab.
- 2. Set the desired customization options. The specific options and settings available in the tabs are described in the following sections of this chapter.
- 3. To save your options for the current circuit *only*, click **OK**.
  - or
  - To save your options as the default, affecting both the current and subsequent circuits, click **Set as Default**, then click **OK**.
  - or
  - To restore your previously saved settings (if you don't like the changes you made and would like to start over again), click **Restore Default** and then continue making changes.
  - or
  - To cancel your changes and close the screen, click **Cancel**.

## 2.4.3 The Circuit Tab

The options on this tab control the way your circuit and its components appear on the screen, and the level of detail which appears. Multisim comes with several color schemes that affect the circuit window background color, wire color, and component color. You can also develop your own color scheme to meet your individual needs.



Shows the results of enabling the options on the right.

Enable those items you want shown. You can override your choices for a particular component, as described in "3.9.1 Displaying Identifying Information about a Placed Component" on page 3-18.

Select a built-in color scheme.

Set up a custom color scheme (only if "Custom" selected as scheme).

Shows the currently selected color scheme.

Saves the selected settings for current and subsequent circuits.

Restores the default settings to previously saved ones.

Saves the settings for the current circuit only.

**Note** To set the circuit options for only the *current* circuit, right-click on the circuit window and choose either **Show**, which displays a screen identical to the show options in the Circuit tab of the Preferences screen (shown above), or **Color**, which displays a screen identical to the color options in the tab.

- To use one of the built-in color schemes, choose the scheme from the drop-down list. A representation of the scheme's settings appears in the preview box below the list.

- To create a custom color scheme, choose **Custom** from the drop-down list.
  1. Click on the color bar next to any item. A Color selector screen appears.
  2. Click on the color you want to use for that item and click **OK**. You are returned to the Preferences screen. The results of your choice appear in the preview box.
  3. Repeat until all your color settings are made.
  4. To save your settings for the current circuit and close the screen click **OK**.

or

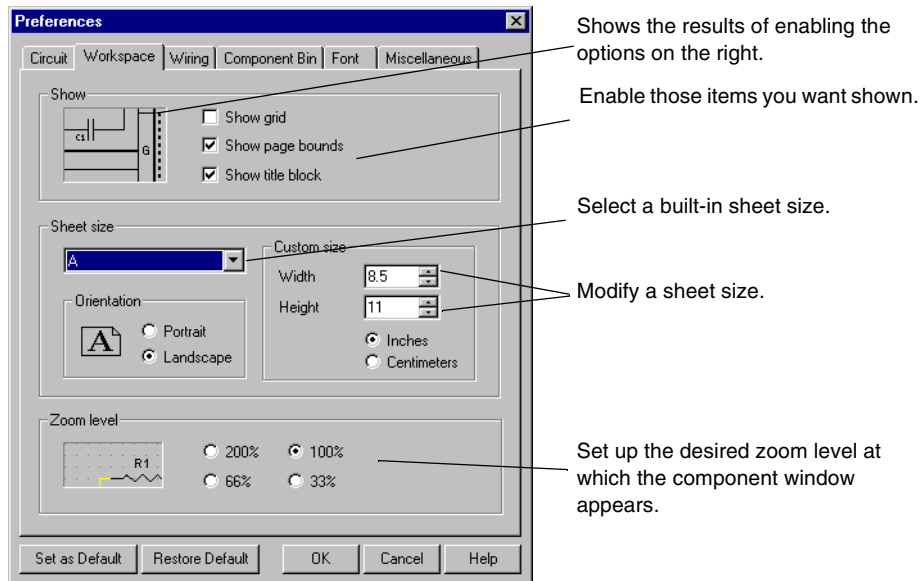
To restore the default settings to the current circuit, click **Restore Default**.

or

To save your settings as the default for the current and subsequent circuits, click **Set as Default**, then click **OK**.

## 2.4.4 The Workspace Tab

The options on this tab determine the appearance and behavior of the circuit window.

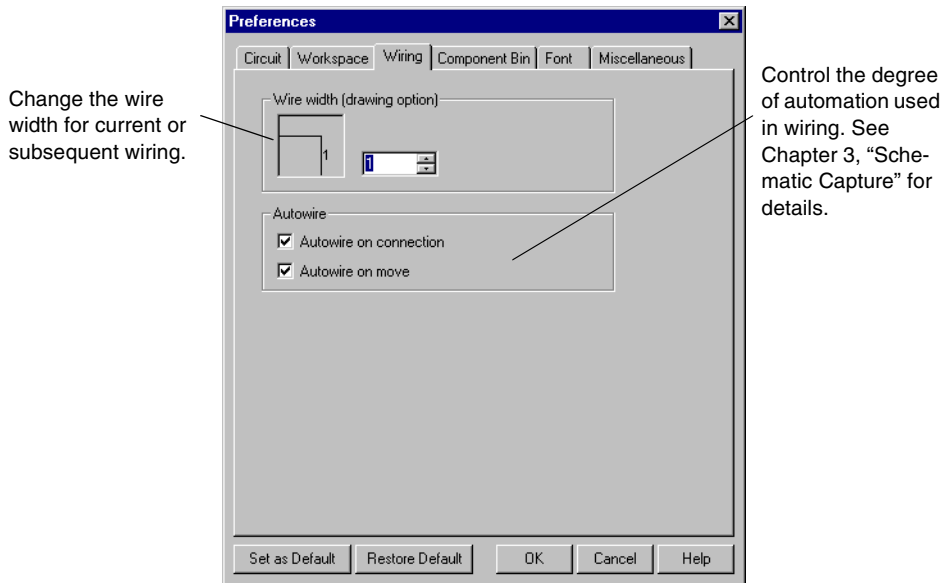


**Note** You can also set the circuit window options for the *current* circuit only by right-clicking on the circuit window and choosing either **Show Grid**, **Show Page Bounds**, or **Show Title Block and Border** from the menu that appears.

Multisim comes with several sheet sizes that you can use for laying out your circuit. You can modify any of the settings of these sizes.

## 2.4.5 The Wiring Tab

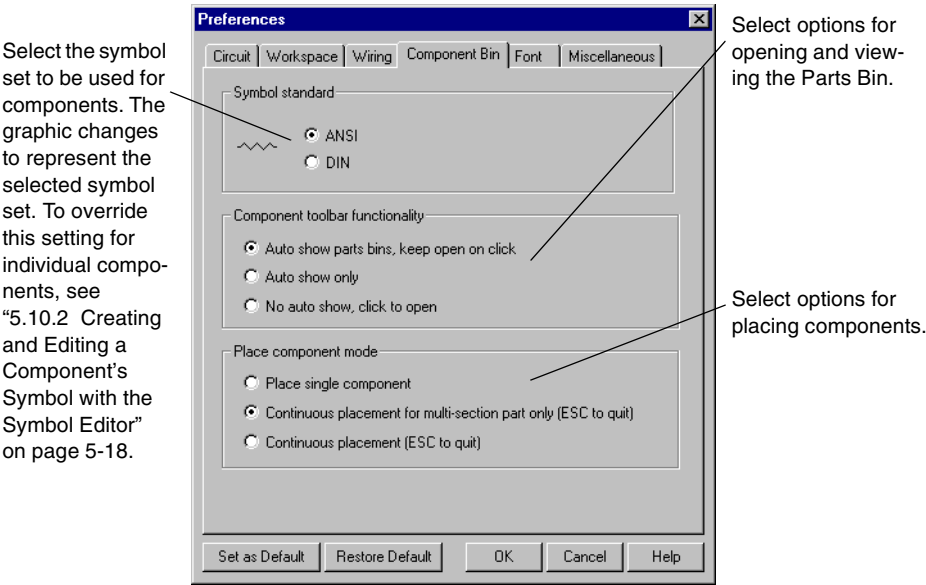
The options on this tab control the wire width and autowire options.



**Note** You can also set the wiring options for the *current* circuit only by right-clicking on the circuit window and choosing **Wire width** from the menu that appears which displays a screen identical to the wire options in the tab.

## 2.4.6 The Component Bin Tab

The options on this tab determine the symbol set, behavior of the Component toolbar and its Parts Bins, and component placement.



- To set the options for opening and viewing the Parts Bin select from the following options in the Component toolbar functionality section of the tab:

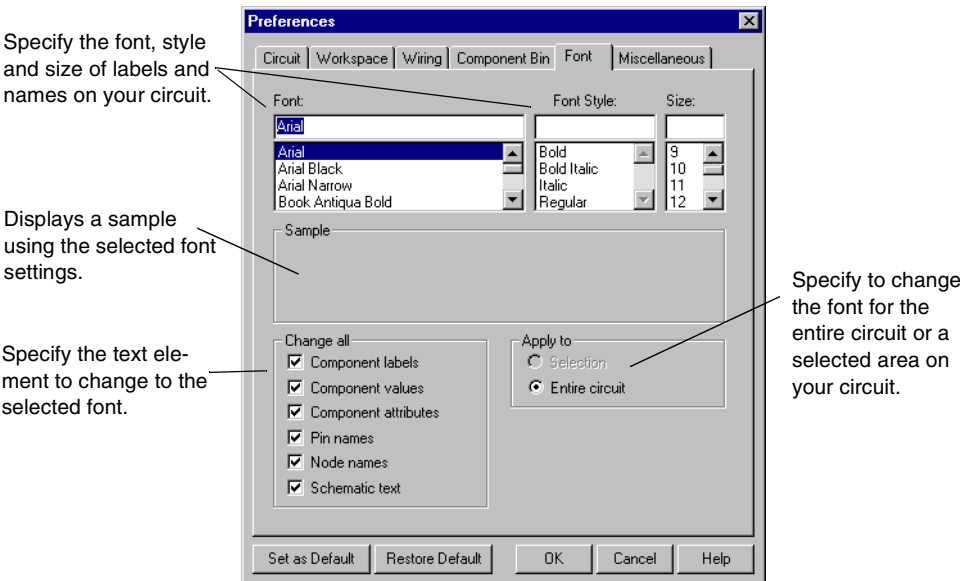
Component Toolbar Option	Description
Auto show parts bin, keep open on click	Parts bin opens when you hover over a Parts Bin button. Clicking on a Parts Bin button will keep the Parts Bin open even after a component is placed.
Auto show only	Parts bin opens when you hover over a Parts Bin button and closes after a component is selected.
No auto show, click to open	Parts bin opens when you click on a Parts Bin button and stays open after a component is placed.

- To set the options for placing components select from the following options in the Place component mode section of the tab:

Place Component Option	Description
Place Single Component	Allows you to place one selected component at a time.
Continuous Placement for Multipart Only	Allows you to place multiple instances of a component by continuing to click once the component is placed. End continuous placement by pressing ESC. For example, 7400N has 4 NAND gates under one component, so using this option means each time you place a 7400N you place a different one of its NAND gates.
Continuous Placement	Allows you to place several consecutive components of the same type by continuing to click once the component is placed. End continuous placement by pressing ESC.

## 2.4.7 The Font Tab

The options on this tab set the font for component labels and values, node and pin names, schematic text and component attributes.



**Note** You can also set the font options for the *current* circuit only by right-clicking on the circuit window and choosing **Font** from the menu that appears. This displays a screen identical to the font options in the tab.

- To change the font for any text element of the circuit:
  1. Select the font, font style and font size for the desired elements.
  2. Select the elements you want to change from the following options in the Change All section of the tab:

Font Option	Description
Component labels	The Component label assigned by Multisim and identified in the Label tab of the Component properties screen. For details on changing this label see “3.11.1 Modifying Component Labels and Attributes” on page 3-25.
Component values	The value being used by the component and identified in the Values tab of the Component properties screen. For details on viewing and modifying this value see “3.9.2 Viewing a Placed Component’s Value/Model” on page 3-19.
Component attributes	Descriptive information you add to specific components using the Component properties screen. For details see “3.11.1 Modifying Component Labels and Attributes” on page 3-25.
Pin names	The number automatically assigned to a pin when it is placed on your circuit. For details on changing this number see “5.10.2.6 Pins” on page 5-21.
Node names	The number automatically assigned to a node when it is placed on your circuit. For details on changing this number see “3.11.2 Modifying Node Numbers” on page 3-26.
Schematic text	Notes that you add to your circuit by using the Place/Place Text command. For details see “3.11.4 Adding Miscellaneous Text” on page 3-27.

## 2.4.8 The Miscellaneous Tab

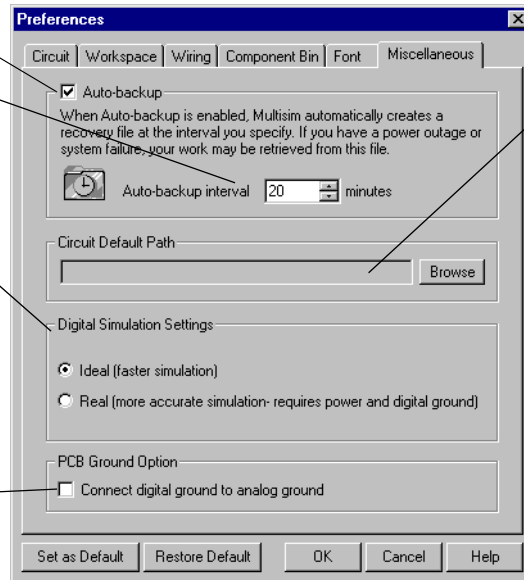
The options on this tab control the auto-backup options, default path and folder, digital simulation settings and PCB Ground options.

Enable or disable auto-backup.

Specify the interval at which auto-backup will be performed

Set the realism of the output when simulating digital components. For details, see “7.3.1 Simulating Circuits with Digital Components” on page 7-3.

Select this option to make digital and analog ground the same when exporting to PCB layout packages.



Set the default path for saving circuits.

## 2.4.9 Other Customization Options

You can also customize the interface by showing or hiding, dragging to a new location and, optionally, resizing any of the following:

- System toolbar
- Design Bar
- “In Use” list
- Instruments toolbar
- Zoom toolbar

These changes apply to all circuits you are working with. Moved or resized items will return to that location and size when Multisim is next opened.

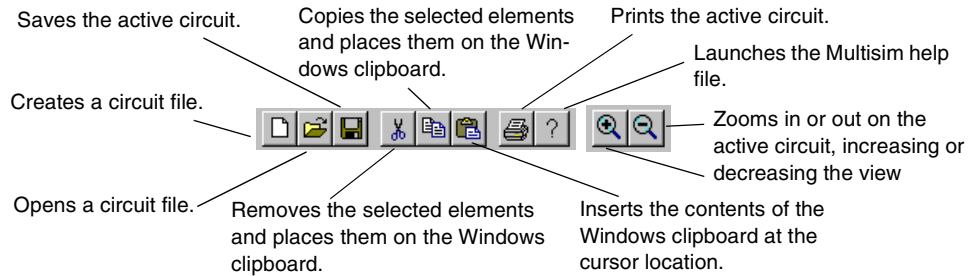
You can use **Options/Simplified Version** to hide the Design Bar and several analyses and instruments. For more details on the simplified version option, see “13.4.1.2 Simplified Version” on page 13-7.



Finally, you can use the **View** menu to display or hide various elements, as described in “2.6.3 View Menu” on page 2-18.

## 2.5 System Toolbar and Zoom Toolbar Buttons

The system toolbar and zoom toolbar offer the following standard Windows functions:



## 2.6 Menus and Commands

This section explains, in brief, all available Multisim commands. It is intended primarily as a reference.

### 2.6.1 File Menu

Contains commands for managing circuit files created with Multisim.

#### 2.6.1.1 File/New (Ctrl+N)

Opens an untitled circuit window that can be used to create a circuit. The new window opens using your default circuit preferences. Until you save, the circuit window is named “Circuit#”, where “#” is a consecutive number. For example, you could have “Circuit1”, “Circuit2”, “Circuit3”, and so on.

You can create an unlimited number of circuits in one session.



**Note** Users of editions other than Professional or Power Professional can only have one circuit open at a time. For these users, the **File/New** command closes the currently open circuit file.

#### 2.6.1.2 File/Open (Ctrl+O)

Opens a previously created circuit file or netlist. Displays a file browser. If necessary, change to the location of the file you want to open.

**Note** You can open files created with Version 5 of Electronics Workbench, files created in Multisim and netlist files.

#### 2.6.1.3 File/Close

Closes the active circuit file. If any changes were made since the last save of the file, you are prompted to save those changes before closing.

### 2.6.1.4 File/Save (Ctrl+S)

Saves the active circuit file. If this is the first time the file is being saved, displays a file browser. If you want, change to the desired location for saving the file. You can save a circuit file with a name of any length.

The extension `.msm` is added to the file name automatically. For example, a circuit named `Mycircuit` will be saved as `Mycircuit.msm`.

**Tip** To preserve the original circuit without changes, choose **File/Save As**.

### 2.6.1.5 File/Save As

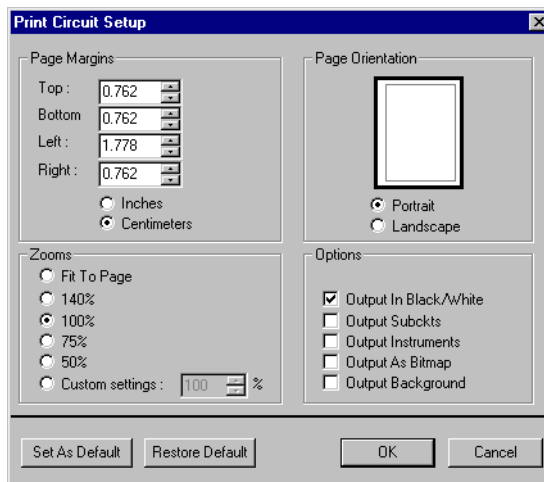
Saves the current circuit with a new file name. The original circuit remains unchanged.

**Tip** Use this command to experiment safely on a copy of a circuit, without changing the original.

### 2.6.1.6 File/Print Circuit

Prints all or some aspects of a circuit on a printer attached to your system. You can choose one of the following options:

- Print—see “3.13 Printing the Circuit” on page 3-31
- Print Preview—previews the circuit as it will be printed. Opens a separate window, where you can move from page to page and zoom in for details. You can also print what you preview. For details, see “3.13 Printing the Circuit” on page 3-31.
- Print Circuit Setup—allows you to set the page margins, page orientation (landscape or portrait), zoom level and output options (for example to include/exclude subcircuits).



These settings apply only to the current circuit. For details on these fields see “3.13 Printing the Circuit” on page 3-31.

**Note** If you change the page orientation using the Page Circuit Setup screen, the change is also reflected in the Print Setup screen.

### 2.6.1.7 File/Print Reports

You can choose from one of the following reports to print:

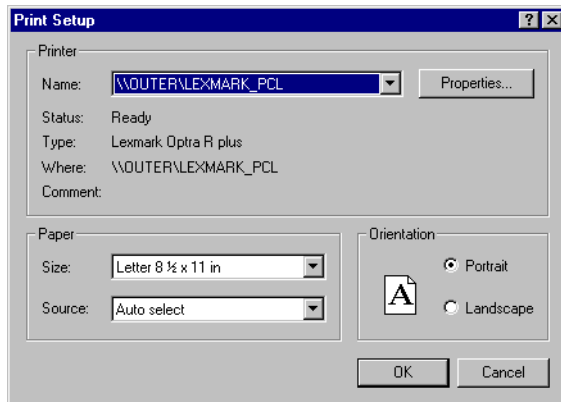
- Database Family List — see “11.2 (virtual)Database Family List” on page 11-1
- Component Detail Report — see “11.3 Component Detail Report” on page 11-3

### 2.6.1.8 File/Print Instruments

Prints the faces of any or all of the instruments in your circuit. For details, see “11.4 Instruments” on page 11-4.

### 2.6.1.9 File/Print Setup

Changes the printer, paper size, paper source and page orientation for a selected printer.



**Note** If you change the page orientation using the Print Setup screen, the change is also reflected in the Print Circuit Setup screen.

### 2.6.1.10 File/Recent Files

Displays a list of the eight most recently saved circuit files. To re-open a file, select it from the list.

### 2.6.1.11 File/Exit

Closes all open circuit windows and exits Multisim. If you have unsaved changes in any circuits, you are prompted to save or cancel them.

## 2.6.2 Edit Menu

Contains commands for removing, duplicating or selecting information. If a command is not applicable to the selected item (for example, a component), it is dimmed.

### 2.6.2.1 Edit/Undo (Ctrl+Z)

Undoes the most recently performed delete.

### 2.6.2.2 Edit/Cut (Ctrl+X)

Removes selected components, circuits or text. What you cut is placed on the Windows clipboard so you can paste it elsewhere.

### 2.6.2.3 Edit/Copy (Ctrl+C)

Copies selected components, circuits or text. The copy is placed on the Windows clipboard. You can then use the **Paste** command to paste the copy elsewhere, including other applications such as word processors.

### 2.6.2.4 Edit/Paste (Ctrl+V)

Places the contents of the clipboard. The cursor shows a “ghosted” image of the item to be pasted. Click to indicate where the item is to be pasted.

### 2.6.2.5 Edit/Delete (Del)

Permanently removes selected components or text. Does not place the selection on the clipboard and does not affect anything currently on the clipboard.

---

#### **Caution**

Use the Delete command with care. Deleted information cannot be retrieved, although the most recent deletion can be recovered using the Undo command.

---

**Note** Deleting a component or instrument removes it from the circuit window, not from its toolbar.

### 2.6.2.6 Edit/Select All (Ctrl+A)

Selects all items in the active circuit window.

**Tip** To select all but a few items, use the **Select All** command and then deselect the ones you don't want by CTRL-clicking.

### 2.6.2.7 Edit/Flip Horizontal (Alt+X)

Flips the selection horizontally. For details, see “3.8 Rotating/Flipping Components” on page 3-16.

### 2.6.2.8 Edit/Flip Vertical (Alt+Y)

Flips the selection vertically. For details, see “3.8 Rotating/Flipping Components” on page 3-16.

### 2.6.2.9 Edit/90 Clockwise (Ctrl+R)

Rotates the selection 90 degrees clockwise. For details, see “3.8 Rotating/Flipping Components” on page 3-16.

### 2.6.2.10 Edit/90 CounterCW (Shift+Ctrl+R)

Rotates the selection 90 degrees counter-clockwise. For details, see “3.8 Rotating/Flipping Components” on page 3-16.

### 2.6.2.11 Edit/Component Properties

Opens the properties screen of the item selected in the active window. For details, see “3.9 Placed Component Properties” on page 3-17.

## 2.6.3 View Menu

Contains commands for displaying and hiding elements of the circuit window such as the grid, toolbars and page boundaries.

### 2.6.3.1 View/Toolbars

Shows or hides the selected toolbar. Choose to show or hide any or all of the following toolbars:

- system toolbar
- design toolbar
- instruments toolbar
- zoom toolbar (the “zoom” buttons of the system toolbar)
- “In Use” list.

For information on these elements, see “2.2 Introduction to the Multisim Interface” on page 2-2.

### 2.6.3.2 View/Component Bars

Shows or hides the appropriate component toolbar for the selected database. Choose to show or hide the component toolbar for any or all of the following databases:

- Multisim database
- User database
- Edaparts Bar

For information on these databases see “4.2 Structure of the Component Database” on page 4-1.

### 2.6.3.3 View/Status Bar

Shows or hides the status bar, which provides useful information about the current operation and a description of the item the cursor is currently pointing to.

### 2.6.3.4 View/Show Simulation Error Log/Audit Trail

Shows or hides the simulation log, which records all the events of a circuit simulation. For details about this log, see “7.3.2 Start/Stop/Pause Simulation” on page 7-4.

### 2.6.3.5 View/Show XSpice Command Line Interface

Opens a window into which you can type Xspice commands to be executed. Press RETURN to execute the command. The command is listed at the top part of the window and any errors are reported in the error log/audit trail.

### 2.6.3.6 View/Show Grapher (Ctrl+G)

Shows or hides the Grapher screen, which shows the results of simulation on a graph or chart. For details about this screen, see “8.23 Viewing the Analysis Results: Grapher” on page 8-57.

### 2.6.3.7 View/Show Simulate Switch

Shows or hides the simulation on/off switch. An alternative to using the Design Bar button or menu commands. The switch looks like this:



### 2.6.3.8 View/Show Text Description Box (Ctrl+D)

Opens a window in which you can enter comments or information about the circuit. For details, see “3.11.4 Adding Miscellaneous Text” on page 3-27.

### 2.6.3.9 View/Show Grid

Shows or hides a grid in the background of the circuit window. This helps you place elements in specific locations on a grid. For details, see “3.3.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-3.

### 2.6.3.10 View/Show Page Bounds

Shows or hides page boundaries in the circuit window. This helps you note where circuits will appear on printed output. For details, see “3.3.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-3.

### 2.6.3.11 View/Show Title Block and Border

Shows or hides the circuit’s title block and border. For details, see “3.3.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-3.

### 2.6.3.12 View/Zoom In (F8)

Lets you increase the magnification of your view of the circuit. For information on setting the default zoom level see “2.4.4 The Workspace Tab” on page 2-7.

### 2.6.3.13 View/Zoom Out (F9)

Lets you decrease the magnification of your view of the circuit. For information on setting the default zoom level see “2.4.4 The Workspace Tab” on page 2-7.



### 2.6.3.14 View/Find (Ctrl+F)

Displays a list of the reference IDs in the current circuit. You can select one or more of these reference IDs, whose associated components are then selected in the circuit window. For details, see “3.10 Finding Components in Your Circuit” on page 3-24.

## 2.6.4 Place Menu

Contains commands for placing elements in the circuit window

### 2.6.4.1 Place/Place Component (Ctrl+W)

Lets you browse the databases (“Multisim Master”, “Corporate Library” and “User”) for components to be placed. For details, see “3.5.1 Choosing a Component and Using the Browser Screen” on page 3-6.

### 2.6.4.2 Place/Place Junction (Ctrl+J)

Places a junction when you click. For details, see “3.7 Manually Adding a Junction (Connector)” on page 3-16.

### 2.6.4.3 Place/Place Bus (Ctrl+U)

Places a bus with segments created as you click. For details, see “3.14 Placing a Bus” on page 3-33.

### 2.6.4.4 Place/Place/Input/Output (Ctrl+I)

Adds connecting nodes to a circuit for use as a subcircuit. For details, see “3.12.2 Setting up a Circuit for Use as a Subcircuit” on page 3-29.

### 2.6.4.5 Place/Place Text (Ctrl+T)

Lets you place text on the circuit. For details, see “3.11.4 Adding Miscellaneous Text” on page 3-27.

### 2.6.4.6 Place/Replace Component

Before using, select a component on the circuit window to be replaced. Invokes the Browser screen (see Chapter 3, “Schematic Capture” for more information) from which you can select a new component. Click **OK** to replace the old component with the new one.

### 2.6.4.7 Place/Place as Subcircuit (Ctrl+B)

Places the contents of the clipboard as a subcircuit. For details, see “3.12.3 Adding Subcircuits to a Circuit” on page 3-30.

### 2.6.4.8 Place/Replace by Subcircuit (Ctrl+Shift+B)

Replaces the selected components with a subcircuit of the same components. For details, see “3.12.3 Adding Subcircuits to a Circuit” on page 3-30.

## 2.6.5 Simulate Menu

### 2.6.5.1 Simulate/Run (F5)

Runs/stops the simulation of the active circuit. Running a simulation starts a sequence of mathematical operations to compute values for the nodes (testpoints) in the circuit.

**Tip** A digital circuit can also be activated from the word generator. See Chapter 6, “Instruments” for more information.

### 2.6.5.2 Simulate/Pause (F6)

Pauses/resumes the current simulation.

### 2.6.5.3 Simulate/Default Instrument Settings

Allows you to set default settings for instruments that are based on a transient analysis (such as the oscilloscope, spectrum analyzer and logic analyzer). For details, see “6.4 Default Instrument Analysis Settings” on page 6-5.

### 2.6.5.4 Simulate/Digital Simulation Settings

Allows you to choose between optimizing on accuracy or speed of simulation when digital components are included in your circuit. For details, see “7.3.1 Simulating Circuits with Digital Components” on page 7-3.

### 2.6.5.5 Simulate/Instruments

Contains commands you use to place instruments (an alternative to using the instruments toolbar or the Design Bar). For details on these instruments, see Chapter 6, “Instruments”.

**Note** Ammeter and voltmeter instruments are available from the Indicators Parts Bin.

### **Simulate/Instruments/Multimeter**

Places a multimeter on the circuit window. A multimeter is used to measure AC or DC voltage or current, resistance, or decibel loss between two nodes in a circuit. For details, see “6.7 Multimeter” on page 6-8.

### **Simulate/Instruments/Function Generator**

Places a function generator on the circuit window. A function generator is a voltage source that supplies sine, triangular or square waves. It provides a convenient and realistic way to supply power to a circuit. For details, see “6.8 Function Generator” on page 6-12.

### **Simulate/Instruments/Wattmeter**

Places a wattmeter on the circuit window. A wattmeter provides the combined functions of a voltmeter and an ammeter. It is used to measure the magnitude of the active power, that is, the product of the voltage difference and the current flowing through the current terminals in a circuit. For details, see “6.12 Word Generator” on page 6-24.

### **Simulate/Instruments/Oscilloscope**

Places an oscilloscope on the circuit window. The dual-channel oscilloscope displays the magnitude and frequency variations of electronic signals. It can provide a graph of the strength of one or two signals over time, or allow comparison of one waveform to another. For details, see “6.10 Oscilloscope” on page 6-15.

### **Simulate/Instruments/Bode Plotter**

Places a Bode plotter on the circuit window. A Bode plotter produces a graph of a circuit’s frequency response and is useful for analyzing filter circuits. For details, see “6.11 Bode Plotter” on page 6-21.

### **Simulate/Instruments/Word Generator**

Places a word generator on the circuit window. A word generator sends digital words or patterns of bits into circuits to test them. For details, see “6.12 Word Generator” on page 6-24.

### **Simulate/Instruments/Logic Analyzer**

Places a logic analyzer on the circuit window. A logic analyzer displays the levels of up to 16 digital signals in a circuit. It is used for fast data acquisition of logic states and advanced timing analysis to help design large systems and carry out troubleshooting. For details, see “6.13 Logic Analyzer” on page 6-29.

### Simulate/Instruments/Logic Converter

Places a logic converter on the circuit window. A logic converter is able to perform several transformations of a circuit representation. It has no real world counterpart. For details, see “6.14 Logic Converter” on page 6-33.

### Simulate/Instruments/Distortion Analyzer

Places a distortion analyzer on the circuit window. A typical distortion analyzer provides distortion measurements for audio signals in the range of 20 Hz to 100 KHz. For details, see “6.15 Distortion Analyzer” on page 6-35.

**Note** This instrument is hidden when the simplified version option is selected. For details, “13.4.1.2 Simplified Version” on page 13-7.

### Simulate/Instruments/Spectrum Analyzer

Places a spectrum analyzer on the circuit window. The spectrum analyzer is used to measure frequency versus amplitude. For details, see “6.16 Spectrum Analyzer” on page 6-37.

**Note** This instrument is hidden when the simplified version option is selected. For details, “13.4.1.2 Simplified Version” on page 13-7.

### Simulate/Instruments/Network Analyzer

Places a network analyzer on the circuit window. The network analyzer is used to measure the scattering parameters (or S-parameters) of a circuit, commonly used to characterize a circuit intended to operate at higher frequencies. For details, see “6.17 Network Analyzer” on page 6-37.

**Note** This instrument is hidden when the simplified version option is selected. For details, “13.4.1.2 Simplified Version” on page 13-7.

## 2.6.5.6 Simulate/Analyses

Contains commands you use to set up and run the circuit’s analysis (an alternative to using the Design Bar). For details on these analyses, see Chapter 8, “Analyses”.

### Simulate/Analyses/DC Operating Point

Sets up and runs DC operating point analysis, which determines the DC operating point of a circuit. For details, see “8.4 DC Operating Point Analysis” on page 8-10.

### Simulate/Analyses/AC Analysis

Sets up and runs AC analysis, in which the DC operating point is first calculated to obtain linear, small-signal models for all nonlinear components. Then a complex matrix (containing both real and imaginary components) is created. For details, see “8.5 AC Analysis” on page 8-12.

## **Simulate/Analyses/Transient Analysis**

Sets up and runs Transient analysis, also called time-domain transient analysis, which computes the circuit's response as a function of time. For details, see "8.6 Transient Analysis" on page 8-14.

## **Simulate/Analyses/Fourier Analysis**

Sets up and runs Fourier analysis, which evaluates the DC, fundamental and harmonic components of a time-domain signal. For details, see "8.7 Fourier Analysis" on page 8-17.

## **Simulate/Analyses/Noise Analysis**

Sets up and runs Noise analysis, which is used to detect the magnitude of noise power in the output of electronic circuits. For details, see "8.8 Noise Analysis" on page 8-21.

## **Simulate/Analyses/Distortion Analysis**

Sets up and runs Distortion analysis, which measures harmonic distortion and intermodulation distortion products. For details, see "8.9 Distortion Analysis" on page 8-25.

## **Simulate/Analyses/DC Sweep**

Sets up and runs DC sweep analysis, which computes the DC operating point of a node in the circuit for various values of one or two DC sources in the circuit. For details, see "8.10 DC Sweep Analysis" on page 8-27.

**Note** This analysis is hidden when the simplified version option is selected. For details, see "13.4.1.2 Simplified Version" on page 13-7.

## **Simulate/Analyses/Sensitivity**

Sets up and runs Sensitivity analysis, which calculates the sensitivity of an output node voltage or current with respect to the parameters of all components (DC sensitivity) or one component (AC sensitivity) in a circuit. For details, see "8.11 DC and AC Sensitivity Analyses" on page 8-29.

**Note** This analysis is hidden when the simplified version option is selected. For details, see "13.4.1.2 Simplified Version" on page 13-7.

## **Simulate/Analyses/Parameter Sweep**

Sets up and runs Parameter Sweep analysis, which verifies the operation of a circuit by simulating it across a range of values for a component parameter. For details, see "8.12 Parameter Sweep Analysis" on page 8-32.

**Note** This analysis is hidden when the simplified version option is selected. For details, see "13.4.1.2 Simplified Version" on page 13-7.

## Simulate/Analyses/Temperature Sweep

Sets up and runs Temperature Sweep analysis, which quickly verifies the operation of a circuit by simulating it at different temperatures. The effect is the same as simulating the circuit several times, once for each different temperature. You control the temperature values. For details, see “8.13 Temperature Sweep Analysis” on page 8-35.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Pole Zero

Sets up and runs Pole Zero analysis, which finds the poles and zeros in the small-signal AC transfer function of a circuit. For details, see “8.16 Pole Zero Analysis” on page 8-44.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Transfer Function

Sets up and runs Transfer Function analysis, which calculates the DC small-signal transfer function between an input source and two output nodes (for voltage) or an output variable (for current) in a circuit. It also calculates input and output resistances. For details, see “8.14 Transfer Function Analysis” on page 8-39.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Worst Case

Sets up and runs Worst Case analysis, a statistical analysis that lets you explore the worst possible effects on circuit performance of variations in component parameters. For details, see “8.15 Worst Case Analysis” on page 8-41.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Monte Carlo

Sets up and runs Monte Carlo analysis, a statistical analysis that lets you explore how changing component properties affects circuit performance. For details, see “8.17 Monte Carlo Analysis” on page 8-50.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Batched Analyses

Sets up and runs batched analyses, which let you set up a series of different analyses, or different variations on the same analysis, to be performed on a circuit in sequence. For details, see “8.19 Batched Analyses” on page 8-54.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/User-Defined Analysis

Sets up and runs a user-defined analysis. This command presents you with a screen into which you can type the SPICE commands to be executed to perform the analysis. For details, see “8.20 User-Defined Analyses” on page 8-55.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Noise Figure Analysis

This analysis is part of Multisim’s RF Design module (standard in the Power Professional edition, optional in the Professional edition) and is described in Chapter 14, “RF”.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/Stop Analysis

Stops the currently running analysis.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## Simulate/Analyses/RF Analyses

Included in the menu for the purposes of completeness. Since RF analyses are available from the Network Analyzer instrument, this command places the Network Analyzer for you. For details, see “14.4.2 Network Analyzer” on page 14-16.

**Note** This analysis is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.5.7 Simulate/Postprocess

Opens the Postprocessor screen, which you use to combine the results of several analyses in different ways. To use the Postprocessor, you must have performed at least one analysis on your circuit. For details, see Chapter 9, “Postprocessor”.

**Note** This function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.5.8 Simulate/VHDL Simulation



Runs the VHDL simulation module. For details, see Chapter 10, “HDLs and Programmable Logic”.

**Note** This simulation is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.5.9 Simulate/Verilog HDL Simulation



Runs the Verilog HDL simulation module. For details, see Chapter 10, “HDLs and Programmable Logic”.

**Note** This simulation is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.5.10 Simulate/Auto Fault Option

Applies faults to randomly selected components in the circuit. You choose the number of faults (either in total, or the number of each type of fault) to be applied. For details, see “7.7.1 Using the Auto Fault Option” on page 7-14.

### 2.6.5.11 Simulate/Global Component Tolerances

Multisim’s virtual components are, by default, “ideal”—they have no internal resistance and their output is consistent. You can choose to use global components instead. These randomly introduce variances to simulate the performance of actual, physical components. Global component settings affect the simulation results. See “4.6 Component Nominal Values and Tolerances” on page 4-32 for details.

**Note** This function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.



## 2.6.6 Transfer Menu

### 2.6.6.1 Transfer/Transfer to Ultiboard

Displays a file browser where you choose or enter a file name for the transferred data. A file of the correct format is created. If you plan to use backannotation, you *must* save your file immediately.

### 2.6.6.2 Transfer/Transfer to other PCB Layout

Displays a file browser where you choose or enter a file name for the transferred data. You can also choose the appropriate file type from a list of available types.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.6.3 Transfer/Backannotate from Ultiboard

Backannotates changes made to a circuit in Ultiboard (for example, deleted components) to the Multisim circuit file. Displays a file browser where you choose the backannotation file corresponding to your circuit file. The circuit file must be open before you use this command.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.6.4 Transfer/VHDL Synthesis



Runs the VHDL Synthesis program on a file created from the current circuit. You are prompted to save the file, and then VHDL Synthesis appears with the file loaded in it. For details, see Chapter 10, “HDLs and Programmable Logic”.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.6.5 Transfer/Export Simulation Results to MathCAD

Exports the results of your simulation to a file format readable by MathCAD™. For details, see Chapter 12, “Transfer/Communication”.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.6.6 Transfer/Export Simulation Results to Excel

Exports the results of your simulation to a file format readable by Excel™. For details, see Chapter 12, “Transfer/Communication”.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.6.7 Transfer/Export Netlist

Exports the netlist of your design. Opens a standard file browser where you can choose the file name and folder for the exported netlist.

**Note** This transfer function is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

## 2.6.7 Tools Menu

**Note** The Tools menu is hidden when the simplified version option is selected. For details, see “13.4.1.2 Simplified Version” on page 13-7.

### 2.6.7.1 Tools/Create Component

Invokes the Create Component Wizard, which takes you through the steps of creating a component. For details, see “5.7 Adding Components” on page 5-9.

### 2.6.7.2 Tools/Edit Component

Lets you edit any component in the database. For details, see “5.4 Editing Components” on page 5-5.

### 2.6.7.3 Tools/Copy Component

Lets you copy any component in the database. For details, see “5.9 Copying Components” on page 5-15.

### 2.6.7.4 Tools/Delete Component

Lets you delete any component in the Corporate Library or User databases. For details, see “5.8 Removing Components” on page 5-14.

### **2.6.7.5 Tools/Database Management**

Lets you add component families, assign button icons to families, and (for some editions of Multisim) enter or modify user field titles for those component families. For details, see “5.3 Managing the Database” on page 5-3.

### **2.6.7.6 Tools/Update Models**

Updates any newly loaded Multisim Master models that are being used from the Multisim Master database in the active circuit. Use this option only after you have loaded a new Multisim Master database.

### **2.6.7.7 Tools/Remote Control/Design Sharing**

Allows you to communicate with and share designs with others, either across a network or using the Internet. For Project/Team Design module or Educational edition users only. For details, see Chapter 13, “Educators Guide”.

### **2.6.7.8 Tools/Edaparts.com**

Launches your Internet browser loaded with the Electronic Workbench Edaparts Web site. You can then download components into the “User” database. For details, see “4.4 Using Edaparts.com” on page 4-24.

## **2.6.8 Options Menu**

### **2.6.8.1 Options/Preferences**

Lets you specify preferences for current or subsequent circuits. For details see “2.4 Customizing the Interface” on page 2-4.

### **2.6.8.2 Options/Modify Title Block**

Lets you enter data to appear in the circuit’s title block. For details, see “3.11.3 Adding a Title Block” on page 3-27.

### **2.6.8.3 Options/Simplified Version**

Changes the interface display by hiding the design bar and hiding some the more complex instruments and analyses. The simplified version option can be enabled or disabled using Global Restrictions — see “13.4.1.2 Simplified Version” on page 13-7. If the simplified version is disabled, it is greyed out in the Options menu.

### **2.6.8.4 Options/Circuit Restrictions**

Lets you set restrictions on a particular circuit’s functionality for other users. You control these restrictions through the use of passwords. For details, see “13.4.2 Setting Circuit Restrictions” on page 13-9.

## **2.6.9 Window Menu**

Contains commands used to control the display of Multisim windows. Lists all open circuit windows.

### **2.6.9.1 Window/Cascade**

Arranges circuit windows so that they overlap.

### **2.6.9.2 Window/Tile**

Resizes all open circuit windows so they all show on the screen. Allows you to quickly scan all open circuit files.

### **2.6.9.3 Window/Arrange Icons**

Lines up minimized windows.

### **2.6.9.4 Window (open files)**

Lists the open Multisim circuit files. Select one to make it active.

## 2.6.10 Help Menu

Contains commands that display on-line help and Multisim edition information.

**Tip** If you want to be able to refer to Help information as you work on a circuit, use the Keep Help on Top function in the Help window's Options menu.

### 2.6.10.1 Help/Multisim Help

Displays Multisim's basic help file. The contents of this help file are identical to the contents of the chapters of this guide.

### 2.6.10.2 Help/Multisim Reference

Displays Multisim's reference help file. The contents of this help file are identical to the contents of the appendices of this guide.

### 2.6.10.3 Help/Release Notes

Displays edition information, additions or corrections to printed documentation, and other useful information.

### 2.6.10.4 Help/About Multisim

Displays information on the edition number of Multisim. Customer support may ask for this number.



# Chapter 3

## Schematic Capture

### 3.1 About this Chapter

This chapter describes all the basic functions involved in creating a circuit in the circuit window. This chapter explains the fundamental steps in circuit creation, but is not intended to describe all the potential aspects of circuit design. For example, you should look to other chapters for details on the component database, instructions on editing components, and information on adding instruments.

### 3.2 Introduction to Schematic Capture

Schematic capture is the first stage in developing your circuit. In this stage you choose the components you want to use, place them on the circuit window in the desired position and orientation, wire them together, and otherwise prepare your design. Multisim also allows you to modify component properties, orient your circuit on a grid, add text and a title block, add sub-circuits and buses, and control the color of the circuit window background, components and wires.

### 3.3 Setting up Your Circuit Window

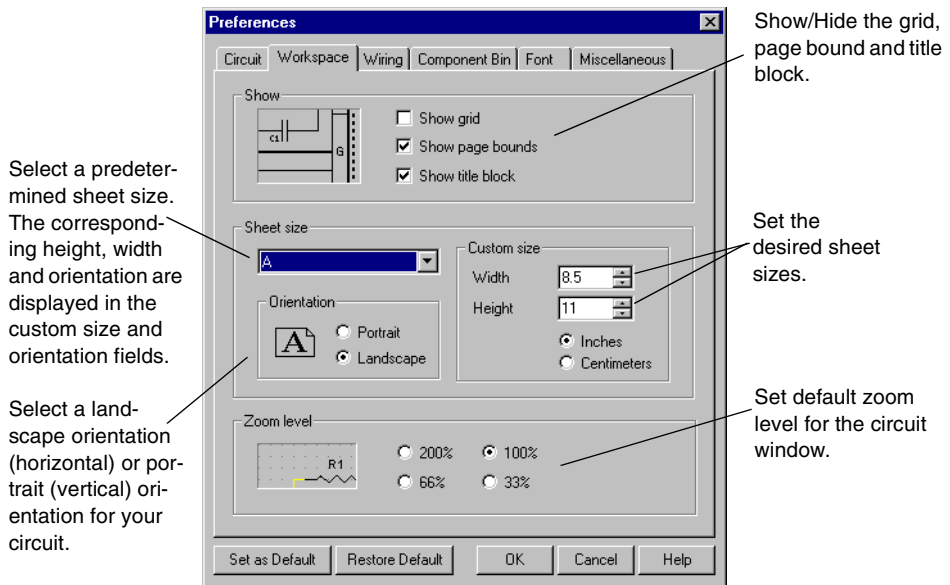
Multisim lets you set up the circuit window to match your design requirements. In particular, you can set:

- sheet size
- whether or not the grid, page bounds, page borders and title block are visible
- the symbol set (ANSI or DIN) you want to use
- color scheme for background, components, wiring and text
- fonts for names and values of components, pins and nodes.

When you create a component file, the current settings in the Preferences screen (as described in “2.3.1 About Customizing the Interface” on page 2-5) are used as the defaults for these options. You can override the defaults, with your new settings being stored with the circuit file. This allows you to have unique settings for each file you create. If you click **Set as default** when setting preferences, your settings affect all subsequent circuits as well as the current one.

### 3.3.1 Setting up a Sheet Size

- To set the sheet size for this circuit:
  1. Choose **Options/Preferences** and click the Workspace tab.



2. Choose the desired sheet size from the drop-down list.
3. Change any of the characteristics (orientation or size) necessary.



4. To confirm the sheet size, click **OK**. To cancel it, click **Cancel**.

or

To restore the default settings to the current circuit (and start again), click **Restore Default**. Then continue to make your changes.

### 3.3.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders

Multisim lets you display a background grid on the circuit window, to help you orient your components. It also lets you show or hide the component's title block (explained in more detail in "3.10.3 Adding a Title Block" on page 3-29), the page bounds that show the parameters of your sheet size, and the page borders that display a mapping grid.

As described in "2.3.4 The Workspace Tab" on page 2-8, the current and default settings that specify which of these items are displayed is set in the Preferences screen. You can also use the pop-up (right-click) menu to set these defaults for the current circuit only.

- To affect what is shown or hidden in the current circuit, do one of the following:
  - enable **View/Show Grid**, **View/Show Page Bounds** or **View/Show Title Block and Border**
  - or
  - right-click on the circuit window and choose **Show Grid**, **Show Page Bounds**, or **Show Title Block and Border** from the pop-up menu that appears.

### 3.3.3 Selecting a Symbol Set

Multisim allows you to use either ANSI or DIN symbols on your circuit window.

- To choose the desired symbol set, choose **Options/Preferences**, display the Component Bin tab, and select the desired symbol set. Select the symbol set to be used for components. The graphic changes to represent the selected symbol set. To override this setting for individual components, see "5.9.2 Creating and Editing a Component's Symbol with the Symbol Editor" on page 5-19.

### 3.3.4 Selecting a Circuit Color Scheme

New circuits are created using the default color scheme.

- To select a different color scheme or create your own, choose **Options/Preferences**, display the Circuit tab, and select a built in color scheme, or create a custom one. The selected colors appear in the preview box. For more details about these settings see “2.3.3 The Circuit Tab” on page 2-7.

### 3.3.5 Selecting Fonts for Labels, Values and Names

- To select a font for text elements on your circuits (e.g. component labels and text descriptions), choose **Options/Preferences**, display the Font tab, and select the desired font, font size and style. An example of the selected font appear in the sample window. For more details about these settings see “2.3.7 The Font Tab” on page 2-11.

## 3.4 Selecting Components from the Database



The first step in schematic capture is placing the appropriate components on your circuit window. Multisim components are stored in three databases: “Multisim Master”, “Corporate Library”, and “User”. (These databases are discussed more thoroughly in Chapter 4, “Components”.)

There are two methods of locating a component in a database:

- through either the component toolbars or **Place/Place Component**, you can browse through all the component groups/families, as explained in “3.4.1 Choosing a Component and Using the Browser Screen” on page 3-7
- by searching a specific component group/family of the database, as explained in Chapter 4, “Components”.

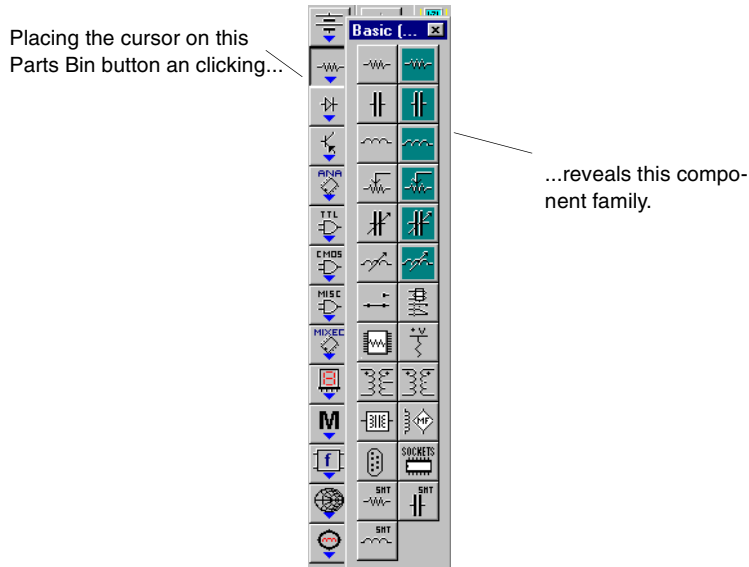
The first of these choices is used most commonly. The families of components you need to create a circuit are grouped into logical divisions, each grouping represented by a Parts Bin button on the Component toolbar. This logical grouping is a key advantage of Multisim, saving you time and frustration. You can toggle the Component toolbar on and off by clicking the **Components** button on the Design Bar.



Each Parts Bin button on the Component toolbar corresponds to a group of components with similar functionality. Placing your cursor over one of these buttons and clicking, opens the corresponding Parts Bin containing a button for each component family. The behaviour of the Component toolbar and its Parts Bins is affected by your circuit’s preferences. For details, see “2.3.6 The Component Bin Tab” on page 2-10.

**Note** For a detailed look at the make-up of the component Parts Bins, see “4.1.3 Classification of Components in the Database” on page 4-3.

For example:



Electronics Workbench provides the unique concept of virtual components in Multisim. Virtual components are not “real”, that is, they cannot be purchased, and have no footprint. They are included for simulation flexibility. Virtual component family buttons are color coded green on the Parts Bin. Virtual components appear by default in a different color from other components on the circuit window. You control the color as described in “2.3.3 The Circuit Tab” on page 2-7.

## 3.5 Placing Components

### 3.5.1 Choosing a Component and Using the Browser Screen



By default, the Component Design Bar button is enabled and one or more Component toolbars are visible. If no toolbar is visible, click the Component button on the Design Bar or select **View/Component Bars** and choose the desired database.

➤ To choose and place a component.

1. On the desired Component toolbar, place the cursor on the desired Parts Bin button and click. The associated Parts Bin appears.
2. From the Parts Bin, click the button for the desired component family. If the selected component is a virtual component, you can simply place the component. For other components, a simplified Browser screen appears by default.
3. To view the full Browser screen click **Advanced**. Alternatively, you can display the full Browser screen by choosing **Place/Place Component**.
4. From the Browser screen, select the desired component from the **Component List**. Information about that component appears.
5. If you selected the wrong component family from the toolbar, choose the correct component family from the Component family field of the Browser screen. The information in the Select Component area will change accordingly.

**Tip** To make your scroll through the Browser's **Components List** faster, simply type the first few characters of the component's name. Click **Advanced** to see the full version of the Browser screen.

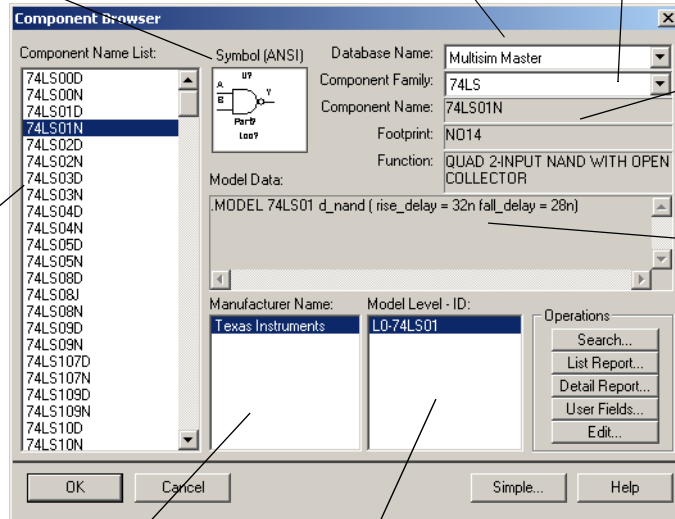
**Note** Virtual components are identified by a green colored button in the Parts Bins. These components cannot be purchased.

Shows the symbol that will be used to represent the component selected in the Component List.

Indicates the database from which to draw the component (defaults to the database of the selected Parts Bin).

Indicates the component family with which you are working (defaults to the selected family toolbar)

Lists the components available in the family named in Component Family field.



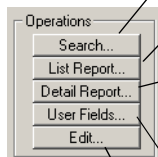
Information about the component selected in the Component List

Contains the model's information.

Lists the manufacturers of the component selected in the Component List.

Lists the model levels available for the component selected in the Component List.

Click to search the database. For details, see "4.2 Locating Components in the Database" on page 4-22.



Produces a list of the components in the family. For details, see "11.1 (virtual)(virtual)Database Family List" on

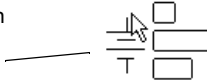
Produces detailed information about the selected component. For details, see "11.2 Component Detail Report" on page 11-3.

Allows you to enter user field information for the selected component. For details, see "13.7 Working with User Fields" on page 13-7.

Allows you to edit the component. For details, see Chapter 5, "Component Editing".

- To confirm that this is the component you want to place, click **OK**. (To cancel placing the component, click **Cancel**.) The Browser screen disappears and the cursor on the circuit window changes to a ghost image of the component you wish to place. This indicates a component is ready to be placed.

A ghost image of the component appears on the circuit window showing you exactly where the component will be placed.



- Move your cursor to the location where you want the component placed. The workspace automatically scrolls if you move your cursor to the edges of the workspace.
- Click on the circuit window where you want the component placed. The component's symbol and labels appear (unless you have specified that they are not to be displayed, as explained in "3.10.1 Modifying Component Labels and Attributes" on page 3-27), as well as a unique reference ID made up of a letter and number. The letter represents the type of component and the number is a sequential number that indicates the order in which the components were originally placed. For example, the first digital component has the reference ID "U1", the next is "U2", the first inductor has the reference ID "L1", and so on.

**Note** If the component you place is a virtual component (that is, it has no equivalent in the real world, and will therefore not be exported to Ultiboard), it is a different color from real components and the Browser is not required (i.e. no Step 3). This color is set in the Preferences screen, as explained in "2.3.3 The Circuit Tab" on page 2-7.

**Note** If you are placing a component whose package includes multiple "devices" (for example, four separate gates), you are prompted to specify which of the sections you want to place. You can choose any one and, depending on your settings, you can use all of the available devices from one chip before starting to use a second. Your settings for placing components with multiple devices is set in the Preferences screen, as explained in "2.3.6 The Component Bin Tab" on page 2-10.

## 3.5.2 Using the "In Use" List

Each time you place a component or subcircuit, it is added to the "In Use" list for easy re-use. To place a copy of any currently placed component, simply select it from the list. The copied component appears at the top of your circuit window—you can move it to any location you like.

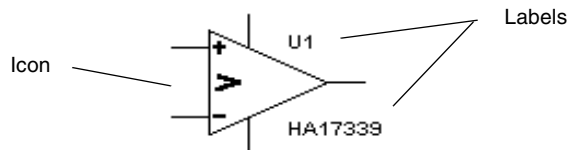
For educational purposes, you can hide the "In Use" list from students' view by using restrictions. See "13.3 Using Restrictions" on page 13-4 for details.

### 3.5.3 Moving a Placed Component

You can move a placed component to another location by doing one of the following:

- dragging the component
- selecting the component and pressing the arrow keys on your keyboard to move it up, down, or to either side in increments of one grid space.

**Note** A component's icon and labels can be moved independently or together—if you plan to move the component, be sure the whole component is selected, not just its label.



**Note** Moving automatically adjust the wiring to an appropriate configuration, unless you have disabled the “autowire on move” option described in “3.5.4 Setting Wiring Preferences” on page 3-15.

### 3.5.4 Copying a Placed Component

➤ To copy a placed component:

1. Select the desired component and choose **Edit/Copy**.

or

Right-click on the desired component, and, from the pop-up menu that appears, choose **Copy**.

2. From the **Edit** menu, choose **Paste**.

or

Right-click anywhere on the circuit window and, from the pop-up menu that appears, choose **Paste**.

3. The cursor shows a “ghosted” version of the copied component. Click at the location where you want the copied component placed.

Once you have placed the copied component, you can click and drag it to the desired location.

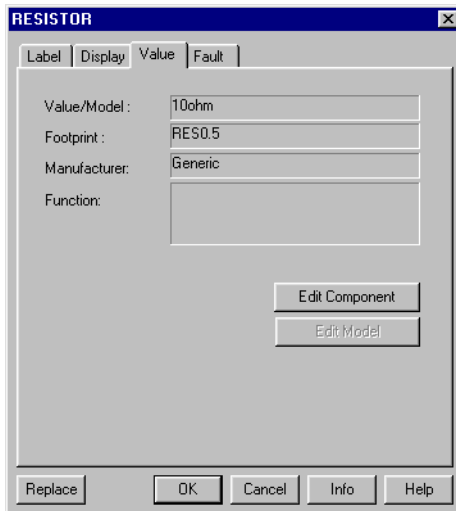
You can also copy a component using the Windows control keys for cut (CTRL - X), copy (CTRL - C) and paste (CTRL - V).

## 3.5.5 Replacing a Placed Component

You can easily replace a placed component by using the **Replace** button on the component's "properties" screen.

➤ To replace a placed component:

1. Double-click on the component you want to replace. The component's "properties" screen appears.



2. Click **Replace**. The Browser screen appears, already loaded with the selected component's component family.
3. Select a new component from the **Component Name List**, and click **OK**. The new component appears on the circuit window in the place of the previous one. For more details about the Component Browser, see "3.4.1 Choosing a Component and Using the Browser Screen" on page 3-7.



### 3.5.6 Controlling Component Color

The default color used for a component and the background color of the circuit window are controlled by the Preferences screen, as described in “2.3.1 About Customizing the Interface” on page 2-5.

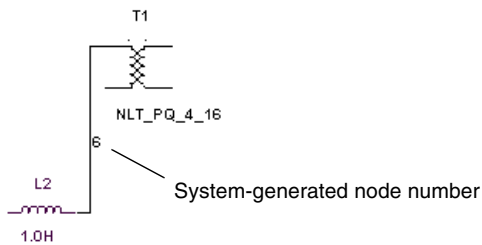
- To change the color of the placed component, right-click on the component and choose **Color** from the pop-up menu that appears. You are presented with a color palette. Choose a color and click **OK** to apply it to the selected item.
- To change the color of the background, and the default color scheme used throughout the circuit, right-click on the circuit window. The window that appears allows you to set a different color scheme. For more on color schemes, see “2.3.3 The Circuit Tab” on page 2-7.

## 3.6 Wiring Components

Once you have placed components on the circuit window, you will want to wire them together. All components have pins that you can use to wire them to other components or instruments. You can choose to wire components either automatically or manually. Automatic wiring, a feature unique to Multisim, means Multisim finds the path for wire placement for you and avoids wiring through other components or overlapping wires. Manual wiring means you control the flow of the wire on the circuit window. You can easily combine these methods for a single connection, for example, by starting with manual wiring and then switching to automatic.

### 3.6.1 Wiring Components Automatically

- To wire two components together, automatically:
  1. Click on a pin from the first component to start the connection (your pointer turns into a + sign) and drag. A wire appears, attached to your cursor.
  2. Click on a pin on the second component to finish the connection. Multisim automatically places the wire, which snaps to an appropriate configuration (unless you have disabled the “autowire on connection” option, as described in “3.5.4 Setting Wiring Preferences” on page 3-15). The wire is numbered as a node.



**Tip** If the connection was not successful, you may be trying to place the wire too close to other surrounding components. Try to make the connection at a slightly different location, or use manual wiring, as described in the following section.

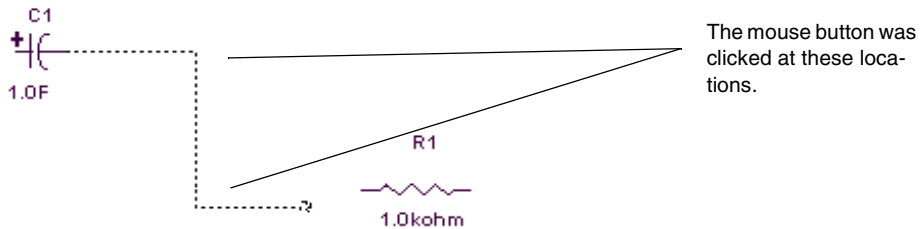
For information on changing the color of the wire, see “3.5.6 Controlling Wire Color” on page 3-16.

**Note** After a wire is connected between two pins the cursor returns to its normal mode and is ready for your next command.

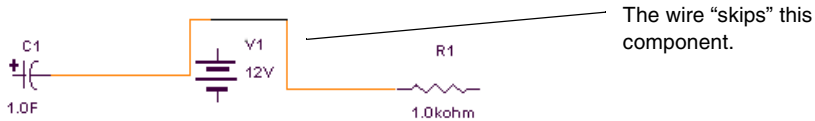
- To delete a wire, click on it and press DELETE or right-click on it and choose **Delete** from the pop-up menu that appears.

## 3.6.2 Wiring Components Manually

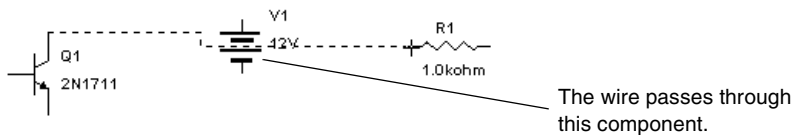
- To wire two components together, manually:
  1. Click on a pin from the first component to start the connection (your pointer turns into a + sign) and drag. A wire appears, attached to your cursor.
  2. Control the flow of the wire by clicking on points as you drag. Each click “fixes” the wire to that point. For example:



By default, Multisim “skips over” (avoids) components to which it is not connected. For example:



To pass through intermediary components instead, position the wire at the desired location beside the intermediary component and press SHIFT while dragging the wire. For example:



3. Click on the desired pin of the second component to finish the connection. The wire snaps to an appropriate configuration and the connection is numbered.
- Note** To stop the wiring process at any time, press ESC.
- To delete a wire, click on it and press DELETE or right-click on it and choose **Delete** from the pop-up menu that appears.

### 3.6.3 Combining Automatic and Manual Wiring

You can combine the two methods of wiring when placing a single wire. Multisim assumes you always want to perform automatic wiring until you click somewhere, which “locks” the wire to that point (this is manual wiring). Multisim then continues with automatic wiring, until you click once more—either at a destination pin or wire to complete the connection, or at another interim point on the wire you are placing. This method allows you to use automatic wiring for most connections, and use manual wiring only for difficult paths or portions of paths.

### 3.6.4 Setting Wiring Preferences

You can set your preferences for how Multisim controls the automatic wiring.

➤ To set your wiring preferences:

1. Choose **Options/Preferences** and select the Wiring tab.
2. In the autowire section of this tab, you can set two options: **Autowire on connection** and **Autowire on move**.

Enabling **Autowire on connection** ensures that Multisim will choose the most efficient path to wire the selected components together. Disabling this option gives you more control over the wire path, as Multisim follows the exact path of your cursor.

Enabling **Autowire on move** ensures that, when you move a wire a component, Multisim reshapes the wire to the most efficient path. Disabling this option means that the wires will move exactly in the shape and direction that you move a wired component.

3. In the wire width section of this tab, specify the width of all wires for the current circuit or subsequent circuits.
4. To save your options for the current circuit, click **OK**.

or

To save your options for the current and subsequent circuits, click **Set as Default**, then click **OK**.

or

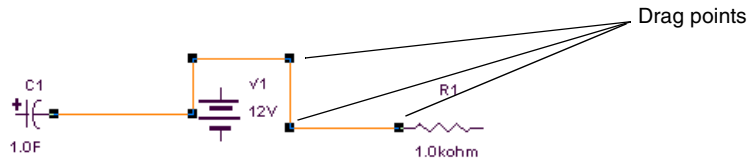
To restore your previously saved settings (if you don't like the changes you made and would like to start over again), click **Restore Default** and then continue making changes.

or

To cancel your changes and close the screen, click **Cancel**.

### 3.6.5 Modifying the Wire Path

- To alter the shape of the path once it is placed:
  1. Click on the wire. A number of drag points appear on the wire:



2. Click any of these and drag to modify the shape.

or

Move your cursor anywhere on the wire. When your cursor changes to a double arrow, click and drag, in the direction of the arrows, to modify the shape.

You can add or remove drag points to give you even more control over the wire shape.

- To add or remove drag points, press CTRL and click on the wire at the location where you want the drag point added or removed.

### 3.6.6 Controlling Wire Color

The default color used for wires is controlled by the Preferences screen, as described in “2.3.3 The Circuit Tab” on page 2-7.

- To change the color of the placed wire from its default values, right-click on the wire and choose **Color** from the pop-up menu that appears. You are presented with a color palette. Choose a color and click **OK** to apply it to the selected item.
- To change the color scheme (including the default wire color) for the current circuit only, right-click on the circuit window. The window that appears allows you to set a different color scheme. For more on color schemes, see “2.3.3 The Circuit Tab” on page 2-7.

## 3.7 Manually Adding a Junction (Connector)

If you want to start a wire at a position that is neither a pin nor a junction, you must add a junction. Multisim automatically inserts junctions when you connect one wire to another wire to differentiate them from wires that are crossing but not connected.

- To manually add a junction:
  1. Choose **Place/Place Junction**. Your cursor changes to indicate that a junction is ready to be placed.
  2. Click on the location where you want the junction placed. A connector appears at the selected location.
- To make a connection from a placed junction:
  1. Move your cursor close to the junction, until the cursor changes to a (+) symbol.
  2. Click and drag a wire from the junction to the desired location.

## 3.8 Rotating/Flipping Components

You can rotate or flip a component by either using the pop-up menu or selecting the component and using commands from the **Edit** menu. The instructions below describe the pop-up menu method only.

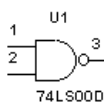
- To rotate a component:
  1. Right-click on the component.
  2. From the pop-up menu that appears, choose **90 Clockwise** to rotate the component 90 degrees clockwise.

or

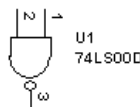
Choose **90 CounterCW** to rotate the component 90 degrees counter clockwise.

For example:

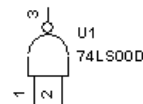
Unrotated:



Rotated 90 degrees clockwise:



Rotated 90 degrees counter-clockwise:



**Note** Text associated with the component, such as labels, values, and model information, will be repositioned and rotated as a result of your action. Pin numbers will rotate along with the associated pin. Any wires attached to the component are rerouted automatically.

➤ To flip a component:

1. Right-click on the component.
2. From the pop-up menu that appears, choose **Flip Horizontal** to flip the component horizontally.

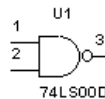
or

Choose **Flip Vertical** to flip the component vertically.

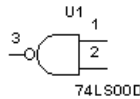
**Note** Text associated with the component, such as labels, values, and model information, may be repositioned, but is not flipped. Any wires attached to the component are rerouted automatically.

For example:

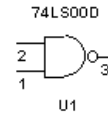
Unflipped:



Flipped horizontally



Flipped vertically



## 3.9 Placed Component Properties

Each component placed on the circuit window has a set of properties that control certain aspects of it beyond those stored in the Multisim database. These properties affect only the placed component, not other instances of that component in other circuits or other locations in this circuit. Depending on the type of component, these properties determine some or all of the following:

- the identifying information and labels about the placed component to be displayed on the circuit window (for details, see “3.10.1 Modifying Component Labels and Attributes” on page 3-27)
- the model of the placed component

- for some components, how the placed component will be used in analyses
- the faults to be used for the placed component's nodes.

The properties also show the component's value or model and footprint.

### 3.9.1 Displaying Identifying Information about a Placed Component

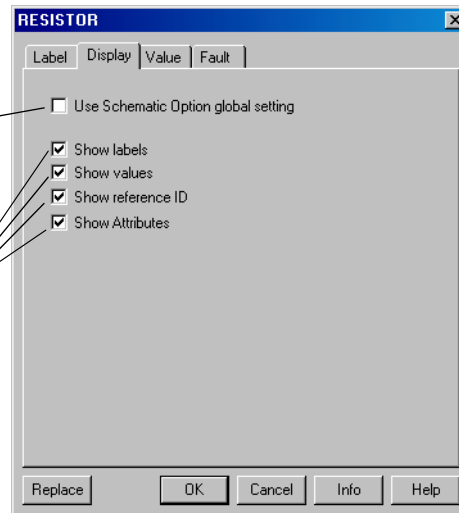
As described in “2.3.3 The Circuit Tab” on page 2-7, your settings in the Circuit tab of the Preferences screen determine which of the four pieces of identifying information (label, values, reference ID and attributes) is displayed on your circuit. You can also right-click on a the circuit window to display a window where you can set these defaults for the current circuit only. Finally, you can override these settings for an individual placed component, as described here.

For educational purposes, you can also hide this information from students' view for all the components in an entire circuit by using restrictions. See “13.3.2 Setting Circuit Restrictions” on page 13-8 for details.

- To set the identifying information to be displayed for a placed component:
1. Double-click on the component. A “properties” screen for the selected component appears.
  2. Click the Display tab.

When this option is enabled, the types of identifying information displayed for this individual component are controlled by the circuit's settings.

When the above global setting option is not enabled, these options are available and determine which identifying information is displayed for this individual component.



3. Disable the **Use Schematic Option global setting** option.



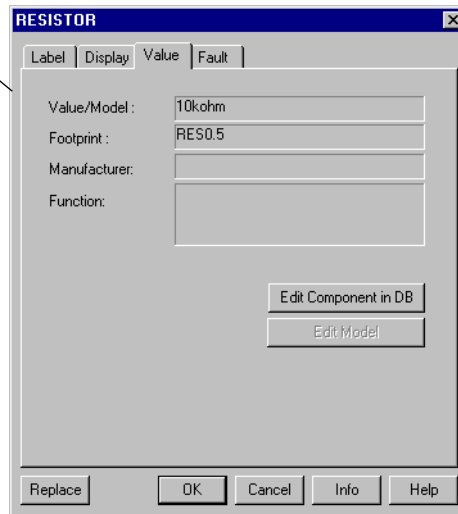
4. Enable the identifying information you want displayed for this component, and disable the identifying information you do not want displayed for this component.
5. To cancel your settings, click **Cancel**. To save your settings, click **OK**.

### 3.9.2 Viewing a Placed Component's Value/Model

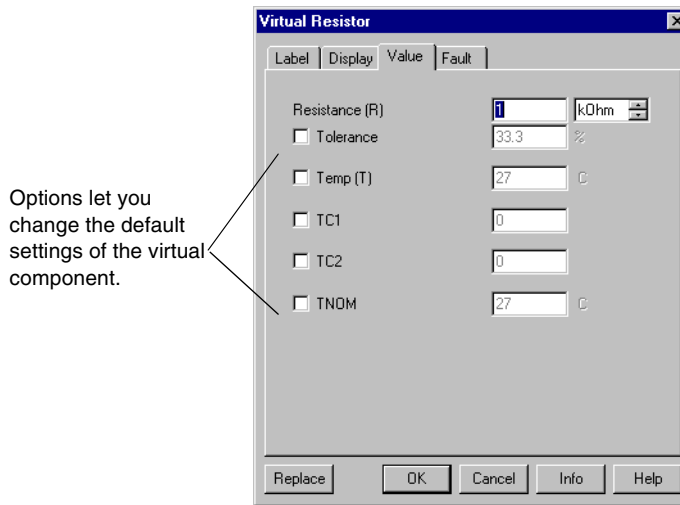
The Values tab of the “properties” screen for a component shows the value/model being used for the placed component. Depending on the type of component, you see one of two types of tabs.

For real components, the tab looks like this:

Identifies the value information of the component being used.



For virtual components, whose “value” can be set manually, the tab looks similar to this:



You can modify any of these fields (if a field is not editable, be sure you have enabled its corresponding option). To cancel your changes, click **Cancel**. To save your changes, click **OK**.

The ability to change the value or model of a component is allowed only for “virtual” components. It is important that you understand these components. Virtual components are not real; that is, you could not go to a supplier and purchase them. They are provided for your convenience. Multisim treats them slightly differently from real components in two ways. First, by default, virtual components are shown in a different color from that of real components on your schematic. This is to remind you that, since they are not real, these components will not be exported to PCB Layout software. Second, when you place such parts you do not need to choose from the Browser, since you can set the value or model of a virtual part to anything you want.

Virtual parts include all sources, virtual resistor/capacitor/inductor parts, and numerous others. Virtual parts also include other “ideal” devices intended to provide theoretical equivalents of, for example, the perfect opamp.

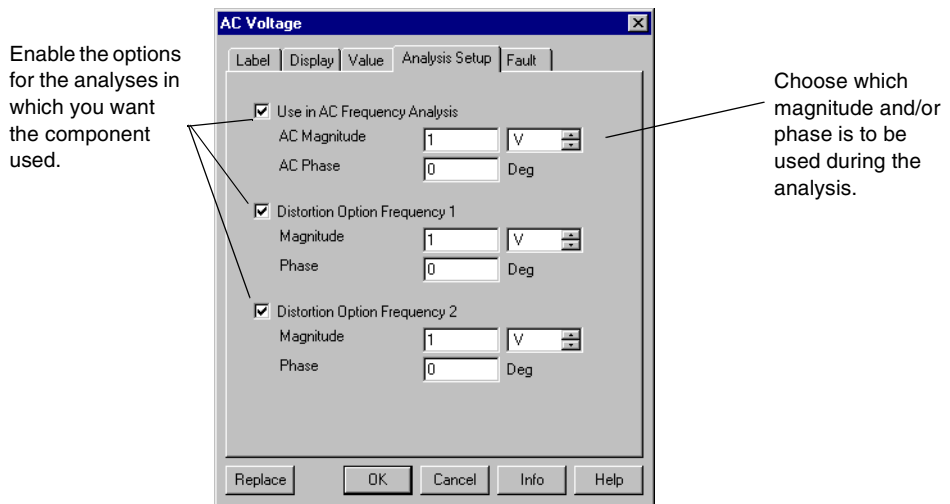
For educational purposes, you can hide the values of all components in a circuit from students’ view by using circuit restrictions. See “13.3.2 Setting Circuit Restrictions” on page 13-8 for details.

### 3.9.3 Controlling How a Placed Component is Used in Analyses

For some components, you can determine how they are to be used in any analyses you might perform on the circuit. These components offer an additional “properties” screen tab — Analysis Setup.

For educational purposes, you can hide analysis information for all components in a circuit. See “13.3.2 Setting Circuit Restrictions” on page 13-8 for details.

- To control how the component is used in analyses:
  1. Double-click on the component. The “properties” screen for the component appears.
  2. Click the Analysis Setup tab:

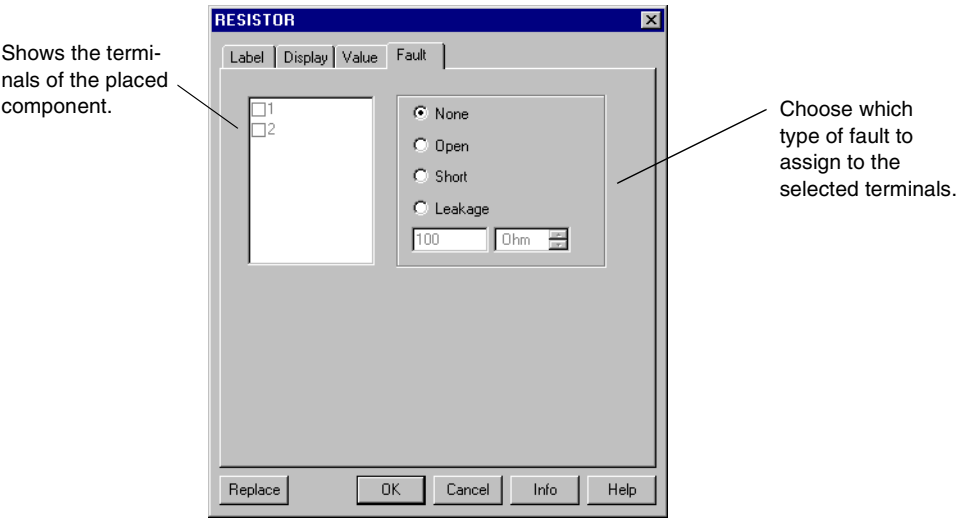


3. To cancel your changes, click **Cancel**. To save your changes, click **OK**.

### 3.9.4 Setting a Placed Component's Faults

You can assign a fault to any terminal of the placed component using the Faults tab of that component's "properties" screen.

- To assign a fault to a placed component:
  1. Double-click on the component. The component's "properties" screen appears.
  2. Click the Faults tab:



3. Select the terminals to which the fault should apply.
4. Enable the type of fault you want assigned to a terminal. The options are:

Option	Description
None	No fault
Open	Assigns a very high resistance to the terminals, as if the wire leading to the terminals was broken.
Short	Assigns a very low resistance to the terminal, so the component has no measurable affect on the circuit.
Leakage	Assigns the resistance value specified in the fields below the option, in parallel with the selected terminals. This causes the current to leak past the terminals instead of going through them.

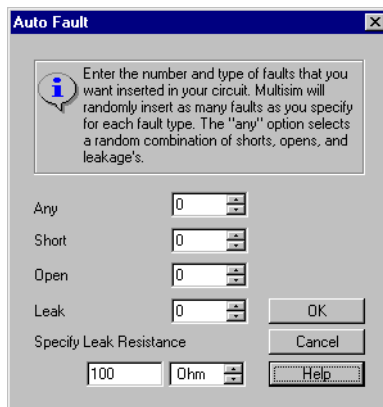
5. To cancel your changes, click **Cancel**. To save your changes, click **OK**.

### 3.9.5 Using the Auto Fault Option

When you use the Auto Fault option, you specify the number of faults or, optionally, the number of faults per different type of fault, that you want Multisim to assign to placed components in the circuit.

- To use the auto fault option:

1. Choose **Simulate/Auto Fault Option**. The Auto Fault screen appears:



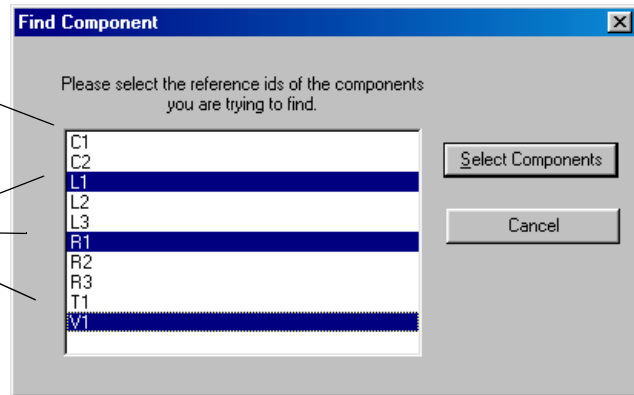
2. Use the up/down arrow keys or enter numerical values directly in the **Short**, **Open**, and **Leak** fields, or enter a numerical value in the **Any** field to let Multisim randomly select the type of faults to assign.
3. If you specify a number of leaks, enter a number and unit of measurement in the **Specify Leak Resistance** fields.
4. Click **OK** to apply the faults, or **Cancel** to cancel, and return to the circuit window.

## 3.10 Finding Components in Your Circuit

- To quickly find a component in the circuit window:
  1. Choose **View/Find**. A list of the components in your circuit appears.

This is the list of the reference IDs of all components in the circuit.

Components selected here will also be selected in the circuit window.



2. Select any number of components (hold down the SHIFT key while clicking to select more than one).
3. Click **Select Components**. All selected components are also selected in the circuit window.

## 3.11 Labelling

Multisim assigns a label to a placed component, node or pin. You can modify or move the component or node label. Pin labels are set in Component Editing, as explained in Chapter 5, “Component Editing”. You can control which elements are displayed at the circuit or component level, as described in “3.8.1 Displaying Identifying Information about a Placed Component” on page 3-19. You can select a font, font style and size for labels, as described in “2.3.7 The Font Tab” on page 2-11.

Multisim also allows you to add a title block (described in “3.10.3 Adding a Title Block” on page 3-29) and additional text to your circuit (described in “3.10.4 Adding Miscellaneous Text” on page 3-29).

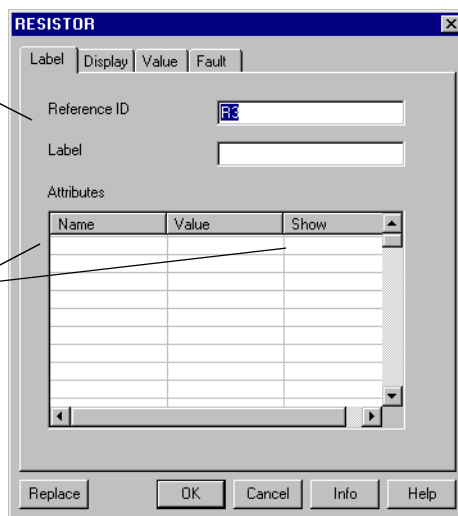
### 3.11.1 Modifying Component Labels and Attributes

Labels and, for most components, a reference ID are assigned by Multisim to a placed component. You can also assign this information using the Label tab of the component's "properties" screen.

- To assign a label and/or reference ID to a placed component:
  1. Double-click on the component. The component's "properties" screen appears.
  2. Click the Label tab:

Enter or modify the reference ID and/or label here.

Enter any name or value you wish and have it displayed with the component.



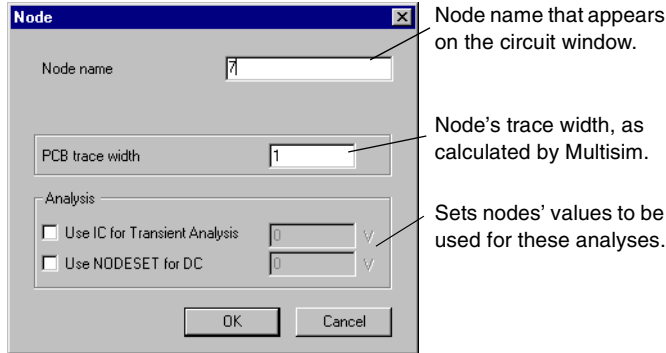
3. Enter or modify the label and/or reference ID text (which must be composed of letters or numbers only — no special characters or spaces).
  4. Enter or modify the component attributes (which can be any name or value you choose to give them.) For example, you could give the component the manufacturer name or a name that is meaningful to you such as "new resistor" or "revised May 15".
  5. Select the component attributes to display. Attributes will be displayed with the component.
- Note** If you assign the same reference ID to more than one component, Multisim warns you that this is not normally desirable. cannot proceed with assigning the same reference ID to multiple components.
6. To cancel your changes, click **Cancel**. To save your changes, click **OK**.

## 3.11.2 Modifying Node Numbers

Multisim automatically assigns a node number to each node in the circuit. You can modify and move these labels.

➤ To modify a node label:

1. Double-click on the wire. The Node properties screen appears:



2. Make the desired settings.
3. To confirm your settings, click **OK**. To cancel them, click **Cancel**.

**Note** You should exercise caution when changing node names, as they are critical to your circuit's connectivity as understood by simulation or PCB layout.

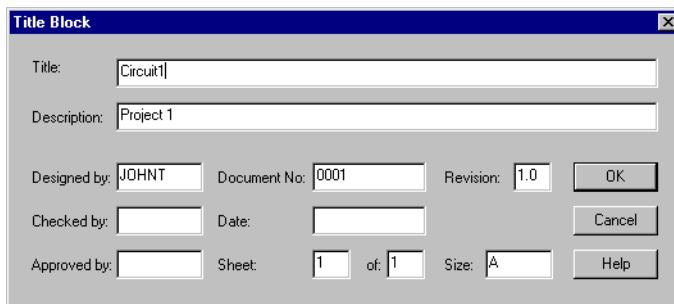
➤ To move a node label, simply select it and drag it to a new location.



### 3.11.3 Adding a Title Block

You can enter information about your circuit, including title, description and size, using the Title Block screen. Whether or not the title block appears is controlled as described in “3.2.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-4.

- To enter information about your circuit:
  1. Choose **Options/Modify Title Block**. The Title Block screen appears.



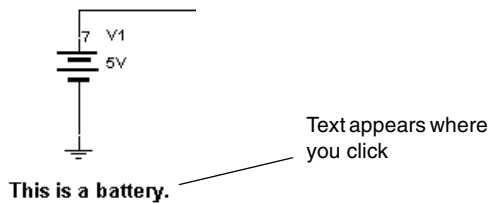
2. Enter information about your circuit and click **OK**. The title block appears at the bottom right corner of the sheet. If the title block does not appear, it may be set to be hidden. See “3.2.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-4 for details.
- To edit the contents of a title block, choose **Options/Modify Title Block** and modify the text.

### 3.11.4 Adding Miscellaneous Text

Multisim allows you to add notes to a circuit, for example to explain a particular part of a circuit.

- To add text:
  1. Choose **Place/Place Text**.
  2. Click on the location where you want the text placed. A text box appears.

3. Type the text.

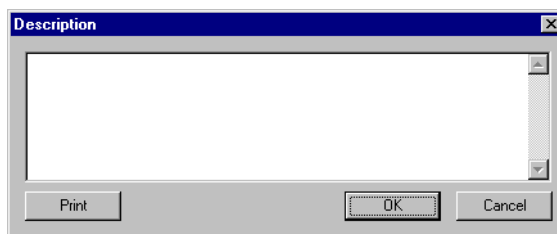


4. Click elsewhere on the circuit window to stop adding text.
- To delete text, right-click on the text box and choose **Delete** from the pop-up menu that appears, or press DELETE.
  - To change the color of text, right-click on the text box, choose **Color** from the pop-up menu that appears, and choose the desired color.
  - To change the font options for the text, right-click on the circuit window, choose **Font** from the pop-up menu that appears, and choose the desired font options. Select “Schematic text” in the Change all section.

### 3.11.5 Adding a Description

In addition to adding text to a particular portion of a circuit, you can also add general comments to your circuit. These comments can be edited, moved around the circuit window, and printed.

- To add a description:
  1. Choose **View/Show Text Description Box**. The Description window appears:



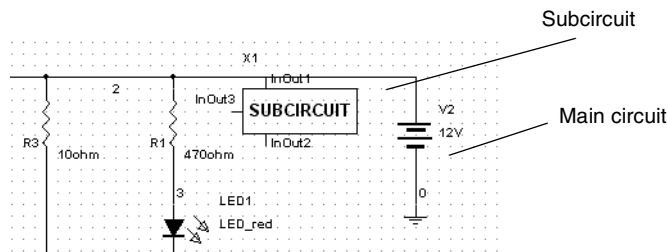
2. Enter text by typing directly into the window.
  3. When you are finished entering text, click **OK** to save the text and close the description window, or **Cancel** to close the window without saving.
- To print your description, from the Description window, click **Print**.

- To add to or modify the description, choose **View/Show Text Description Box** again. The Description window appears, with the text saved from the last time you saved it. Use the scroll bar on the right side of the window to find the desired location in the text.
- To move the description window around the circuit window, click in the title bar area and drag the window to a new location.

## 3.12 Subcircuits and Hierarchy

### 3.12.1 Subcircuits vs. Hierarchy

Multisim allows you to use one circuit inside another. The embedded circuit, or subcircuit, appears as a single icon on the circuit window of the circuit in which it is embedded, simplifying the appearance of the circuit.

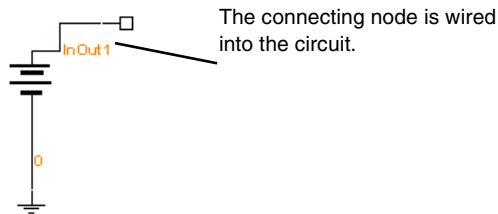


### 3.12.2 Setting up a Circuit for Use as a Subcircuit

To make it possible to wire a subcircuit into your circuit, you should add Input/Output nodes to the circuit which will be the subcircuit. These appear on the subcircuit's icon when the subcircuit is embedded in a circuit, so you can see where to add the connecting wires.

- To add an input/output node to a circuit:
  1. Choose **Place/Place Input/Output**. The cursor changes to indicate a node is ready to be placed.
  2. Click at the location where you want the input/output node placed.

3. The connecting node is placed on your circuit window. You can wire it into your circuit as with any other components.



### 3.12.3 Adding Subcircuits to a Circuit

- To add a subcircuit to a circuit:
  1. Copy or cut the desired circuit file or portion of a circuit file to the clipboard.
  2. Choose **Place/Paste as Subcircuit**. You are prompted for a new name for the subcircuit. Your cursor changes to a ghost image of the subcircuit indicating a subcircuit is ready to be placed.
  3. Click on the location in the circuit where you want the subcircuit placed (you can move it later, if necessary).
  4. The subcircuit appears in the desired location on the circuit window as an icon with the subcircuit name inside it.

The subcircuit name appears with your other components in the “In Use” list. The subcircuit icon can be manipulated as with any other components. For example, you can right-click on the icon and rotate it or set its color. You can also connect wires from the original circuit to any appropriate location in the subcircuit (that is, where potential input/output connections are available), as shown earlier in this section.

- To edit a subcircuit:
  1. Double-click on the subcircuit icon in the main circuit window that contains the subcircuit. The Subcircuit screen appears.
  2. Click **Edit Subcircuit**.

The subcircuit appears in its own window, from which you can edit the subcircuit as you would any other circuit.

For educational purposes, you can “lock” a subcircuit to make it unavailable to students for viewing or editing. See “13.3.2 Setting Circuit Restrictions” on page 13-8.

- To replace a component with a subcircuit:
  1. Select the component on the circuit window that you want to replace.
  2. Click **Place/Replace by Subcircuit**. You are prompted for a new name for the subcircuit. The component is removed from your circuit and your cursor changes to a ghost image of the subcircuit indicating a subcircuit (created from the selected component) is ready to be placed.
  3. Click on the location in the circuit where you want the subcircuit placed (you can move it later, if necessary). The subcircuit appears in the desired location on the circuit window as an icon with the subcircuit name inside it.

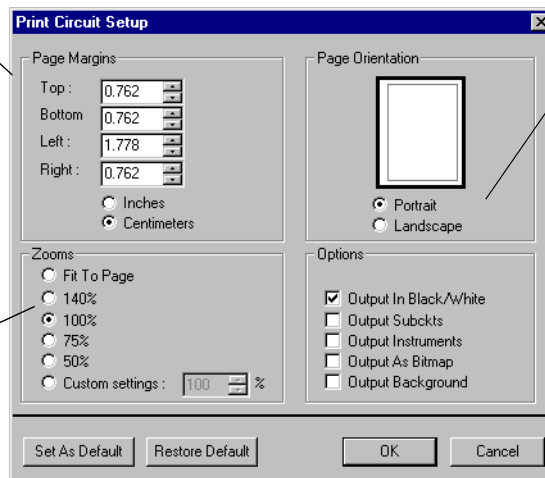
## 3.13 Printing the Circuit

Multisim allows you to control specific aspects of your printing, including:

- whether to output in color or black and white
  - whether to include the background in the printed output
  - page margins for printing
  - scaling of the circuit's image to fit the printed output.
- To set the default printing environment for circuits, use **File/Print Circuit/Print Circuit Setup**.

Set page margins for printed output.

Select an option to scale the circuit down or up in printed output.



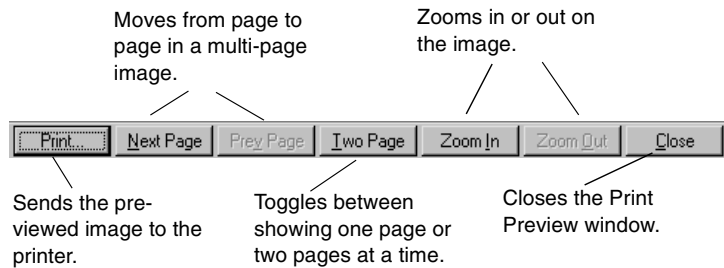
Select to print the circuit in the portrait (vertical) or landscape (horizontal) orientation.

Set your options in the lower right portion of the screen by enabling the appropriate check boxes. Select from the following options:

Print Options	Description
Output in Black/White	Prints the circuit in black and white (for non-color printers). When disabled, colored components print in shades of grey.
Output Subcircuits	Prints the circuit and subcircuit on separate sheets.
Output Instruments	Prints the circuit and faces of the instruments used in the circuit on separate sheets.
Output as Bitmap	
Output Background	Includes the background in printed output. Use for color printers or white on black output.

- To print the circuit file using the specified environment, choose **File/Print Circuit/Print**.
- To preview your printed file, choose **File/Print Circuit/Print Preview**. The circuit appears in a preview window where you can zoom in, move from page to page, and send the circuit to the printer.

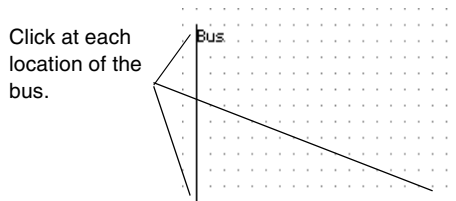
The Print Preview screen offers the following toolbar:



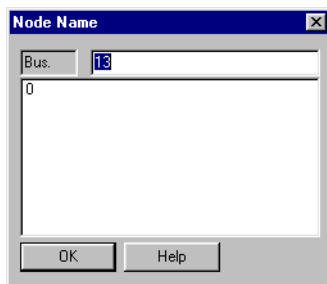
## 3.14 Placing a Bus

Buses are a set of parallel paths in your schematic that are used to connect one group of pins to another group of pins over a similar path. When implemented on a PCB, for example, a bus may in fact be a single piece of copper or series of cables carrying several binary bits representing a digital word in parallel.

- To place a bus in your circuit:
  1. Choose **Place/Place Bus**.
  2. Click on the first point for the bus.
  3. Click on the next point for the bus.
  4. Continue to click on points until the bus is complete.
  5. Double-click to mark the ending point of the bus. The bus is drawn in the same color as set for virtual components.



6. Wire the bus into your circuit by dragging a wire to any location on the bus. The Node Name screen appears:



7. If necessary, change the name shown (this name is appended to the word “Bus” to form the node name) and click **OK**.
- To change the color of the bus, right-click on it and choose **Color** from the pop-up menu that appears.
  - To change the reference ID of the bus (by default, Multisim gives it the reference ID “bus”), double-click the bus and change the reference ID in the properties screen that appears.

## 3.15 Using the Pop-up Menu

### 3.15.1 From Circuit Window, with no Component Selected

If you right-click on the circuit window with no component selected, a pop-up menu of appropriate commands appears. These commands are:

Command	Description
Place Component	Lets you browse the databases (“Multisim Master”, “Corporate Library” and “User”) for components to be placed. For details, see “3.4.1 Choosing a Component and Using the Browser Screen” on page 3-7.
Place Junction	Places a connector when you click. For details, see “3.6 Manually Adding a Junction (Connector)” on page 3-17.
Place Bus	Places a bus with segments created as you click. For details, see “3.13 Placing a Bus” on page 3-35.
Place Input/Output	Places an input/output node to connect a subcircuit to the current circuit. For details, see “3.11.3 Adding Subcircuits to a Circuit” on page 3-32.
Place Text	Lets you place text on the circuit. For details, see “3.10.4 Adding Miscellaneous Text” on page 3-29.
Cut	Removes the selected item from the circuit and stores it on the clipboard.
Copy	Lets you copy the selected item from the circuit to the clipboard.
Paste	Lets you paste the current contents of the clipboard onto your circuit.
Place as Subcircuit	Places an external circuit within the current circuit. For details, see “3.11.3 Adding Subcircuits to a Circuit” on page 3-32.
Replace by Subcircuit	Lets you replace the selected component with an external circuit. For details, see “3.11.3 Adding Subcircuits to a Circuit” on page 3-32.
Show Grid	Shows or hides grid in the background of the circuit window. This helps you place elements in specific locations on a grid. For details, see “3.2.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-4.



Command	Description
Show Page Bounds	Shows or hides the parameters of your sheet size. For details, see “3.2.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-4.
Show Title Block and Border	Shows or hides the circuit’s title block and border. For details, see “3.2.2 Displaying or Hiding the Grid, Title Block, Page Bounds and Page Borders” on page 3-4.
Zoom In	Lets you view the circuit at an increased magnification level. The default magnification levels are set in <b>Options/Preferences</b> in Workspace tab.
Zoom Out	Lets you view the circuit at a decreased magnification level.
Find	Displays a list of the reference IDs in the current circuit. You can select one or more of these reference IDs, which are then selected in the circuit window. For details, see “3.9 Finding Components in Your Circuit” on page 3-26.
Color	Lets you choose or modify the color scheme for the circuit. Overrides the defaults set in <b>Options/Preferences</b> . For details, see “2.3.3 The Circuit Tab” on page 2-7.
Show	Lets you choose what component elements appear on the circuit window. Overrides the defaults set in <b>Options/Preferences</b> . For details, see “2.3.4 The Workspace Tab” on page 2-8.
Font	Lets you select a font, font style and size for component labels and values, pin names and node names.
Wire width	Lets you select a wire width for current circuit.
Help	Opens the Multisim Help file.

### 3.15.2 From a Selected Component or Instrument

If you right-click on a selected component or instrument on the circuit window, a pop-up menu of appropriate commands appears. These commands are:

Command	Description
Cut	Removes the selected components, circuits or text.
Copy	Copies the selected components, circuits or text. For details, see “3.4.4 Copying a Placed Component” on page 3-10.
Flip Horizontal	Flips the selection vertically. For details, see “3.7 Rotating/Flipping Components” on page 3-17.
Flip Vertical	Flips the selection horizontally. For details, see “3.7 Rotating/Flipping Components” on page 3-17.
90 Clockwise	Rotates the selection 90 degrees clockwise. For details, see “3.7 Rotating/Flipping Components” on page 3-17.
90 CounterCW	Rotates the selection 90 degrees counterclockwise. For details, see “3.7 Rotating/Flipping Components” on page 3-17.
Color	Changes the color of the placed component from its default values. For details, see “3.4.6 Controlling Component Color” on page 3-12.
Help	Opens the Multisim help file.

### 3.15.3 From a Selected Wire

If you right-click on a selected wire on the circuit window, a pop-up menu of appropriate commands appears. These commands are:

Command	Description
Delete	Deletes the selected wire.
Color	Changes the color of the selected wire from its default value.
Help	Opens the Multisim help file.



# Chapter 4

## Components

### 4.1 About this Chapter

This chapter introduces you to the underlying structure and organization of the Multisim 2001 component database. It also explains how to access the database for parts and how to search the database for information.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 4.2 Structure of the Component Database

The Multisim component database is designed to hold the information necessary to precisely describe any component. It contains all the details needed for schematic capture (symbols), simulation (models), and PCB layout (footprints), as well as other electrical information. The power of the Multisim database comes from its structure: multiple levels, well-organized component groupings, and detailed information fields.

#### 4.2.1 Databases

Components are stored in three different databases:

- the “Multisim Master” database stores the components as originally designed by Electronics Workbench and shipped with Multisim; these remain the same for all users of Multisim
- the “Corporate Library” database (available only for users with the Project/Team Design module) stores components selected and, possibly, modified or created by an individual user; these remain available to any other selected users



- the “User” database stores components modified, imported or created by you; these are available only to you.

The “User” and “Corporate Library” databases are empty when you first use Multisim. You can use the “User” database to store frequently used components, favorite components or components that you import from Edaparts.com (as described in “4.3 Using Edaparts.com” on page 4-25) or create using Component editing (which is described in the following chapter).



The “Corporate Library” database (available only for users with the Project/Team Design module) is primarily intended for companies or individuals who work on projects where components with specific attributes are shared within a group or project. It can be set up by your company, or, if you wish, by Electronics Workbench. Contact us for more information about this service.

If you modify a component, thereby creating your own edition, you must store it in either the “User” or “Corporate Library” database. You cannot modify the “Multisim Master” database (this is a safety precaution to prevent corruption of the component database shipped as part of Multisim).

You choose which database you want displayed by selecting **View/Component Bars**, as explained in “4.1.2 Displaying Database Information” on page 4-2. For educational purposes, you can also hide databases, and their associated Parts Bins, from students’ view by using restrictions. See “13.3 Using Restrictions” on page 13-4.

### 4.2.2 Displaying Database Information

A separate Component toolbar exists for each database, letting you build a circuit that combines components from any or all of the available databases. By default, only the “Multisim Master” database Component toolbar is displayed.

- To select which database you want displayed, choose **View/Component Bars** and select the desired database.

The associated Component toolbars appear as follows:

**Multisim Master and  
User database  
selected:**



**Only Multisim  
Master database  
selected:**



**All databases  
selected:**



### 4.2.3 Classification of Components in the Database

Multisim divides components into logical groups, each represented by a Parts Bin. Each Parts Bin contains families of related components. The Parts Bins are listed below:

- Sources
- Basic
- Diodes

- Transistors
- Analog ICs
- TTL
- CMOS
- Miscellaneous Digital ICs
- Mixed Chips
- Indicators
- Miscellaneous
- Controls
- RF (for users with RF module)
- Electro-mechanical

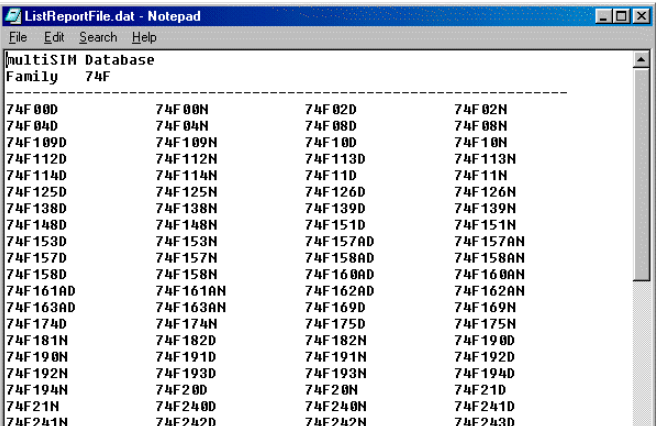
Each toolbar has two versions: ANSI (American standard) and DIN (European standard). The two standards use different icons to represent the components. Both are shown in the sections that follow.

**Note** The content of the toolbars may change as the database expands.

- To switch between ANSI and DIN symbol sets, choose **Options/Preferences**. In the Component Bin tab, select the standard you wish to use.

### 4.2.3.1 Component Families List

- To see the contents of any family:
  1. From the Browser screen that appears when you are placing a component, click **Advanced** to view the full Browser screen.
  2. In the Operations box, click **List Report**.
  3. A Notepad window appears, listing all the components stored within the currently selected family. For example:

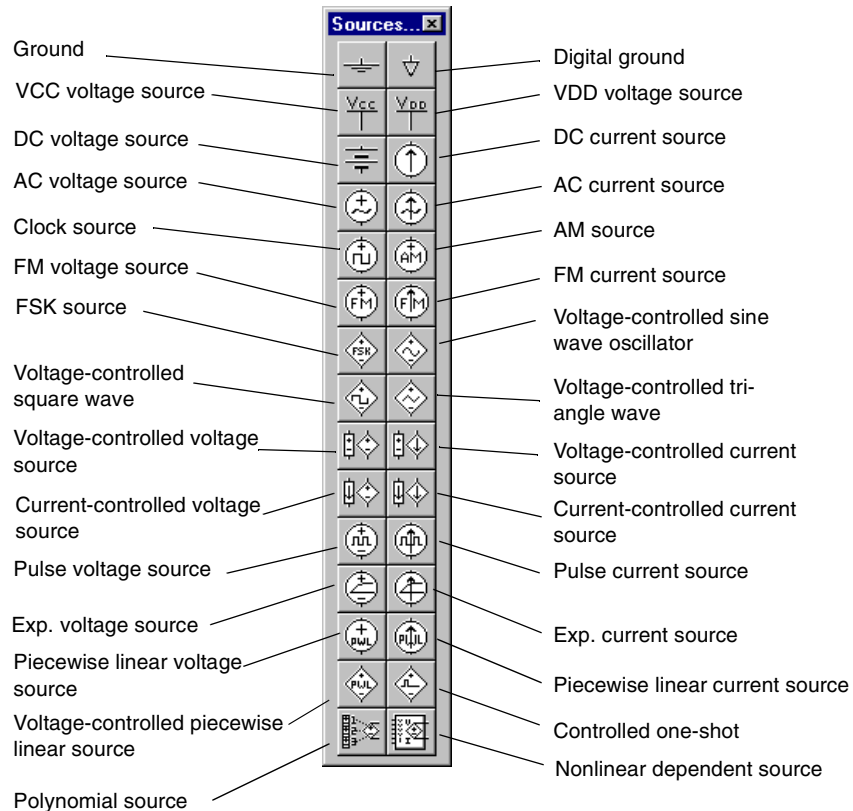


multisim Database			
Family 74F			
74F00D	74F00N	74F02D	74F02N
74F04D	74F04N	74F08D	74F08N
74F109D	74F109N	74F10D	74F10N
74F112D	74F112N	74F113D	74F113N
74F114D	74F114N	74F11D	74F11N
74F125D	74F125N	74F126D	74F126N
74F138D	74F138N	74F139D	74F139N
74F148D	74F148N	74F151D	74F151N
74F153D	74F153N	74F157AD	74F157AN
74F157D	74F157N	74F158AD	74F158AN
74F158D	74F158N	74F160AD	74F160AN
74F161AD	74F161AN	74F162AD	74F162AN
74F163AD	74F163AN	74F169D	74F169N
74F174D	74F174N	74F175D	74F175N
74F181N	74F182D	74F182N	74F190D
74F190N	74F191D	74F191N	74F192D
74F192N	74F193D	74F193N	74F194D
74F194N	74F20D	74F20N	74F21D
74F21N	74F240D	74F240N	74F241D
74F241N	74F242D	74F242N	74F243D



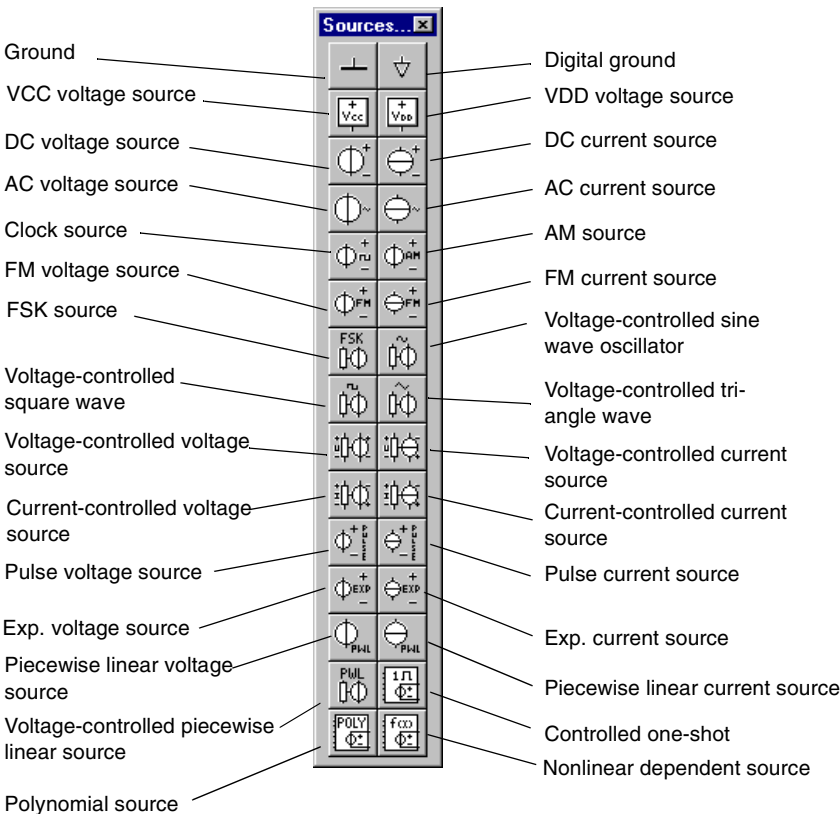
### 4.2.3.2 Sources Toolbar

#### ANSI:



**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

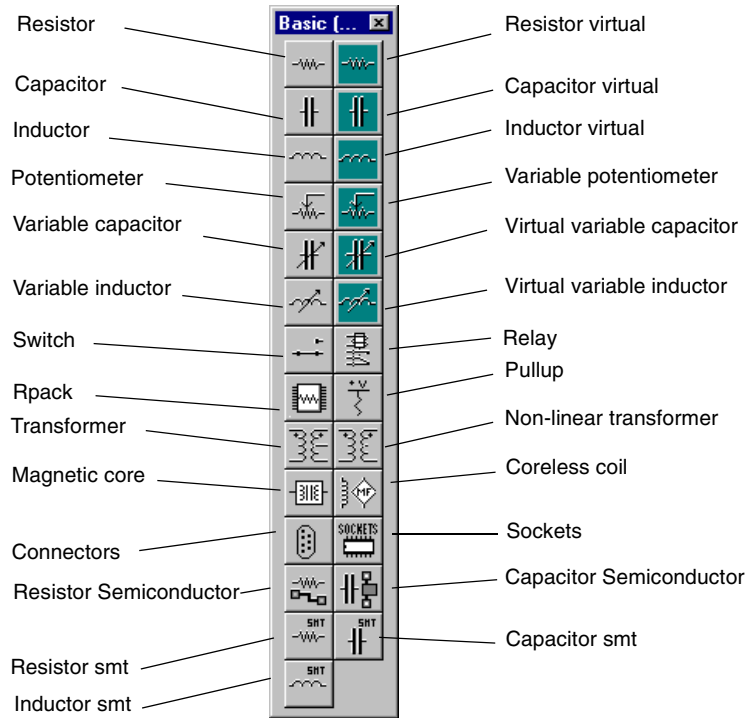
DIN:



**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

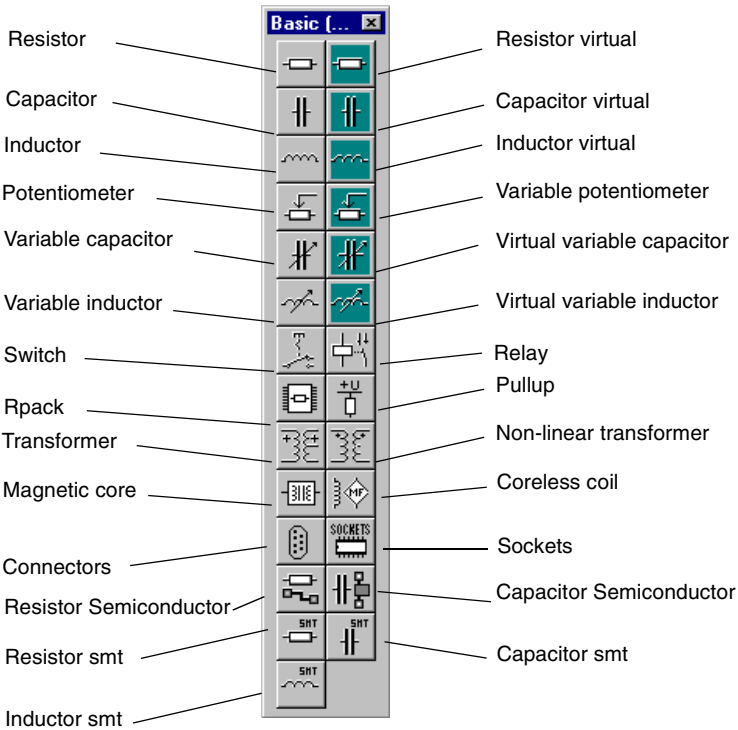
### 4.2.3.3 Basic Toolbar

#### ANSI:



**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

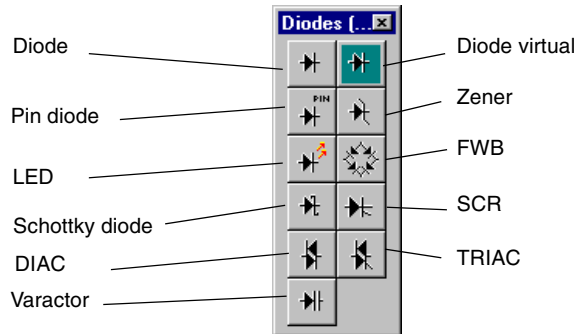
DIN:



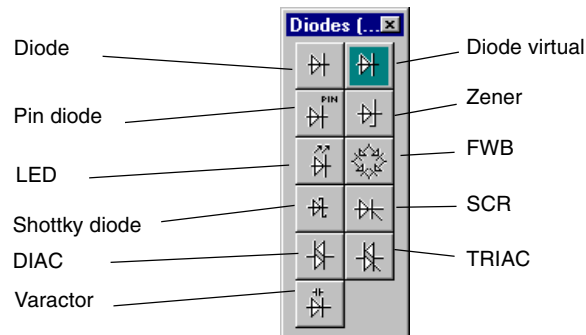
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.4 Diodes Toolbar

#### ANSI:



#### DIN:



**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

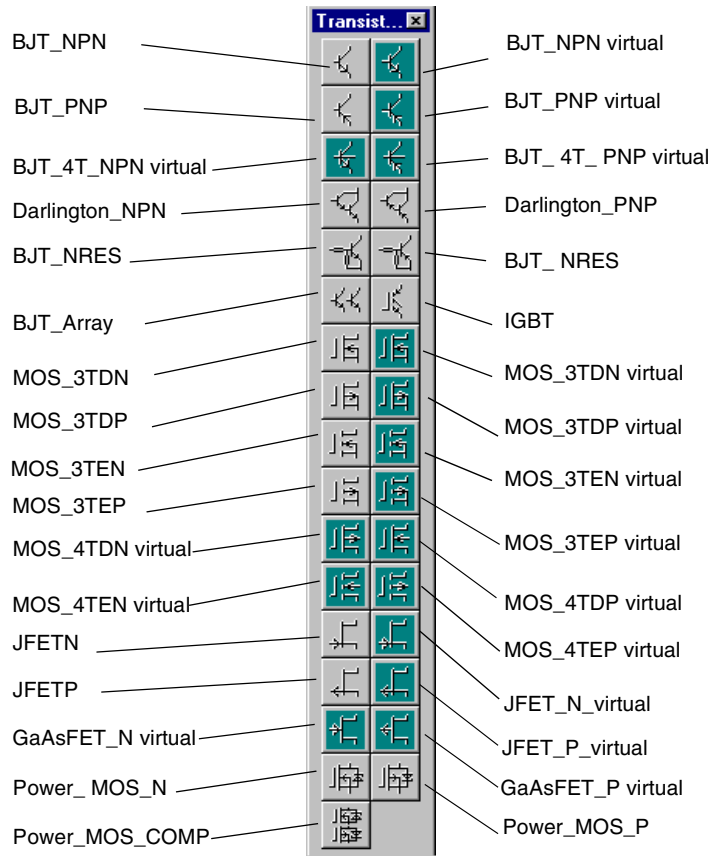
### 4.2.3.5 Transistors Toolbar

**ANSI:**

The image shows a screenshot of the 'Transist...' toolbar in Multisim. The toolbar contains 24 icons arranged in a grid. Lines connect these icons to labels on the left and right sides of the image. The labels are as follows:

Left Labels	Right Labels
BJT_NPN	BJT_NPN virtual
BJT_PNP	BJT_PNP virtual
BJT_4T_NPN virtual	BJT_4T_PNP virtual
Darlington_NPN	Darlington_PNP
BJT_NRES	BJT_NRES
BJT_Array	IGBT
MOS_3TDN	MOS_3TDN virtual
MOS_3TDP	MOS_3TDP virtual
MOS_3TEN	MOS_3TEN virtual
MOS_3TEP	MOS_3TEP virtual
MOS_4TDN virtual	MOS_4TDP virtual
MOS_4TEN virtual	MOS_4TEP virtual
JFETN	JFET_N_virtual
JFETP	JFET_P_virtual
GaAsFET_N virtual	GaAsFET_P virtual
Power_MOS_N	Power_MOS_P
Power_MOS_COMP	

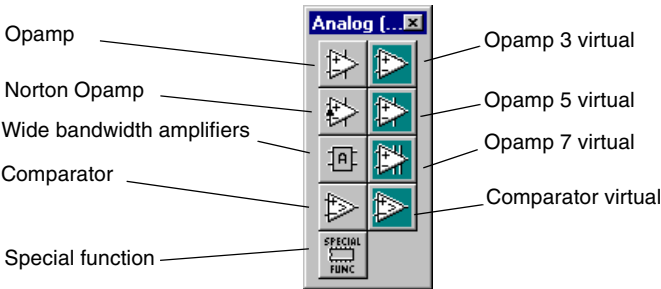
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

**DIN:**

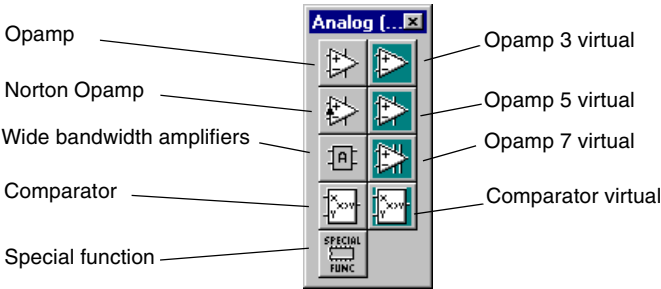
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.6 Analog Toolbar

**ANSI:**



**DIN:**

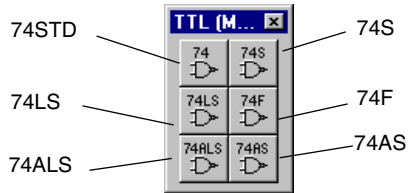


**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

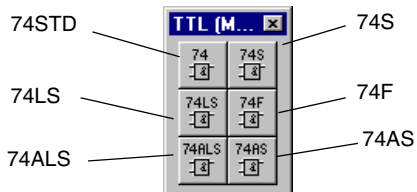


### 4.2.3.7 TTL Toolbar

#### ANSI:



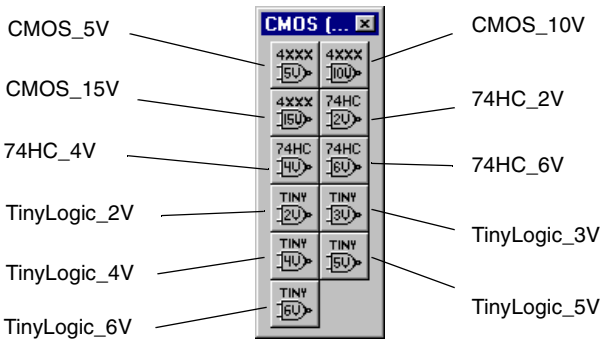
#### DIN:



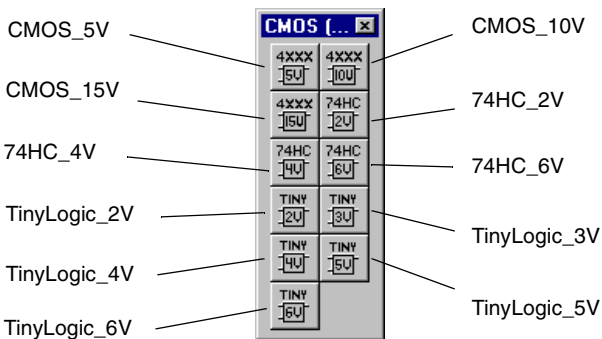
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.8 CMOS Toolbar

**ANSI:**



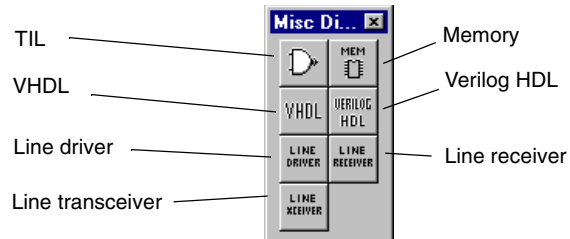
**DIN:**



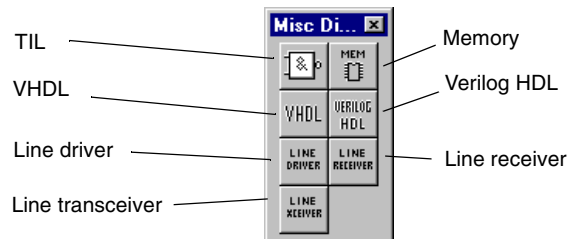
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.9 Miscellaneous Digital Toolbar

#### ANSI:



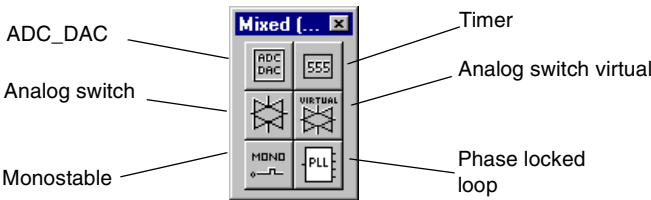
#### DIN:



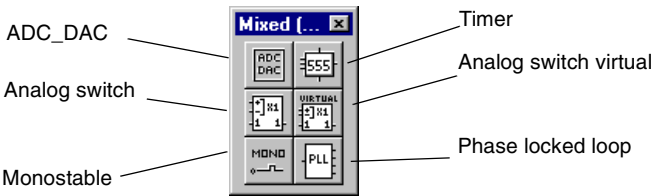
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.10 Mixed Chips Toolbar

**ANSI:**



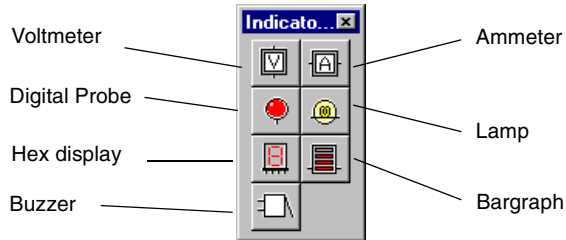
**DIN:**



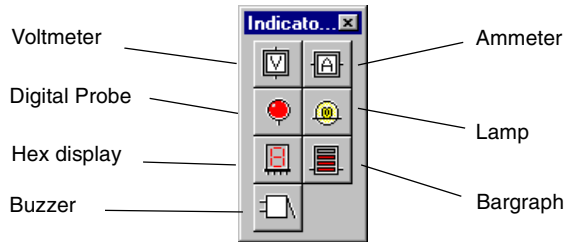
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.11 Indicators Toolbar

#### ANSI:



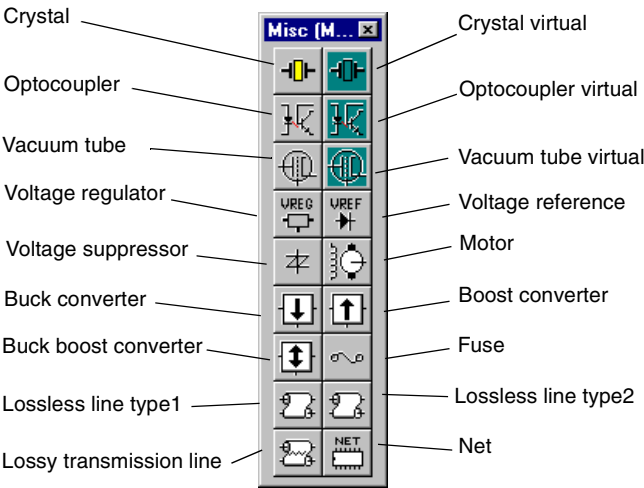
#### DIN:



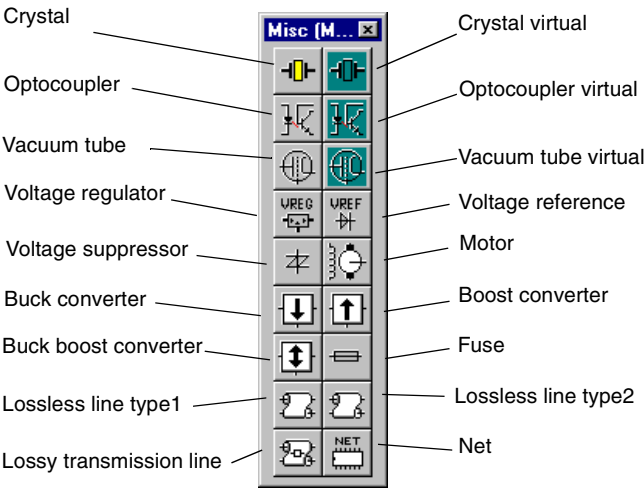
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.12 Miscellaneous Toolbar

**ANSI:**



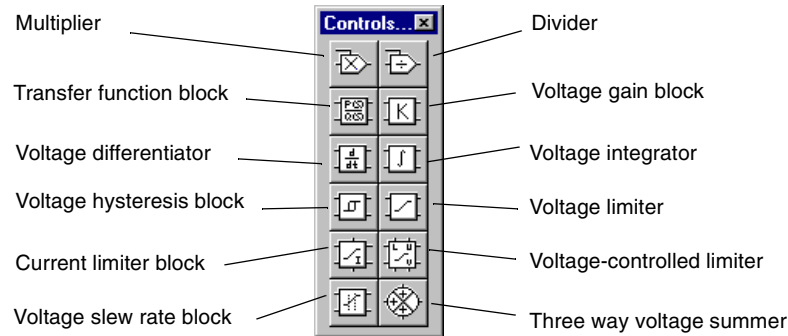
**DIN:**



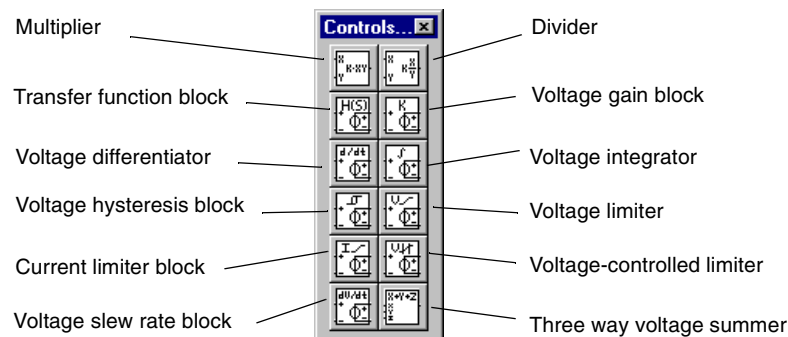
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.13 Controls Toolbar

#### ANSI:



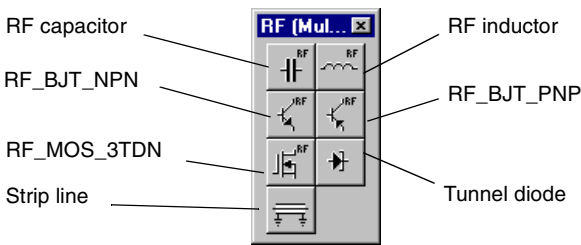
#### DIN:



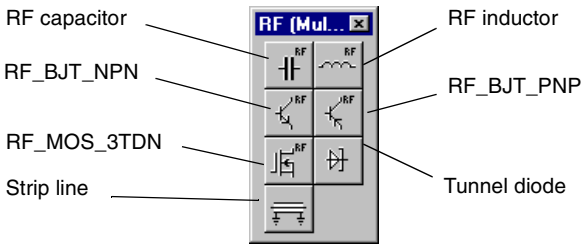
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

### 4.2.3.14 RF Toolbar

**ANSI:**



**DIN:**



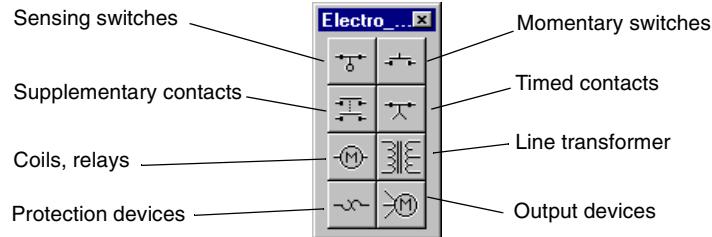
**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.



### 4.2.3.15 Electro-mechanical Toolbar



#### ANSI:



#### DIN:

Same as ANSI.

**Note** For details about these component families, see the appendices in either the printed manual or the PDF file shipped with Multisim.

## 4.3 Locating Components in the Database

You can locate components in a specific component family within a specific database by either browsing through the available data, or by searching for a component that meets specific criteria.

### 4.3.1 Browsing for Components

When you are placing a component, the Browser screen that appears lets you browse for components anywhere in the Multisim database. For details, see “3.4.1 Choosing a Component and Using the Browser Screen” on page 3-7.

### 4.3.2 Standard Searching for Components

Multisim comes with a powerful search engine to help you quickly locate components if you know some information about the type of component you need. Multisim searches its database for components that meet your criteria and presents them to you, enabling you to choose the component that most suits the needs of your application from the list of candidates. You might need to select a component with a specific package because of space limitations, with a specific power dissipation because of your design, with a specific electrostatic discharge

because of its relationship to other components, or with a specific manufacturer because of company requirements.

- To perform a standard search of the database:
1. Display the Browser screen, normally by clicking on the appropriate Parts Bins and component family in which you want to search, or by choosing **Place/Place Component**.
  2. In the **Database Name** field, select the desired database (Multisim Master, Corporate Library or User).
  3. In the **Component Family** field, select the desired Component family.
  4. Click **Search**. The Search screen appears:

**Search - BJT\_NPN**

Component Name(String):

Manufacturer(String):

Footprint(String):

Common Parameters:

Thermal Resistance Junction(Num):	=		AND		=
Thermal resistance Case(Num):	=		AND		=
Power Dissipation(Num):	=		AND		=
Derating Knee Point(Num):	=		AND		=
Min. Operating Temperature(Num):	=		AND		=
Max. Operating Temperature(Num):	=		AND		=
ESD(Num):	=		AND		=

In the fields above, enter any information you have on the part you are searching for. You can use characters and any SQL acceptable symbols (e.g. \*, ?, etc.) in string-fields, enter one of the symbols [=, >, <, >=, <=, <, >, !=] followed by a number in number-fields.  
EXAMPLES:

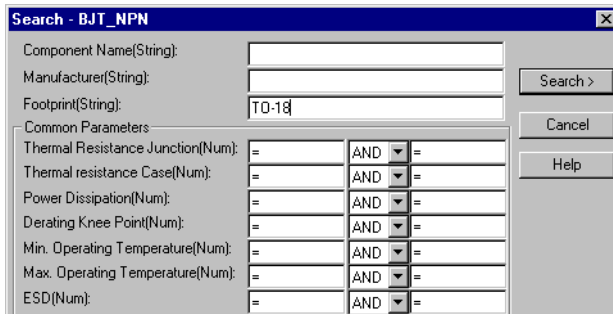
5. In the desired fields, enter your search criteria (you must enter at least one item). Enter text or numbers using scientific notation. Numbers must be prefixed with a symbol (for example, “=”). You can also use “>”, “<”, “>=”, and “<=” in conjunction with numbers, to set a range. For text, case is not considered, and you can use the “\*” wildcard to find partial strings.

For example, in the **Footprint** field:

- “CASE646-06” finds only the exact string “CASE646-06”
- “\*06” finds any string ending with “06”
- “CASE\*” finds any string starting with “CASE”
- “\*646\*” finds any string with “646” inside it

For more information about fields in the Search screen, see “4.4.1.2 Common Parameters” on page 4-33.

The following example shows the value you would enter to find a transistor component with a footprint of “TO-18”:

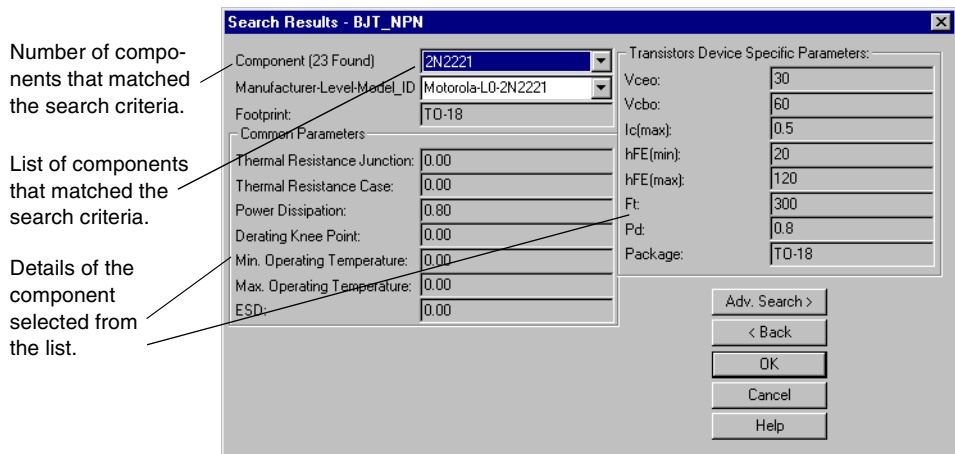


6. To carry out the search, click **Search**. When the search is complete, the Search Results screen appears.

**Tip** The more specific your search criteria, the smaller the number of matching components.

- To select a component from the search results:

When the search is complete, the Search Result screen appears, displaying information about the first component that matched your criteria. The **Component** drop-down list contains a list of all the components that matched your criteria. For example, using the search example above, the results look like this:



From the **Component** drop-down list, select the component you are interested in. To view information about any component found by the search, simply choose it from the drop-down list and the display fields change accordingly.

7. To place the selected component, click **OK**. You return to the circuit window, where you can place the component by clicking the desired location on the screen.
- To return to the Browser screen, click **Back** or **Cancel**.

## 4.4 Using Edaparts.com

The Edaparts.com Web site is designed for Multisim users, where you can:

- search for components to download into your Multisim User database
- update your Multisim Master database with new components created specifically for your edition of Multisim
- search the suppliers' Internet sites for specific parts
- request quotes from suppliers.

The first time you visit the site, you will need to create a password to log on.

### 4.4.1 Locating and Downloading Components

The following procedure explains how to launch the Web site from Multisim and then navigate through the Web site to locate and then download components. There are also instructions on the Web site to assist you at every step of the process.

#### 4.4.1.1 Locating Components

- To locate and download components:

**Note** Components downloaded from Edaparts.com do not have simulation models and may not have footprint information. To simulate these components, you must add model information using Component Editing. Footprint information can also be added using Component Editing. For details about adding model and footprint information see “5.10 Creating or Editing a Component Model” on page 5-26 and “5.11 Creating and Editing Component Footprints” on page 5-30.



1. Click the Edaparts.com button or choose, **Tools/Edaparts.com**. Your default Internet browser opens, loaded with the Edaparts.com Web site.

or

Use your Internet browser to navigate to <http://www.Edaparts.com>.




2. To search for a component, click **FreeTradeZone**. You are prompted for a password.
3. Follow the directions on the login screen to either login (if you already have a password) or create new login information. After you have logged on, the “freetradezone” screen

appears divided into three main sections; “research it”, “find it”, or “buy it”. For more information on the “find it” and “buy it” sections of this screen, see “4.3.3 Finding, Buying and Requesting Quotes for Parts” on page 4-30.

4. To search for components that you can then download, click “research it”.
5. Enter a part number, manufacturer or keyword in the fields, by following the instructions on the screen.
6. Click **Search**. The search results are displayed. If your search criteria results in multiple categories of components, the categories are listed with a folder icon to the left of the category.

➔ Your search criteria found devices in multiple categories. Select one of the categories below to view parts.

#### Select Categories

-  Integrated Circuit - Analog - Power Supply - Voltage Reference
-  Integrated Circuit - Consumer - Radio Receiver Circuit - FM
-  Integrated Circuit - Consumer - Remote Control - Transmitter/Encoder

- If your search criteria results in individual components, the components are listed.

<input type="checkbox"/>	<a href="#">LM1875T</a>	Single Audio Amplifier	<a href="#">National Semiconductor Corp</a>	Active			
<input type="checkbox"/>	<a href="#">LM1875T/A+</a>	Single Audio Amplifier	<a href="#">National Semiconductor Corp</a>	Discontinued			
<input type="checkbox"/>	<a href="#">LM1875T/B+</a>	Single Audio Amplifier	<a href="#">National Semiconductor Corp</a>	Discontinued			

- If your search criteria has no results, click the **Back** button on your browser and try again.

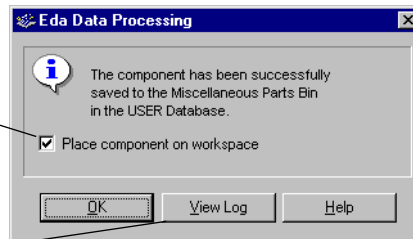
7. If your search resulted in categories, display the list of components contained in the category by clicking the folder icon beside the desired category.
8. Click the Manufacturer part number (in blue) of the desired component. More details about the component appear.

### 4.4.1.2 Downloading Components Using the Edaparts.com Command

- After you have selected a component using the procedure above, you can download that component. If you accessed Edaparts.com from within the Multisim program, use the following procedure:
1. To proceed to download the selected component, click **Download to Multisim** in the bottom right section of the screen. The Eda Data Processing screen appears.

Select to place the component on the active Multisim workspace and save it in the Miscellaneous Parts Bin of the User database.

Opens the View Log that shows the download status and component data.



2. Select from the following options:
  - To place the component on the active Multisim workspace and save it in the Miscellaneous Parts Bin of the User database, select **Place component on Workspace**.
  - To save the part to the Miscellaneous Parts Bin of the User database, click **OK**.
  - To view the download status of the component being saved, click **View Log**. The View Log screen appears as shown below:

Indicates if the component has a footprint.

Indicates if the component has a symbol.

Indicates the method used to create the symbol being loaded. There are two possible methods: "Pin Table" (Pinout information) and "Part Type" (MSM understands the type of part).

Comp_Name	Read	Save to DB	Footprint	Symbol	Symbol_Type
MC7497F	OK	OK	N/A	Yes	Pin Table

Parts Data:

Comp\_Name = MC7497F  
Manufacturer\_Name = Motorola Semiconductor Products Inc  
Comp\_Type = ANALOG  
\*\*\*\*\* Package\_Data

Processing message:

Parsing EDA parts data.  
Please wait ...  
Read the EDA parts data from file: C:\TEMP\W0T1IDOC.MXM  
Warning: Syntax error in line (2): No right mark '>'

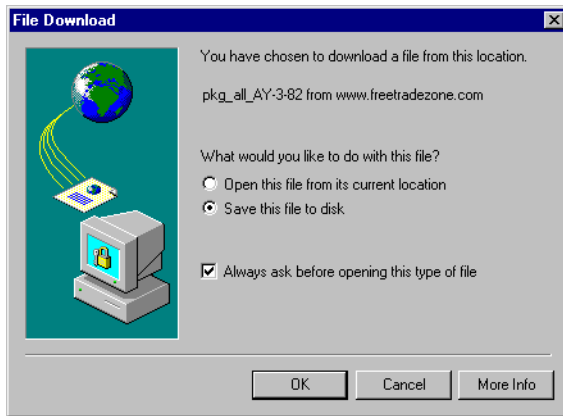
Part information loaded with the component.

Processing information

**Note** A component will not be saved if it does not have pinout information or a package type.

### 4.4.1.3 Downloading Components Using your Internet Browser

- If you have accessed Edaparts.com by navigating to the Web site using your Internet browser, use the following procedure to download the selected component:
1. To proceed to download the selected component, click **Download to Multisim** in the bottom right section of the screen. The Windows File Download screen appears.



2. To save the component and to load into Multisim at a later time, select **Save this file to disk**. A Windows Save As screen appears.
  3. Select the desired location to save the component on a computer that has Multisim installed. The component is saved with an .mxml file extension.
- To open a saved component file and load it into the Multisim User database:
1. Double click on a component file that has a .mxml file extension. Your Multisim program launches and the Eda Data Processing screen appears.
  2. Select from the following options:
    - To place the component on the active Multisim workspace and save it in the Miscellaneous Parts Bin of the User database, select **Place component on workspace**.
    - To save the part to the Miscellaneous Parts Bin of the User database, click **OK**.
    - To view the download status of the component being saved, click **View Log**.



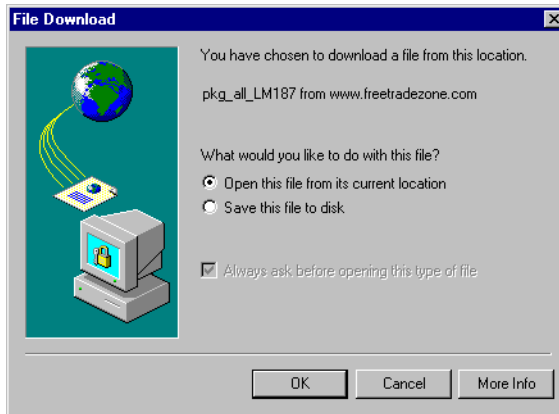
## 4.4.2 Updating your Multisim Master Database

This procedure explains how to update your Multisim Master database with new components that have been created specifically for your edition of Multisim.

- To update your Multisim Master database using Edaparts.com:



1. Click the Edaparts.com button or choose, **Tools/Edaparts.com**. Your default Internet browser opens, loaded with the Edaparts.com Web site.
2. Click the name of your edition of Multisim (e.g. Professional, Power Professional, Educational, Personal and Student). The Windows File Download screen appears.



3. Select “Open this file from its current location”.
4. Click **OK**. Your Multisim program opens, displaying the Data Merge screen and begins to update your database. When the update is completed, the Data Merge screen displays “done” and the screen closes.
5. To stop the update, click **Cancel**.

## 4.4.3 Finding, Buying and Requesting Quotes for Parts

You can use the Edaparts.com web site to search suppliers’ Internet sites for specific parts and requesting quotes for parts from suppliers. For more information about using these functions, see the instructions under the “find it” and “research it” sections of the Edaparts.com Web site.

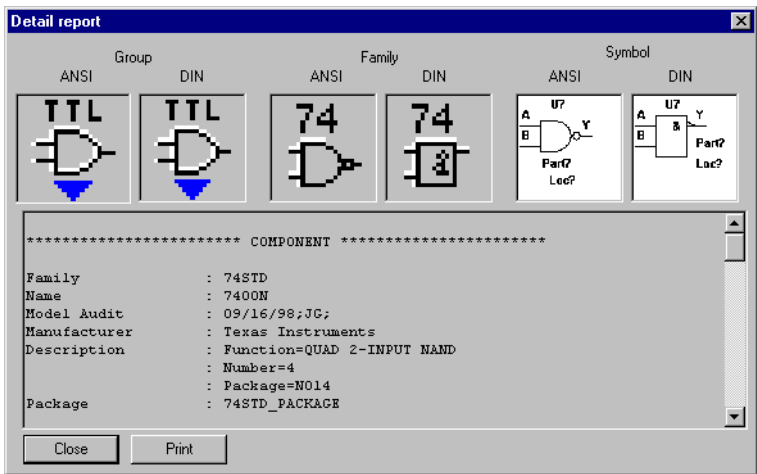
# 4.5 Types of Information Stored for Components

The Multisim databases store information about components in pre-defined fields (that is, fields that are pre-filled in Multisim) and user fields (that is, fields you can use to capture information that you want to record about a component). Both types of information appear in the Browser and Search screens. User fields only appear if you have the Project/Team Design module.

Multisim also offers a detailed report of information about components, their models, and their packages.

➤ To see this report:

From the Browser screen, click **Detail Report**. A detailed report appears for you to view or print. For example:



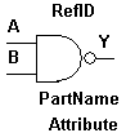
## 4.5.1 Pre-Defined Fields

For each component, the following information is stored in the “Multisim Master” database:

- general information
- common parameters
- component-specific data.

### 4.5.1.1 General Information

The following fields appear on the Browser screen:

Field	Description	Example
Database Name	Name of Multisim database in which the component is stored.	Multisim Master
Component Family	Name of family to which the component belongs. Determines which toolbar icon is used to place the component.	74S
Component Name	Name of the individual component.	74S00D
Footprint	Footprint for the component (real components only). Used in Ultiboard or other vendors' PCB layout products.	DO14
Function	Describes the component.	QUAD 2-INPUT NAND
Model Data	Contains the model's code.	in SPICE
Manufacturer Name	Name of the company that manufactures the component. Recorded in the Bill of Materials.	Texas Instruments
Model Level ID	Used during simulation (could be SPICE code, VHDL, Verilog HDL, etc.)	L.0
Symbol	Symbol used (either ANSI or DIN) to represent the component during schematic capture.	

### 4.5.1.2 Common Parameters

This is information specifying the parameters that are common to all components. This information can be searched using the basic search, and appears on the Search screen.

Field	Description	Example
Thermal Resistance Junction	The thermal resistance within the component (watts or degrees centigrade).	0.00
Thermal Resistance Case	The thermal resistance of the whole case/package (watts or degrees centigrade).	0.00
Power Dissipation	The power dissipation of the whole component (watts).	0.08
Derating Knee Point	The point at which the component's power starts being re-rated (degrees centigrade).	0.00
Min. Operating Temperature	Minimum operating temperature for the component (degrees centigrade).	0.00
Max. Operating Temperature	Maximum operating temperature for the component (degrees centigrade).	0.70
ESD	Electrostatic discharge that the component can tolerate (degrees centigrade).	0.00

### 4.5.1.3 Component-Specific Data

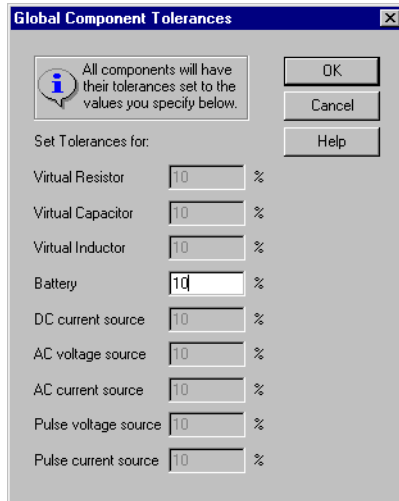
This is important electrical information that is different for each type of component. It is often needed in advanced searches. For more details, see the component's description in the corresponding appendix.

## 4.6 Component Nominal Values and Tolerances

Multisim uses the nominal values for each component in simulation.

When you want to observe circuit behavior that is more typical of real world results, you can choose to use tolerances instead. The components in this case randomly introduce variances to simulate the performance of actual, physical components. Tolerance settings affect simulation results. For example, a 1 kohm resistor with a 10% variance could vary 100 ohms either way.

- To set the tolerances to be used for components:
1. Choose **Simulate/Global Component Tolerances**. If you have undeclared tolerances, you see a screen that allows you to set a percentage to be applied to undeclared tolerances in the circuit for each type of eligible component in the circuit. For example:



Only those components that exist in the active circuit can have values assigned to them in this screen.

2. Enter the desired variances.
3. To cancel your changes, click **Cancel**. To save them, click **OK**. The simulation will now use random values as specified.



# Chapter 5

## Component Editing

### 5.1 About this Chapter

This chapter explains how to create, edit, copy, or delete a component, and how to manage the Multisim 2001 component database. It also explains how to load into the Multisim 2001 database any models that you may have developed, obtained or purchased, and how to create simulation models using Multisim's Model Makers or code modelling.



**Note** Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description. Some of the menu items and Design Bar buttons used in component editing are hidden if you have selected **Options/Simplified Version**. For more information about the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

### 5.2 Introduction to Component Editing

You can modify any component stored in the Multisim component database. (See Chapter 4, “Components” for more information on this database.) For example, an existing component might now be available in a new package (originally pin-through hole, now surface mount). You can easily copy the component information and change only the package details to create this new component. You can also create your own component and place it into the database or load a component from another source.

**Note** Multisim's database stores extensive information about components. Creating a component, depending on its use, may require entering many details. Where possible, we recommend that you modify an existing, similar component, rather than to create one.

As described in Chapter 4, “Components”, in the component database each component is identified by the following types of information:

- general information (such as name, manufacturer, date and author) — see “5.4 Editing a Component, General Properties” on page 5-8 for details

- symbol (pictorial representation of the component for schematic capture) — see “5.9 Editing and Creating a Component Symbol” on page 5-17 for details
- model (information used to represent the actual operation/behavior of the component during simulation) — necessary only for a component that will be simulated. See “5.10 Creating or Editing a Component Model” on page 5-26 for details.
- footprint (the package that Multisim uses when exporting a schematic containing this component to a PCB Layout package such as Ultiboard) — see “5.11 Creating and Editing Component Footprints” on page 5-30 for details
- electronic parameters of the component — see “5.5 Editing a Component, Electronic Parameters” on page 5-9 for details
- user fields (if used to further define the components)

In addition, all components can be organized by using Database Management, which lets you:

- add and remove component families from the User database
- modify user field titles for any database
- add and change toolbar button symbols

**Note** If you modify any information about a component in the Multisim Master database, you must store the modified information in the “User” or “Corporate Library” database to prevent corruption of the Multisim Master database.

---

**Warning:** If you modify any piece of information about any component in the “Corporate Library” or “User” databases, you are prompted for a new name for the component. If you do not give a new name, Multisim saves the changes to the original location, so the original “User” or “Corporate Library” database component information is lost. It is recommended that you do provide a new name, even for minor variations to the original.

---

For educational purposes, you may choose to disallow component editing by students. You can use restrictions to prevent students from editing components. See “13.3 Using Restrictions” on page 13-4 for more information.



## 5.3 Managing the Database

The Database Management screen allows you to:

- add and remove component families from the User or Corporate Library database. You cannot add or remove families in the Multisim Master database.
- set up or modify user field titles for any database
- add or change toolbar button symbols for component families in User or Corporate Library databases.

➤ To invoke the Database Management screen:

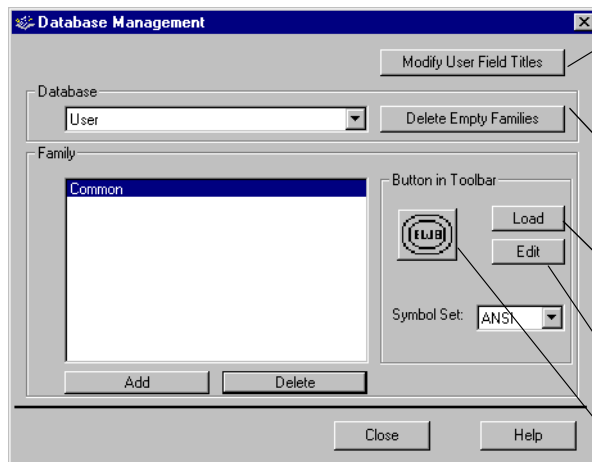


1. Click the Component Editing button on the Design Bar and choose **Database Management** from the pop-up menu that appears.

or

Choose **Tools/Database Management**.

The Database Management screen appears.



Lets you edit the user field titles which appear on the User Field tab of the Component Properties screen for components in the selected database.

Lets you delete all empty component families in the selected database

Lets you select a different toolbar button to be associated with the selected family and symbol set.

Lets you modify and existing toolbar button

Shows the toolbar button associated with the selected component family

➤ To add a component family to the User or Corporate Library database:

1. In the Database list, choose the database in which you want to work.
2. Click **Add**. You are prompted to select a family group (e.g. Transistors, Analog) and enter a family name (e.g. Sample).
3. Click **OK**. You return to the Database Management screen.
4. A default icon of the group is automatically assigned to the new family. To change it, follow the procedure below.

5. Your new family appears in the associated component family (group) toolbar. For example, a new Sample family will appear in the Passive Parts Bin of the library you selected.
- To load a component toolbar button:
    1. With the desired component family selected, click **Load**. You will be prompted to navigate to the desired toolbar button file.
    2. Select the desired toolbar button file and click **Open**.
    3. The new toolbar button will be displayed on the Database Management screen in the Button in Toolbar area.
  - To edit the default component toolbar button:
    1. With the desired component family selected, click **Edit**.
    2. Your paint program is launched and the bitmap file of the button opened.
    3. Edit the bitmap file to your requirements and then save and close the paint program.
    4. The revised toolbar button will appear as the component toolbar button.
  - To delete a component family from the User or Corporate Library database.
    1. In the Database list, choose the database from which you want to delete a component family.
    2. In the Family list, choose the component family that you want to delete.
    3. Click **Delete**. You are prompted to confirm your deletion.
    4. To complete the delete process, click **OK**. You return to the Database Management screen.
  - To modify User Field Titles:



1. Click **Modify User Field Titles**. The Modify User-Field Titles screen appears.

2. Enter the desired information in the Title fields.
3. Click **OK**. You return to the Database Management screen.

**Note** Modifying user field titles changes the titles for all databases, not only the selected database. User field titles can be changed again at any time.

- To delete empty families from the User or Corporate Library database:
  1. Click **Delete Empty Families**. You are prompted to confirm your command to delete all empty family folders.
  2. To proceed click **OK**. All empty family folders will be deleted from the Family list on the Database Management screen for the selected database.

## 5.4 Editing Components

- To edit an existing component:

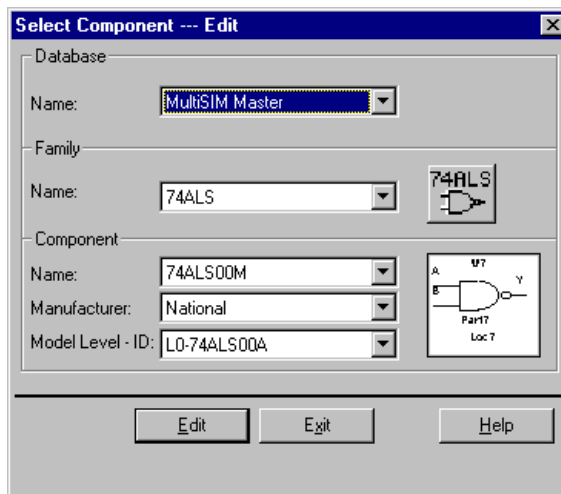


1. Click the Component Editing button on the Design Bar and choose **Edit Component** from the pop-up menu that appears.

or

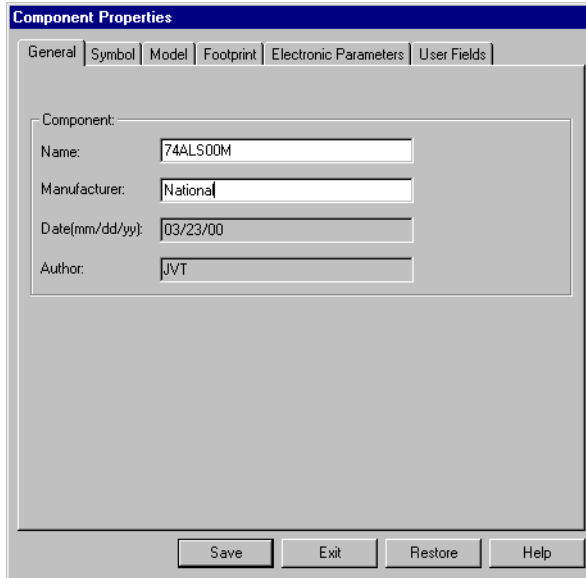
Choose **Tools/Edit Component**.

The Select Component screen appears:



2. From the **Database Name** list, choose the database containing the component you want to edit.
3. From the **Family Name** list, choose the component family containing the component you want to edit. The family's icon appears on the screen.
4. From the **Component Name** list, choose the component you want to edit. The component's icon appears on the screen.

5. If necessary, choose the Manufacturer and **Model Level ID** of the component you want to edit (if more than one manufacturer or model exists).
6. To continue, click **Edit**. (To cancel, click **Exit**.) The Component Properties screen appears, consisting of multiple tabs:



The screenshot shows the 'Component Properties' dialog box. The 'General' tab is active, displaying fields for Name (74ALS00M), Manufacturer (National), Date (03/23/00), and Author (JVT). The dialog includes tabs for General, Symbol, Model, Footprint, Electronic Parameters, and User Fields. At the bottom are buttons for Save, Exit, Restore, and Help.

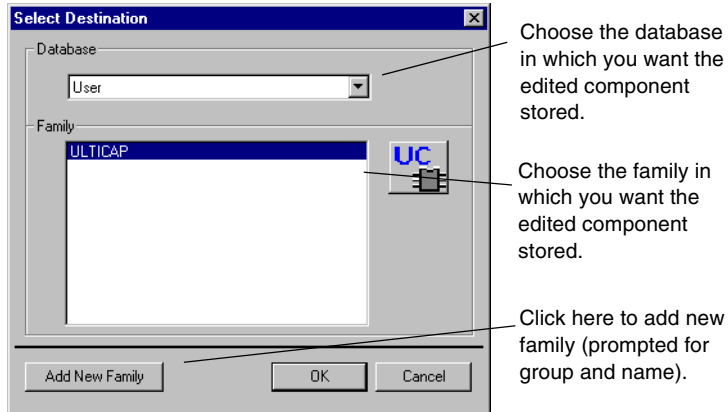
You can click **Save** from any of the tabs to save the edited component.

For information on those tabs see:

- “5.4 Editing a Component, General Properties” on page 5-8
- “5.9 Editing and Creating a Component Symbol” on page 5-17
- “5.10 Creating or Editing a Component Model” on page 5-26
- “5.11 Creating and Editing Component Footprints” on page 5-30
- “5.5 Editing a Component, Electronic Parameters” on page 5-9

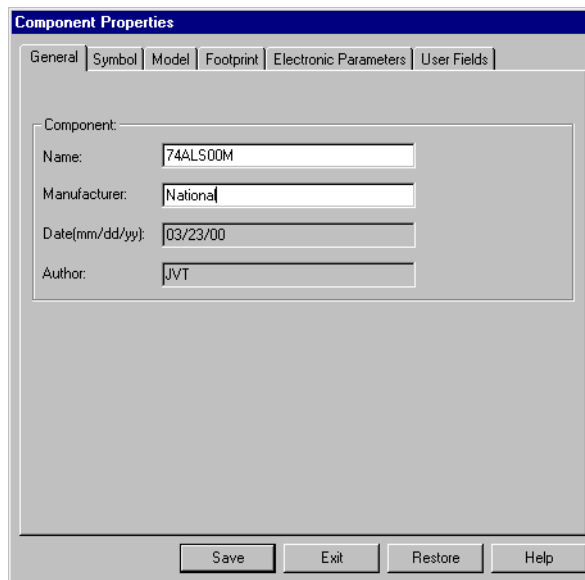


You are prompted to specify the location where you want the edited component saved.



## 5.5 Editing a Component, General Properties

The General tab of the Component Properties screen allows you to modify the component's name and manufacturer. The data and author are drawn from system information and cannot be changed.



# 5.6 Editing a Component, Electronic Parameters

The screen consists of two sets of fields. The device parameters vary depending on the type of component. The Appendices of this manual (provided in print format only with the Power Professional edition) describe each component family’s parameters in detail.

Component Properties

GeneralSymbolModelFootprintElectronic ParametersUser Fields

Common Parameters:

Thermal Resistance Junction:

0.00

Thermal Resistance Case:

116.00

Power Dissipation:

0.01

Derating Knee Point:

0.00

Min. Operating Temperature:

0.00

Max. Operating Temperature:

70.00

ESD Rating:

0.00

Device Specific Parameters:

Label	Value
Function	QUAD 2-INPUT NAND
Number	4
Package	M14A

Save

Exit

Restore

Help

!

The electronic parameters displayed do not affect the simulation results. These parameters are only used for detail report.

Common parameters are the same for all components.

Device parameters vary depending on component.

The common parameter fields are common to all components. These are:

Field	Description
Thermal Resistance Junction	Enter or modify the thermal characteristics within the component (from the junction to the case), in watts or degrees centigrade.
Thermal Resistance Case	Enter or modify the thermal characteristics of the whole package (component) in watts or degrees centigrade.
Power Dissipation	Enter or modify the power dissipation of the component, in watts.
Derating Knee Point	Enter or modify the temperature at which the power of the component/package begins to be de-rated, in order to operate the device in its safe operating range. Use degrees centigrade.
Min. Operating Temperature	Enter or modify the lowest ambient temperature at which the component can operate reliably. Use degrees centigrade.

Field	Description
Max. Operating Temperature	Enter or modify the highest ambient temperature at which the component can operate reliably. Use degrees centigrade.
ESD Rating	Enter or modify the electro-static discharge for the component.

## 5.7 Adding Components

Multisim includes a Create Component Wizard to step you through the process of creating a component.

- To create a component:



1. Click the Component editing button on the Design Bar and choose **Create Component** from the pop-up menu that appears.

or

Choose **Tools/Create Component**.

The Create Component Wizard appears.

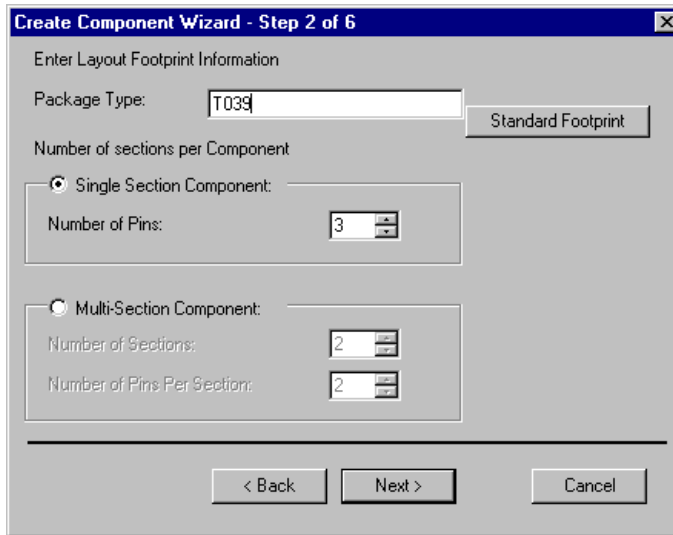
The screenshot shows the 'Create Component Wizard - Step 1 of 6' dialog box. It has a title bar with a close button. The main area is titled 'Enter Component Information'. It contains three text input fields: 'Component Name:' with the text 'test', 'Manufacturer Name:' with the text 'ccc', and 'Component Type:' with a dropdown menu showing 'Analog'. Below these fields are three radio button options: 'I will use this component for both simulation and layout (model and footprint)' (which is selected), 'Simulation only (model )', and 'Layout only (footprint)'. At the bottom right are two buttons: 'Next >' and 'Cancel'.

- Enter the component name, manufacturer and type.
- Select the component simulation and layout footprint requirements for the component.
- To use the component for simulation and PCB layout, select **I will use this component for both simulation and layout (model and footprint)**.

- To use the component only for simulation select **Simulation only (model)**.
- To use the component only for PCB layout, select **Layout only (footprint)**.

**Note** The number of Steps you will have to complete will depend upon which component requirements you select in Step 1. For example, if your component only requires a simulator model, the wizard will skip Step 6 on pin and model mapping.

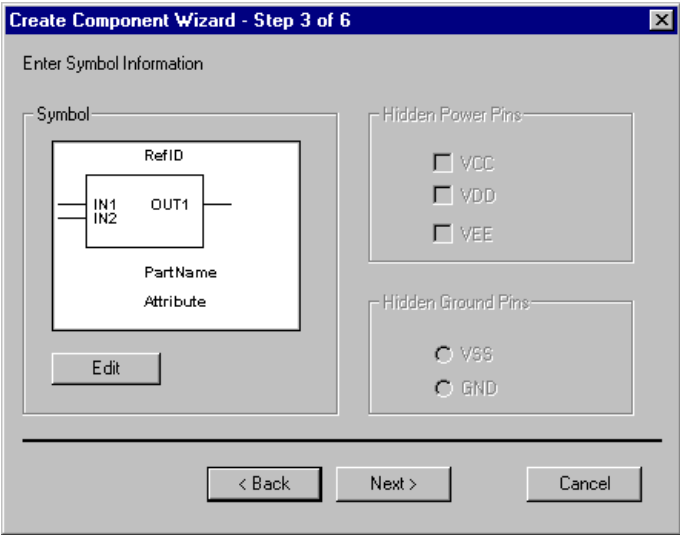
2. Click **Next**. Step 2 of the wizard appears.



- Specify the package type and pin number information for the component. The footprint information you enter here will appear in the Footprint tab of the Component Properties screen.



3. Click **Next**. Step 3 of the wizard appears.

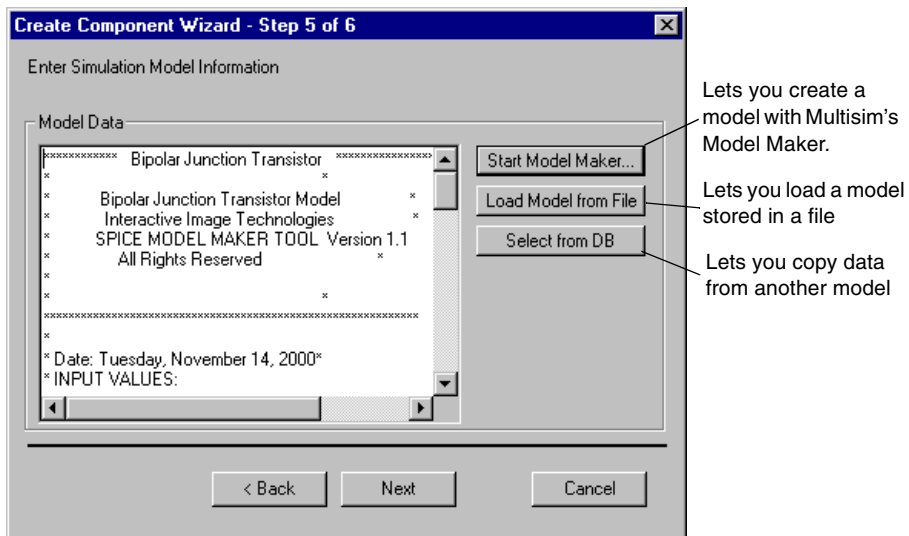


- The symbol you assign to this component will appear when it's placed on the circuit window.
  - To modify the symbol information, click **Edit** and a symbol editing program will be launched — see “5.9.2.3 Working with the Symbol Editor” on page 5-21 for more details.
4. To accept the symbol information displayed, click **Next**. Step 4 of the wizard appears.



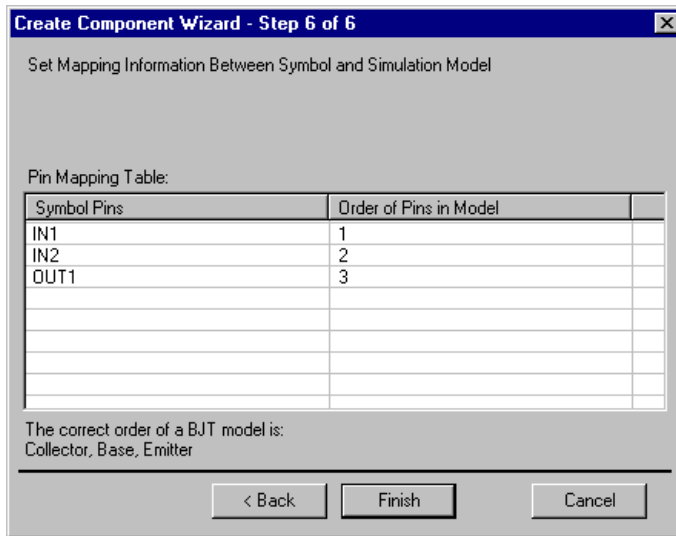
The logical and footprint pin mapping is needed for exporting to a layout package. The relationship of logical and footprint pins must follow the syntax in the databook in order to work in Multisim — see “5.9.2.6 Pins” on page 5-22 for more details.

- For each logical pin, enter a corresponding footprint pin. The mapping information you enter here will be displayed in the Footprint tab of the Component Properties screen.
  - To add logical and footprint pins, click **Add**. A logical pin will be added which you can then edit directly on this screen.
  - To delete logical and footprint pins, select the pin and click **Delete**.
5. Click **Next**. Step 5 of the wizard appears only if you indicated in Step 1 that this component requires a simulation model.



- Select a model from the Multisim database or from your own files, or create a model using a Model Maker.
- For more information on model maker see “5.10 Creating or Editing a Component Model” on page 5-26.
- For more information on loading a model see “5.10.2 Loading an Existing Model” on page 5-29.
- For more information on copying data from another model see “5.10.1 Copying a Component’s Model” on page 5-28.

6. Click **Next**. Step 6 of the wizard appears. If you indicated in Step 1 that this component does not require a layout footprint, the wizard will skip this last step.



- For each symbol pin enter a corresponding number to connect to its' respective nodes in the model. The mapping information you enter here will be displayed in the Model tab of the Component Properties screen. When you are done, click **Finish**.

The completed Component Properties screen appears, allowing you to review the component information.

➤ To save the new component:

1. Click **Save**. You are prompted to select a database and family destination for the new component.
2. Select a component family and database and click **OK**. Your new component has been saved.

➤ To edit the component, please see the subsequent sections of this chapter which give detailed information on the tabs of the Component Properties screen:

- the General tab — see “5.4 Editing a Component, General Properties” on page 5-8
- the Symbol tab — see “5.9 Editing and Creating a Component Symbol” on page 5-17
- the Model tab — see “5.10 Creating or Editing a Component Model” on page 5-26
- the Footprint tab — see “5.11 Creating and Editing Component Footprints” on page 5-30
- Electronic Parameters tab — see “5.5 Editing a Component, Electronic Parameters” on page 5-9

## 5.8 Removing Components

- To delete an existing component:

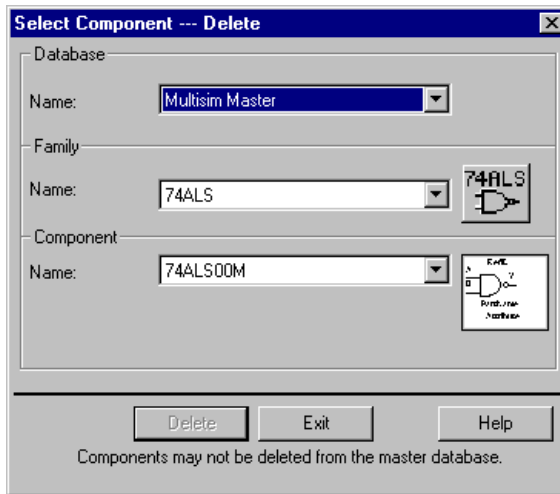
1. Click the Component editing button on the Design Bar and choose **Delete Component** from the pop-up menu that appears.



or

Choose **Tools/Delete Component**.

The Delete Component screen appears:



2. From the list, choose the database containing the component you want to remove (“Corporate Library” and “User” only). You cannot delete a component from the Multisim Master database.
3. From the **Family** list, choose the component family containing the component you want to remove.
4. From the **Component** list, choose the component you want to remove.
5. To continue, click **Delete**. You are prompted to confirm the deletion. To cancel, click **Exit**.

## 5.9 Copying Components

- To copy an existing component:

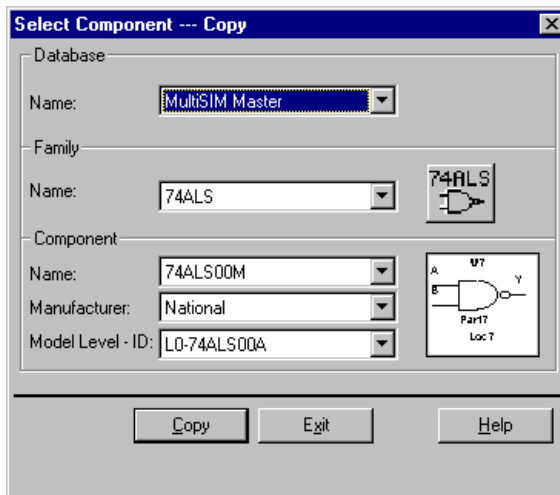


1. Click the Component editing button on the Design Bar and choose **Copy Component** from the pop-up menu that appears.

or

Choose **Tools/Copy Component**.

The Copy Component screen appears:

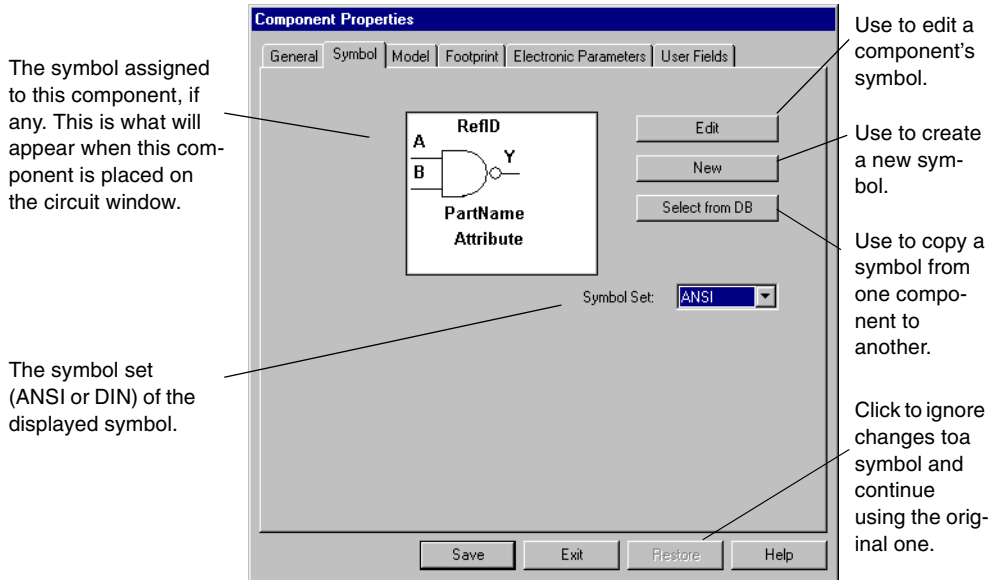


2. Using the drop-down lists, identify the component you want to copy and click **Copy**. You are prompted to select a destination.
3. Select the component database and component family that you want to copy the component to and click **OK**. A confirmation screen appears, assuring you that the component has been copied.
4. Click **OK** to return to the Copy Component screen.
5. When you are finished copying components, click **Exit**.

## 5.10 Editing and Creating a Component Symbol

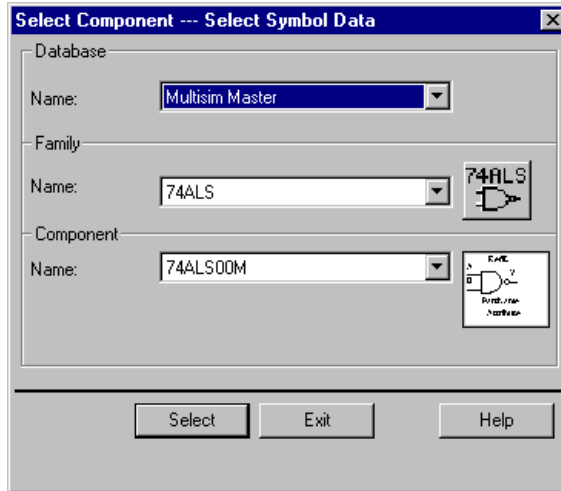
The Symbol tab of the Component Properties screen allows you to:

- edit a component's symbol
- give a component the same symbol as another component
- create a symbol for a component.



## 5.10.1 Copying a Component's Symbol

- To copy a symbol from another component:
  1. From the Component Properties screen, click **Select From DB** (database). The Select Symbol Data screen appears.



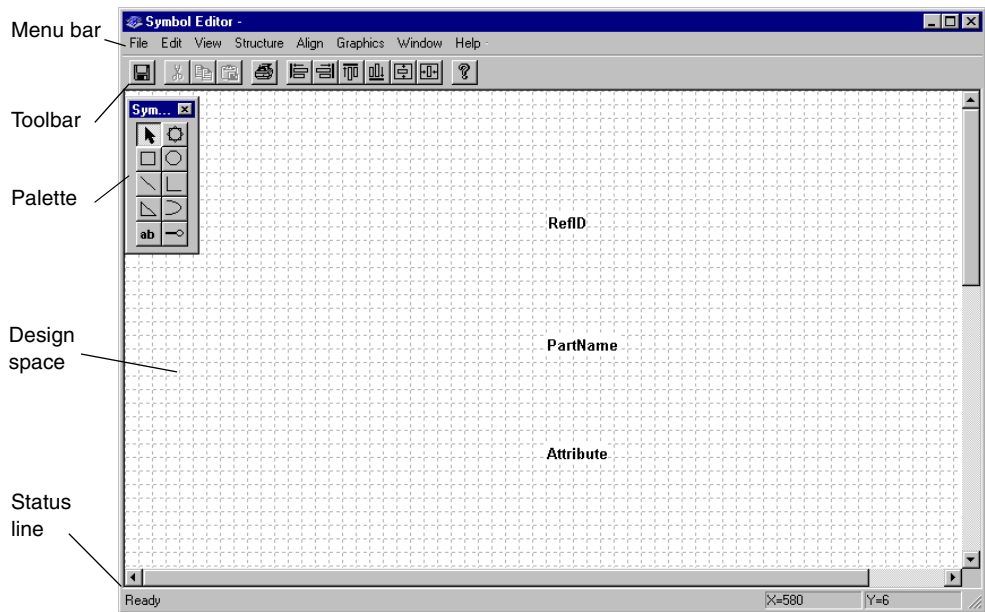
2. Use the drop-down lists to identify the component whose symbol you want to copy and click **Select**. (Click **Exit** to cancel.) You return to the Component Properties screen, where the symbol associated with the selected component appears.
3. If you want to change the symbol set to be associated with this component, select ANSI or DIN from the **Symbol Set** drop-down list. The appropriate symbol appears in the upper part of the screen.
4. To confirm the association of this symbol with your component, click **Save**.

If desired, you can also edit the copied symbol using the Symbol Editor, as described in the following sections.

## 5.10.2 Creating and Editing a Component's Symbol with the Symbol Editor

- To edit a component symbol:
  1. Under the Symbol tab of the Component Properties screen, from the **Symbol Set** drop-down list, be sure you have selected the desired symbol set (ANSI or DIN) whose symbol you would like to edit. The appropriate symbol appears in the upper part of the screen.
  2. Click **Edit**. The Symbol Editor appears, displaying the selected symbol for you to edit.
- To create a new symbol for the component, under the Symbol tab of the Component Properties screen, click **New**. The Symbol Editor appears with a blank screen and the three component labels (these are described in “5.9.2.4 Labels” on page 5-22).

The Symbol Editor looks like this:



The Symbol Editor screen consists of:

- the **menu bar**, which contains the menus with their associated commands
- the **toolbar**, which gives quick access to some commonly used tools.
- the **design space**, which is where you build or modify your symbols.
- the **palette**, which provides quick access to the most common operations in the Symbol Editor.



- the **status line**, which gives information on the currently selected object or action.
- The next sections describe the menus and palette in more detail.

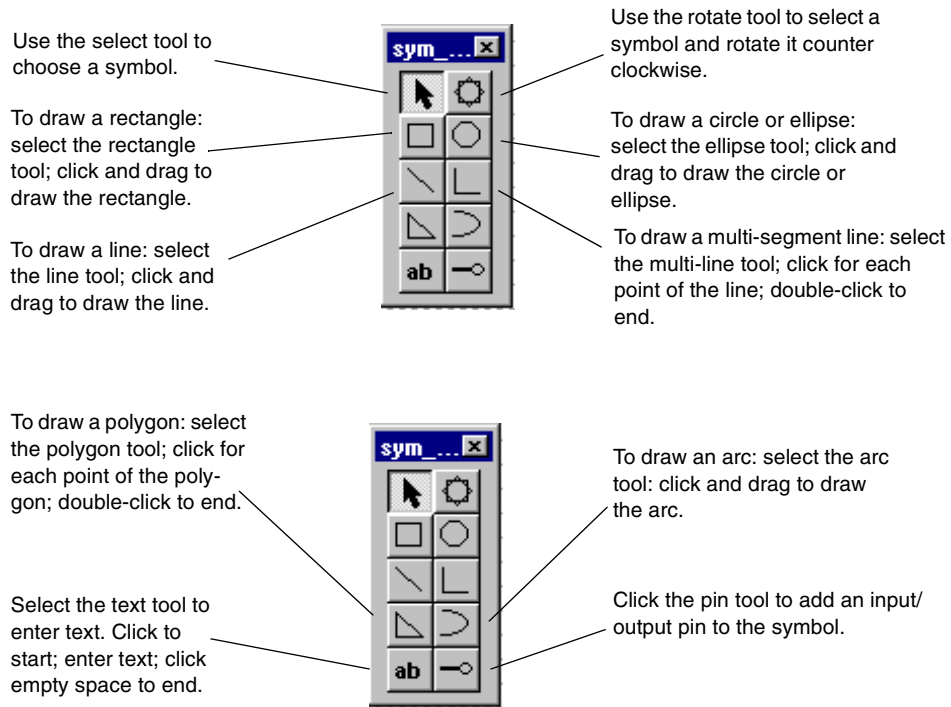
### 5.10.2.1 Symbol Editor Menus

Menu	Use
File	Use the File menu to create a symbol for a component or open an existing file containing a symbol. This menu also lets you preview, save or print the symbol. You also use this menu to exit the Symbol Editor.
Edit	Use the Edit menu to make changes to a symbol. You can choose to cut, copy, paste, or delete selected text or graphics displayed in the Symbol Editor screen. This menu also contains commands used to flip or rotate the displayed graphic.
View	Use the View menu to show/hide the following screen elements: toolbar, status bar, palette, grid and page boundaries. It also contains commands that let you change the grid and the magnification of the symbol.
Structure	Use the Structure menu to group together selections and position them in front of or behind each other.
Align	Use the Align menu to change the position of the selections in the window in relation to each other or to the grid.
Graphics	Use the Graphics menu to change the characteristics (color, font, pattern, pen style, or arrowheads) used for graphics and their accompanying text labels.
Window	Use the Window menu to move among the different open symbol files in the Symbol Editor.

### 5.10.2.2 Symbol Editor Palette

The Symbol Editor palette gives you quick access to the most common operations in the Symbol Editor.

The following illustrations give information on the tools and their functions :



### 5.10.2.3 Working with the Symbol Editor

The Symbol Editor is essentially a graphics editor with the usual range of tasks (placing graphics, changing, their color, size and fill, and so on), along with special additions.

To use the Symbol Editor to create working symbols, however, you need to be familiar with the elements required to make up a symbol in Multisim. The three key elements needed for a symbol are:

- labels
- shape
- pins

These are described in more detail in the following sections.

### 5.10.2.4 Labels



Each symbol has three labels, variables that are replaced by values from the component's model. This allows the same symbol to be used for many different components in a family. The variables are:

- the component's reference ID (represented by the string "RefID")
- the component's value or part name (represented by the string "PartName")
- the component's attribute (represented by the string "Attribute").

All symbols have these variables; if you choose to create a new symbol in the Symbol Editor, these three variables are supplied automatically (although you control where they are located with respect to the shape and how they are presented, for example, their color and font).

Information for the component's reference ID, identified as "RefID", and component "Part-Name" are extracted from the components database and automatically entered by Multisim. For example, "R1" indicates that the component is the first resistor placed on the circuit window. The "R" is extracted from the component database and the "1" is a sequential number placed on the component. Any additional resistors either placed on the circuit window using the Parts Bin or copied will increase sequentially, i.e. R1, R2, R3.

### 5.10.2.5 Shape

A symbol requires a shape to allow users to recognize its general function. For example, a capacitor has a shape of  and a nor gate has a shape of . You utilize the drawing capabilities of Multisim's Symbol Editor to construct a shape that makes logical sense for the component you are creating or modifying. The simplest way to do this is to edit the shape of an existing component. Once this is done, you will need to add pins.

### 5.10.2.6 Pins

There are three main parts to a pin: the "logical pin", the "physical pin", and the shape of the pin.









The "logical pin" is the name given to an actual pin. For example, most digital parts have pins named "Vc" and "GND". These names are the "logical pin" names used to identify the actual pin. The logical pin name can be anything you want as long as it is understood. You can use the acronym "GND" or call the pin "Ground". It is recommended, however, that you use the logical pin names provided in the data book for a component.

The "physical pin" is the physical location of the pin on a given component. For example, if you are creating a symbol that has 16 physical pins attached to it, then you would have physical pins numbered from 1 to 16. The numbering of the physical pins is what is used by PCB layout software to ensure that connections from one component to another are made properly. While it is recommended that you use the logical pin names given in a databook for a compo-

nent, for the physical pin names you *must* use the names from the databook or your component will not work properly.

**Note** The relationship of logical and physical pins must follow the syntax in the databook in order to work in Multisim.

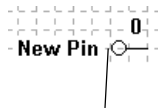
The third part of a pin is its appearance or shape. Multisim provides eight pin shapes that you can use:

- dot  Negative Active Signal
- dot-clock  Negative Active Clock
- line  Positive Active Signal
- short  Positive Active Signal - short format
- zero length  Terminal Pin
- clock  Positive Active Clock
- input wedge  Negative Active Input Signal (DIN symbols)
- output wedge  Negative Active Output Signal (DIN symbols)

None of the pin shapes have any impact on the operation of the component. However, pin shapes do have an impact on the component user; each of the shapes is used for specific identification purposes. Again, it is recommended that you follow the pin shape from the data book when creating a symbol for a component.

Pins must be connected to the symbol shape correctly in order for the symbol to function properly. Follow these rules:

- A pin must always be facing out with no shapes, lines or text blocking its way to be connected by a wire. If a pin is blocked, you may not be able to connect it.
- A pin can only be connected at the far end of the logical side:



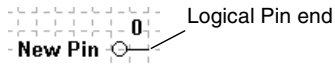
The side marked "New Pin" is the logical end of the pin. Connect the pin at the far end of the logical side.

➤ To add a pin to a symbol:



1. Click the **Pin** button on the palette. A cursor appears.

**Note** Click on the screen where you want the pin to appear. A "new pin" placeholder appears: When placing a pin to a shape, always use the side containing the logical pin



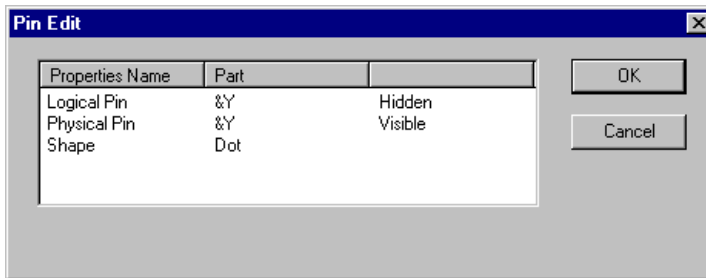
name. Multisim will not recognize a pin that has been connected backwards.

2. If the pin does not touch the shape as you want it to, either click and drag the pin to the desired location or draw a line from the logical pin end to the component shape.

Once you have placed the pin in its location, you can then edit its properties.

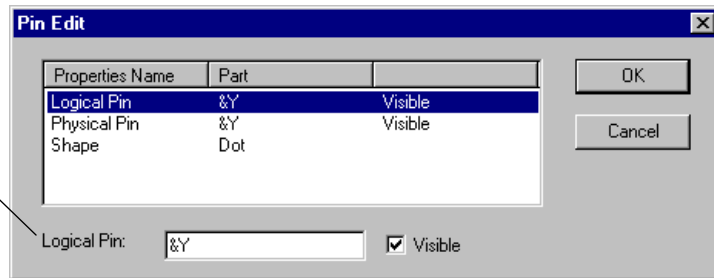
- To edit a pin's properties:

1. Double-click the pin. The Pin Edit screen appears:



2. Click on the pin attribute (Logical Pin, Physical Pin or Shape) you want to edit. A field appears at the bottom of the screen, allowing you to enter the value of the Logical pin and change its shape. For example:

This field lets you set the value for the selected property.



3. Enable or disable **Visible** to make the value visible or hidden by default.

**Note** It is recommended that you use databook names for logical pins. It is important that you use the databook numbering for the physical pins.

4. To cancel your changes, click **Cancel**. To confirm them, click **OK**.

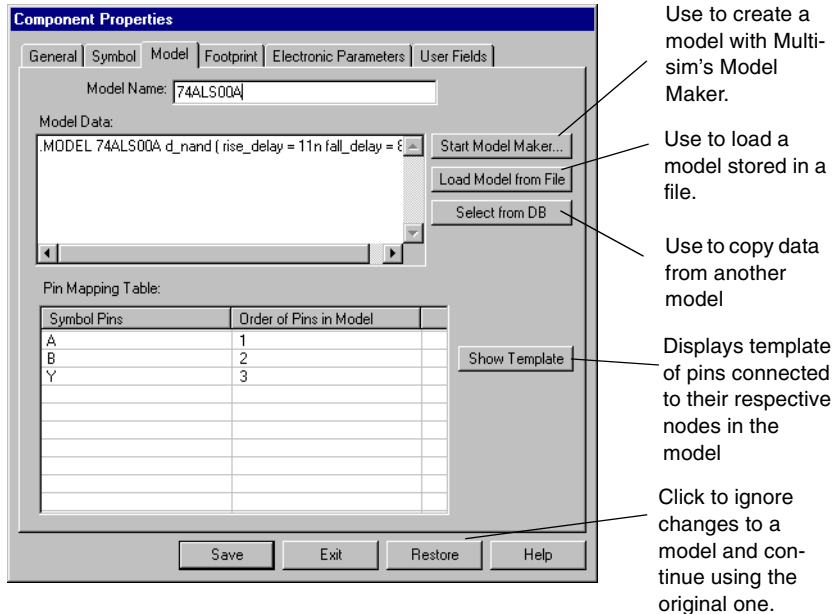
## 5.11 Creating or Editing a Component Model

A component that has an effect on the circuit *must* have a model if you wish to simulate that component. The only components in Multisim that do not have models associated with them are connectors and junctions. These components do not affect the simulation of the circuit.

The Model tab of the Component Properties screen offers you a number of options to choose from to assign a model to your component. You can:

- modify the model information of a component
- load an existing model for a component

- copy a model of one component to another
- create a model for a component



First of all, you can directly modify the existing model data by using the fields of this screen, or you can enter information from scratch in these fields. The **Model Data** and Symbol to Model Pin Mapping fields contain the information that make up the model itself (and are thus the most important part for simulation purposes). The **Model Data** field contains the model's code (for example in SPICE) and the **Pin Mapping table** field lists the pins and their corresponding mapping order. See "5.10 Creating or Editing a Component Model" on page 5-26 for more information on editing model data.

**Note** If you modify a model, the model changes for all other components within the same database, whose model is based on that template. If you want to modify a model for a specific component, save the modified model with a name similar to that of the component.

Alternatively, you can copy a model whose model template matches what you want. More information on copying models is given in "5.10.1 Copying a Component's Model" on page 5-28.

Thirdly, you can import or load an existing model by clicking **Load Model from File**. These options are described in more detail in "5.10.2 Loading an Existing Model" on page 5-29.

Finally, depending on your edition of Multisim, you can create your own model using the Model Maker command. .

**Note** If you choose to edit a model's data or template directly, be very careful when entering information. Making a typing error or removing a character by mistake could cause the model to function improperly. Unless you are experienced at creating/editing models, it is recommended that you copy a model that has the same Template information you require.

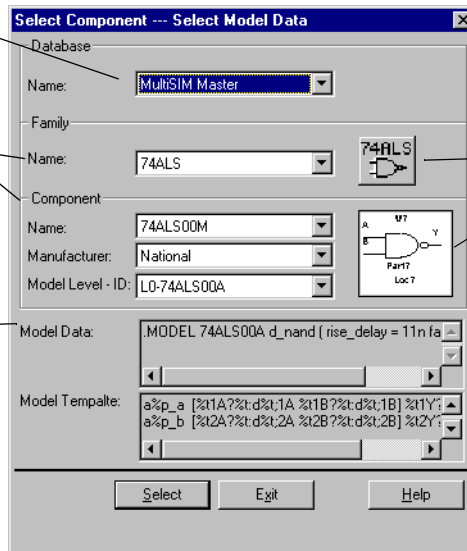
## 5.11.1 Copying a Component's Model

- To copy the model information from an existing component:
  1. Click **Select from DB**. The Select Model Data screen appears.

Select the database you want to choose a model from.

Select the component family, name, manufacturer, model level that you want from the drop-down lists.

The model's data and template appear here.



The component's icon and symbol appear here.

2. Select from among the available databases in the **Name** drop-down list.
3. Using the **Family** and **Component** drop-down lists, choose the component whose model template most closely matches what you want.
4. Click **Select** to return to the Component Properties screen.



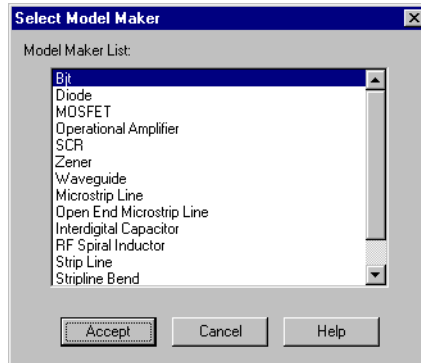
## 5.11.2 Loading an Existing Model



The models for a component can come from a variety of sources. Please note that some of these options may not be available in your edition of Multisim.

- To load or import a model created by Multisim's analog or digital model maker:

1. Click **Start Model Maker**. The Select Model Maker screen appears:



2. Select the Model Maker you wish to use to make a model.
  3. Click **Accept** to continue to start the process of making a model. Click **Cancel** to return to the Model Tab of the Component Properties screen.
  4. For analog model makers, refer to subsequent sections of this chapter for procedures on using specific Model Makers. For RF model makers, see Chapter 14, "RF".
  5. When you have entered in all the required information in the Model Maker screens, click **OK**. The data for the model you have just created will appear in the Model tab fields.
- To load or import an existing VHDL, Verilog HDL, Code Model, or SPICE model for your component:

1. Click **Load Model From File**. A standard file browser appears, allowing you to choose the file type, name and location of the files to be loaded. Before loading a netlist, make sure you know what folder it is in. Most Bspice, Xspice and Pspice netlists end in extensions .cir and .net.
2. Select the file to load and click **OK**. The model data appears in the Model tab fields.

**Note** PSpice is not an industry standard, but is proprietary to the Orcad SPICE simulation tool. Since some component vendors make models for their components available in PSpice format, Multisim has been designed to support PSpice models as extensively as possible. However, you will not be able to share models or circuits with other SPICE users or tools.

## 5.12 Creating and Editing Component Footprints

The Footprint tab of the Component Properties screen allows you to specify/modify package information for a component and map physical and logical pins of a component. The logical and physical pin mapping is needed for exporting to a layout package; the pin group and type information is needed for simulation.

**Component Properties**

General | Symbol | Model | **Footprint** | Electronic Parameters | User Fields

Package Type: M14A

Number of Pins: 14

Number of Sections: 4

Component Technology: 74ALS

Symbol to Footprint Pin Mapping Table:

Logical Pins	Footprint Pins	Section	Type
GND	7	GROUND	GND
3Y	8	C	ACTIVE DRIVER
3A	9	C	INPUT
3B	10	C	INPUT
4Y	11	D	ACTIVE DRIVER
4A	12	D	INPUT
4B	13	D	INPUT
VCC	14	POWER_1	VCC

Enter physical and package information so that your part can be properly exported to PCB layout. The pin mapping table is used to match each pin on your symbol to a physical pin location. The section column is used to identify to which group a pin applies in a multi-sections. The type column identifies the pin's function. A simulation only component always use 'NONE' as its package type.

Save Exit Restore Help

- To modify or enter package information:
  1. In the **Package Type** field, modify or enter the package type (for example, DIP14).
  2. In the **Number of Pins** field, modify or enter the number of pins.
  3. In the **Number of Sections** field, modify or enter the number of sections.
  4. In the **Component Type** field, choose the component type from the drop-down list.
  5. In the **Component Technology** field, choose the component technology from the drop-down list.
- To map logical and physical pins:
  1. Click on the field you want to modify, until a frame appears around the field.
  2. For each logical pin, enter its corresponding physical pin on the package.
  3. For each pin, enter the grouping of pins, using the syntax described in “5.11.1 Pin Group Naming Convention” on page 5-31.

4. For each pin, enter the type, using the syntax described in “5.11.2 Pin Type Naming Convention” on page 5-31.
5. To save your changes, click **OK**. To cancel them, click **Cancel**.

## 5.12.1 Pin Group Naming Convention

For logical pins, use the following formats:

For:	Use:	Where:
pins associated with one section of a component	GRP: <i>n</i>	<i>n</i> is the section
pins common to several sections, but not all sections	GRP: <i>n:m</i>	<i>n</i> and <i>m</i> are the sections
pins common to all sections	COM	
pins associated with voltages	PWR:V0 or PWR: <i>Vn</i>	V0 is ground or <i>n</i> is a voltage
unused pins (no connects)	NC	

## 5.12.2 Pin Type Naming Convention

For digital components, the pin type is used to link together the I/O models to the logical core for each device. In other families, such as analog components where the simulation models are self-contained units, pin types are for information purposes only.

Use the format:

TYPE: MODE: MODEL

where

Type	is either A (analog) or D (digital)
Mode	is one of the following: input output I/O 3-state Open_drain Open_source Open_sink I/O_open_drain I/O_open_source I/O_open_sink Input_ECL Output_ECL I/O_ECL Terminator Power NC
Model	pin model name (none for analog)

## 5.13 Creating a Component Model Using the Model Makers

Multisim offers several advanced Model Makers which automatically generate simulation models for you when you give them databook values as input. Using Model Makers will save you time and effort but do require practice for you to become proficient with them.

For each Model Maker, preset values are provided for a specific model. However, these are not default values, and you can select numerical values based on the component you are interested in, using data from a databook.

When working with databooks, note that different databooks provide parameters for a component model in different styles. While some pieces of information are given numerically in tables or lists for a specific operating point, others are given in the form of a chart or graph. Both types of information are required by Multisim's Model Makers. In the case of tables or lists, you will need to enter the operating point and the value that you want. In the case of charts or graphs, the way you select the points from the appropriate curves will have an impact on the accuracy of the parameters of the final model. We give suggestions on methods for selecting points in the procedures for each Model Maker. Also, note that the pieces of information provided by databooks are usually the same from one manufacturer to another, even though the names or labels and descriptions of parameters are different.

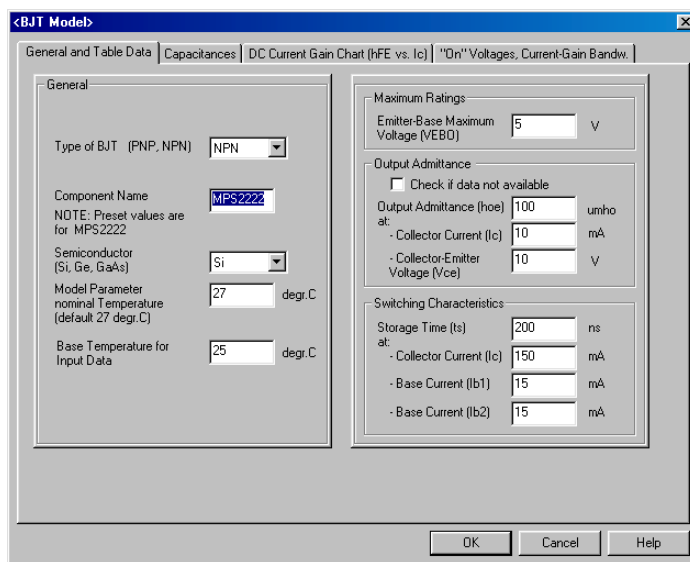
### 5.13.1 BJT Model Maker

1. From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
2. From the Model Maker list, select BJT and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The BJT Model screen appears.
3. Enter values on the BJT Model Maker screen as described in the following sections.
4. When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

**Note** The BJT Model screen shows preset values for the MPS2222A model.

## Entering General and Table Data

1. In the BJT Model Maker, click the General and Table Data tab:



2. Locate data information for the BJT model from a databook.

➤ To enter **General** data:

1. Enter the appropriate BJT type (NPN or PNP) in the **Type of BJT** field. This is usually found on the first page of the data book.
2. Enter the **Component Name**. This is usually found in the top right-hand corner of the datasheet.
3. In the **Semiconductor** field, enter the type of semiconductor. This is usually found written next to the component type.
4. If desired, change the default value set by Multisim for **Model parameter nominal Temperature** of 27 degrees.
5. If desired, change the default value for **Base Temperature for Input Data**. This is typically found in the top left corner of the "Electrical Characteristics" table in the databook.

- To enter **Maximum Ratings** data:

1. In the databook for the BJT, locate the “Maximum Ratings” table — for example:

Enter this information in the **Emitter-Base Maximum Voltage** field.

**MAXIMUM RATINGS**

Rating	Symbol	MPS2222	MPS2222A	Unit
Collector–Emitter Voltage	$V_{CEO}$	30	40	Vdc
Collector–Base Voltage	$V_{CBO}$	60	75	Vdc
Emitter–Base Voltage	$V_{EBO}$	5.0	6.0	Vdc
Collector Current — Continuous	$I_C$	600		mAdc
Total Device Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	625 5.0		mW mW/ $^\circ\text{C}$
Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	1.5 12		Watts mW/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	$T_J, T_{stg}$	–55 to +150		$^\circ\text{C}$

2. Find the value for Emitter-Base Voltage and enter the value in the **Emitter-Base Maximum Voltage (VEBO)** field.

- To enter **Output Admittance** data:

1. In the databook, locate the “Small Signal Characteristics” table, and find the values for Output Admittance — for example:

**SMALL-SIGNAL CHARACTERISTICS**

Current–Gain — Bandwidth Product <sup>(2)</sup> ( $I_C = 20\text{ mA}$ , $V_{CE} = 20\text{ Vdc}$ , $f = 100\text{ MHz}$ )	MPS2222 MPS2222A	$f_T$	250 300	— —	MHz
Output Capacitance ( $V_{CB} = 10\text{ Vdc}$ , $I_E = 0$ , $f = 1.0\text{ MHz}$ )		$C_{obo}$	—	8.0	pF
Input Capacitance ( $V_{EB} = 0.5\text{ Vdc}$ , $I_C = 0$ , $f = 1.0\text{ MHz}$ )	MPS2222 MPS2222A	$C_{ibo}$	— —	30 25	pF
Input Impedance ( $I_C = 1.0\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ ) ( $I_C = 10\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	MPS2222A MPS2222A	$h_{ie}$	2.0 0.25	8.0 1.25	k $\Omega$
Voltage Feedback Ratio ( $I_C = 1.0\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ ) ( $I_C = 10\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	MPS2222A MPS2222A	$h_{re}$	— —	8.0 4.0	$\times 10^{-4}$
Small-Signal Current Gain ( $I_C = 1.0\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ ) ( $I_C = 10\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	MPS2222A MPS2222A	$h_{fe}$	50 75	300 375	—
Output Admittance ( $I_C = 1.0\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ ) ( $I_C = 10\text{ mA}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	MPS2222A MPS2222A	$h_{oe}$	5.0 25	35 200	$\mu\text{mhos}$
Collector Base Time Constant ( $I_E = 20\text{ mA}$ , $V_{CB} = 20\text{ Vdc}$ , $f = 31.8\text{ MHz}$ )	MPS2222A	$\tau_b/\tau_C$	—	150	ps
Noise Figure ( $I_C = 100\text{ }\mu\text{A}$ , $V_{CE} = 10\text{ Vdc}$ , $R_S = 1.0\text{ k}\Omega$ , $f = 1.0\text{ kHz}$ )	MPS2222A	NF	—	4.0	dB

Use this information to enter data in the **Output Admittance** fields.

If data are not available, enable **Check if data not available**.

2. Based on the table data, enter:

- **Output Admittance (hoe)**
- **Collector Current (Ic)**
- **Collector-Emitter Voltage (Vce)**

**Note** Databooks provide maximum and minimum values for the Output Admittance parameter. Select a typical value of output admittance.

➤ To enter **Switching Characteristics** data:

1. In the databook, find the “Switching Characteristics” table — for example:

Use this information to enter data in the **Switching Characteristics** fields.

**SWITCHING CHARACTERISTICS MPS2222A only**

Delay Time	$(V_{CC} = 30 \text{ Vdc}, V_{BE(\text{off})} = -0.5 \text{ Vdc}, I_C = 150 \text{ mAdc}, I_{B1} = 15 \text{ mAdc})$ (Figure 1)	$t_d$	—	10	ns
Rise Time		$t_r$	—	25	ns
Storage Time	$(V_{CC} = 30 \text{ Vdc}, I_C = 150 \text{ mAdc}, I_{B1} = I_{B2} = 15 \text{ mAdc})$ (Figure 2)	$t_s$	—	225	ns
Fall Time		$t_f$	—	60	ns

1. Pulse Test: Pulse Width  $\leq 300 \mu\text{s}$ , Duty Cycle  $\leq 2.0\%$ .
2.  $t_f$  is defined as the frequency at which  $|h_{fe}|$  extrapolates to unity.

2. Based on the table data, enter:

- **Storage Time (ts)**
- **Collector Current (Ic)**
- **Base Current (Ib1)**
- **Base Current (Ib2)**

## Entering Capacitances Data

1. Click the Capacitances tab:

The screenshot shows the "BJT Model" dialog box with the "Capacitances" tab selected. The dialog is divided into two main sections: "Emitter-Base Capacitance (Input Capacitance)" and "Collector-Base Capacitance Chart (Output Capacitance)".

**Emitter-Base Capacitance (Input Capacitance):**

- Capacitance (Ceb1) at: 24 pF
- Lowest Value of Reverse Voltage: 0.1 V
- Intermediate Values:
  - Capacitance (Ceb2) at: 22 pF, Reverse Voltage: 0.3 V
  - Capacitance (Ceb3) at: 20 pF, Reverse Voltage: 0.5 V
  - Capacitance (Ceb4) at: 14 pF, Highest Value of Reverse Voltage: 5 V

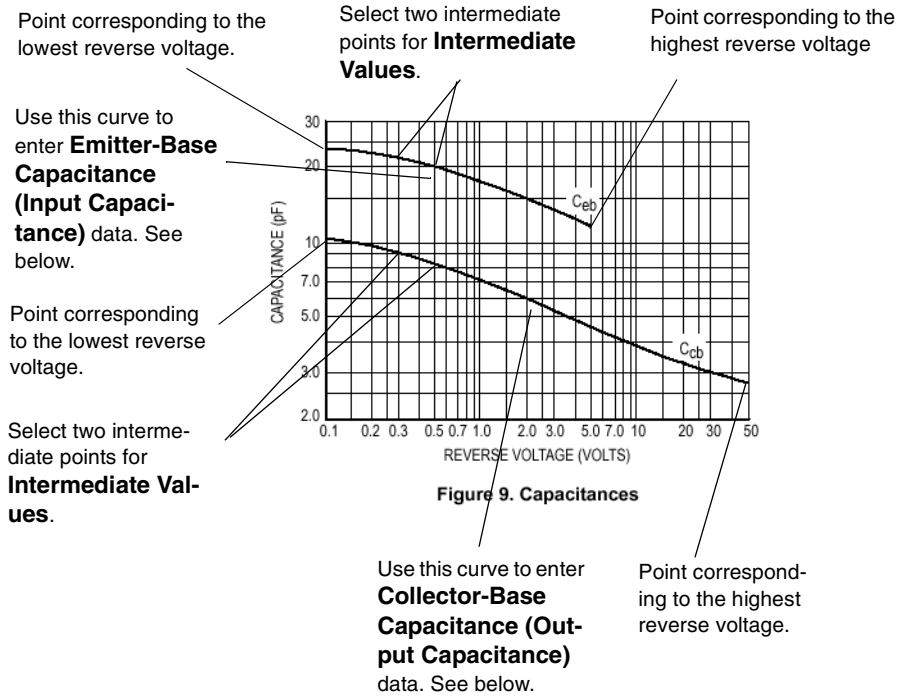
**Collector-Base Capacitance Chart (Output Capacitance):**

- Capacitance (Ccb1) at: 11 pF
- Lowest Value of Reverse Voltage: 0.1 V
- Intermediate Values:
  - Capacitance (Ccb2) at: 9 pF, Reverse Voltage: 0.3 V
  - Capacitance (Ccb3) at: 8 pF, Reverse Voltage: 0.6 V
  - Capacitance (Ccb4) at: 2.5 pF, Highest Value of Reverse Voltage: 50 V

At the bottom of the dialog are buttons for "OK", "Cancel", and "Help".



2. In the databook, locate the “Ceb and Ccb vs. Reverse Voltages (RV)” graph — for example:



- To enter **Emitter-Base Capacitance (Input Capacitance)** data:
  1. On the Ceb curve, locate the point corresponding to the lowest voltage, or the beginning point, of the Ceb curve. Use the coordinates of this point to enter values for:
    - **Capacitance (Ceb1)**
    - **Lowest Value of Reverse Voltage**
  2. On the same curve, locate the point corresponding to the maximum voltage, or the end point. Use the coordinates of this point to enter values for:
    - **Capacitance (Ceb4)**
    - **Highest Value of Reverse Voltage**
  3. To enter **Intermediate Values**, select two intermediate points close to the left side in the low voltage region. Ensure that they are not too close, to avoid excessive error in the model. Use the coordinates of the first and second points to enter values for:
    - **Capacitance (Ceb2) at Reverse Voltage**
    - **Capacitance (Ceb3) at Reverse Voltage**

- To enter **Collector-Base Capacitance Chart (Output Capacitance)** data:
- Using the Ccb curve from the same “Ceb and Ccb vs. Reverse Voltages (RV)” graph, repeat steps 1 through 3 above to enter values for:
    - **Capacitance (Ccb1)**
    - **Lowest Value of Reverse Voltage**
    - **Capacitance (Ccb2) at Reverse Voltage**
    - **Capacitance (Ccb3) at Reverse Voltage**
    - **Capacitance (Ccb4)**
    - **Highest Value of Reverse Voltage**

## Entering DC Current Gain Chart data

- Click the DC Current Gain Chart (hFE vs. Ic) tab:

The screenshot shows the "BJT Model" dialog box with the "DC Current Gain Chart (hFE vs. Ic)" tab selected. The dialog is divided into two main sections: "DC Current Gain (hFE) at base Temperature" and "DC Current Gain (hFE) at another Temperature".

**DC Current Gain (hFE) at base Temperature:**

- DC Current Gain (hFE1) at:  (Minimal Collector Current:  mA)
- DC Current Gain (hFE2) at:  (Intermediate Collector Current (low values range):  mA)
- Max Value of DC Current Gain (hFE\_Max):
- Collector Current (IL) at 0.5 Max DC Current Gain (low values range):  mA
- Collector Current (Ik) at 0.5 Max DC Current Gain (high values range):  mA

**DC Current Gain (hFE) at another Temperature:**

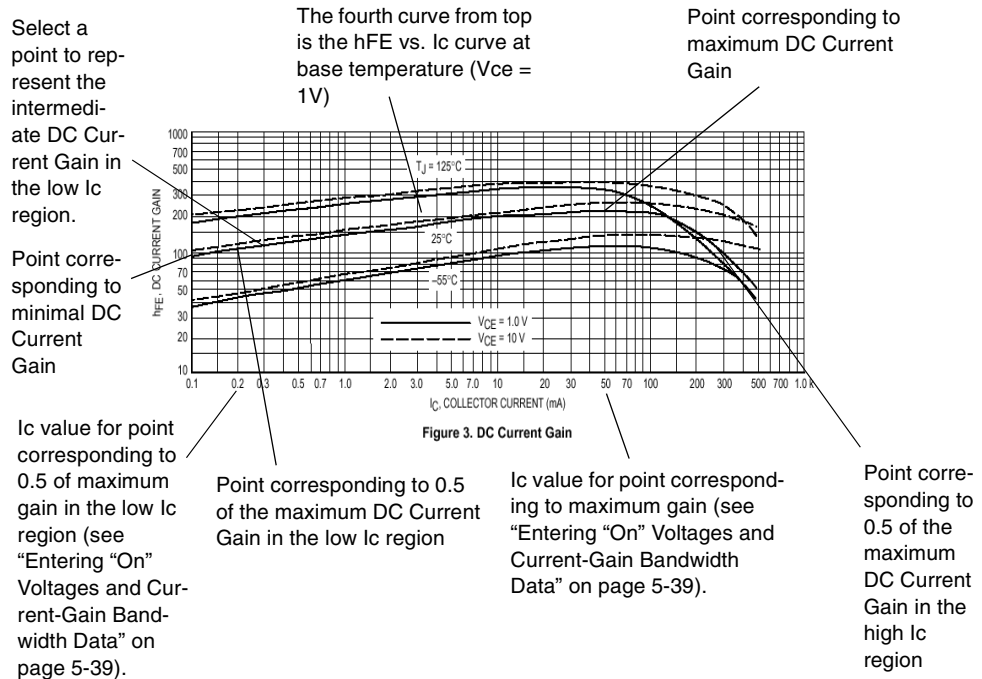
- Another Temperature on the Chart (t2):  deg.C
- DC Current Gain (hFE1\_t2) at:  (Minimal Collector Current:  mA)
- DC Current Gain (hFE2\_t2) at:  (Intermediate Collector Current (low values range):  mA)
- Max Value of DC Current Gain (hFE\_Max\_t2):
- Collector Current (IL\_t2) at 0.5 Max DC Current Gain (low values range):  mA

Buttons at the bottom: OK, Cancel, Help.

- In the databook for the BJT, locate the hFE vs. Ic graph.

➤ To enter **DC Current Gain (hFE) at base Temperature** data:

1. Among the hFE vs.  $I_C$  curves at the base temperature for the BJT, select the one whose  $V_{CE}$  is most likely the operating point for the transistor. For example:



**Note** You will need to select a curve with the same voltage as the  $I_C$ - $V_{BE}$  curve you will use to enter data on the last tab of this screen. See “Entering “On” Voltages and Current-Gain Bandwidth Data” on page 5-39.

2. Find the point on the curve corresponding to the minimal collector current, or the beginning point of the curve. Use the coordinates of this value to enter:
  - **DC Current Gain (hFE1)**
  - **Minimal Collector Current**
3. Select a point from the low  $I_C$  region of the same curve. Use the coordinates of this point to enter:
  - **DC Current Gain (hFE2)**
  - **Intermediate Collector Current (low values range)**
4. Find the highest point on the curve, and enter its DC Current Gain value in the **Max Value of DC Current Gain (hFE\_Max)** field.

**Note** You will need to note the  $I_c$  value of this point to plot points on the  $I_c$ - $V_{be}$  curve you will use to enter data on the last tab of this screen. See “Entering “On” Voltages and Current-Gain Bandwidth Data” on page 5-39.

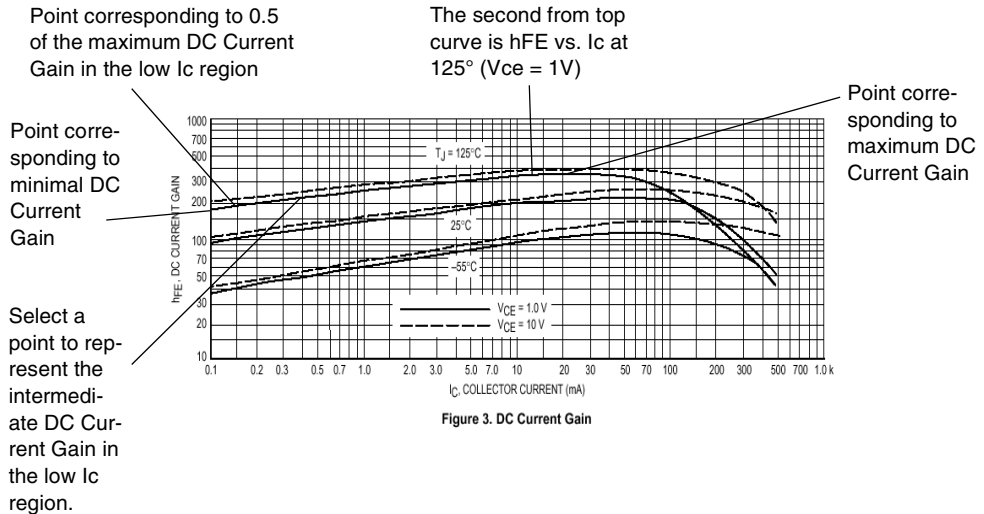
- Find the two points corresponding to 0.5 of the maximum DC current gain value, one in the low  $I_c$  region and one in the high  $I_c$  region. Use these points to enter:

- **Collector Current ( $I_L$ ) at 0.5 Max DC Current Gain (low values range)**
- **Collector Current ( $I_{kf}$ ) at 0.5 Max DC Current Gain (high values range)**

**Note** You will need to note the  $I_c$  value of the point in the low  $I_c$  region to plot points on the  $I_c$ - $V_{be}$  curve you will use to enter data on the last tab of this screen. See “Entering “On” Voltages and Current-Gain Bandwidth Data” on page 5-39.

➤ To enter **DC Current Gain ( $h_{FE}$ ) at another Temperature** data:

- Using the  $h_{FE}$  vs.  $I_c$  graph, find a curve at a different temperature from the base temperature. (This can be any other temperature.) For example:



- Enter the temperature of the selected curve in the **Another temperature on the Chart ( $t_2$ )** field.
- Find the point on the curve corresponding to the minimal collector current, or the beginning point of the curve. Use the coordinates of this value to enter:
  - **DC Current Gain ( $h_{FE1\_t2}$ )**
  - **Minimal Collector Current**

4. Select a point from the low  $I_c$  region of the same curve. Use the coordinates of this point to enter:
  - **DC Current Gain ( $hFE2\_t2$ )**
  - **Intermediate Collector Current (low values range)**
5. Find the highest point on the curve, and enter its DC Current Gain value in the **Max Value of DC Current Gain ( $hFE\_Max2$ )** field.
6. Find a point corresponding to 0.5 of the maximum DC current gain value in the low  $I_c$  region and enter its value in the **Collector Current ( $I_{L\_t2}$ ) at 0.5 Max DC Current Gain (low values range)** field.

## Entering “On” Voltages and Current-Gain Bandwidth Data

1. Click the “On” Voltages, Current-Gain Bandw. tab:

The screenshot shows the "BJT Model" dialog box with the "On" Voltages, Current-Gain Bandw. tab selected. The dialog is divided into several sections with input fields and units.

"On" Voltages Chart		Vbe(sat)-Ic	
Collector-Emitter Voltage for Vbe vs. Ic (same as hFE curve)	0 V	Saturation Base-Emitter Voltage (Vbe1_sat) @ Ic/Ib=10 and at:	0.83 V
"On" Base-Emitter Voltage (Vbe1) at the Lowest Value of Collector Current	0.57 V / 0.1 mA	Collector Current in the high values range	100 mA
"On" Base-Emitter Voltage (Vbe_L) at 0.5 Max Gain Collector Current (low values range)	0.58 V	Saturation Base-Emitter Voltage (Vbe2_sat) at the Highest Value of Collector Current	0.95 V / 500 mA
"On" Base-Emitter Voltage (Vbe_hFEMax) at Max Gain	0.915 V		

Vce(sat)-Ic		Current-Gain Bandwidth Product Chart (fT)		Temperature Coefficients Chart	
Saturation Collector-Emitter Voltage (Vce1_sat) @ Ic/Ib=10 and at:	0.1 V	Maximum Value of Current-Gain Bandwidth Product	250 MHz	Lowest Value of Base-Emitter Voltage Temperature Coefficient	-2.2 mV/deg. C
Collector Current in the high values range	180 mA				
Saturation Collector-Emitter Voltage (Vce2_sat) @ Ic/Ib=10 and at the Highest Value of Collector Current	0.21 V / 500 mA				

Buttons: OK, Cancel, Help

2. In the databook, locate the  $I_c$  vs.  $V_{be}$  graph.

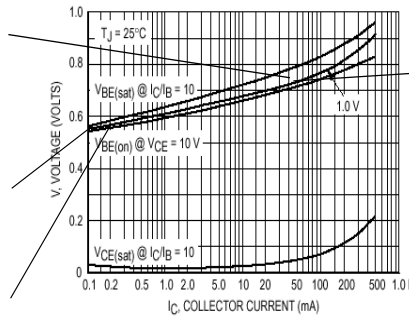
➤ To enter **“On” Voltages Chart** data:

1. On the graph, locate the curve with the same  $V_{ce}$  as the one used in the hFE data. Enter its  $V_{ce}$  value in the **Collector-Emitter Voltage for Vbe vs. Ic (same as hFE curve)** field. For example:

The second from top curve is the  $V_{be}$  vs.  $I_c$  with same  $V_{ce}$  as  $I_c$ -hFE curve. (See “Entering DC Current Gain Chart data” on page 5-36.)

Point corresponding to minimal  $V_{be}$

Point corresponding to 0.5 of the maximum DC Current gain. (See “Entering DC Current Gain Chart data” on page 5-36.)



Point corresponding to maximum DC Current gain. (See “Entering DC Current Gain Chart data” on page 5-36.)

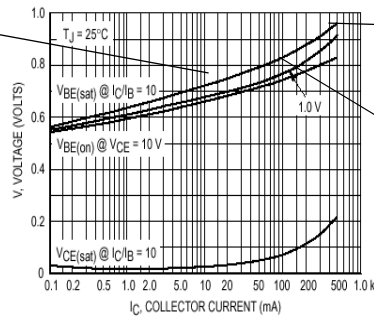
Figure 11. “On” Voltages

2. Find the point on this curve corresponding to the minimal  $I_c$  value, or the beginning point of the curve. Use the coordinates of this point to enter:
  - **“On” Base-Emitter Voltage (Vbe1)**
  - **Lowest Value of Collector Current**
3. Using the  $I_c$  vs. hFE graph from the previous section, locate the  $I_c$ -hFE curve at the base temperature that was used to enter data on the third tab of this screen. At the point of the maximum DC Current Gain (hFE), note the coordinate for the collector current ( $I_c$ ).
4. On the  $I_c$ - $V_{be}$  graph, find the point corresponding to this coordinate for  $I_c$  on the curve used in steps 1 to 4. Enter the voltage for this point in the **“On” Base-Emitter Voltage (Vbe\_hFEMax) at Max Gain** field.
5. Using the  $I_c$  vs. hFE graph from the previous section, locate the  $I_c$ -hFE curve at the base temperature that was used to enter data on the third tab of this screen. At the point corresponding to 0.5 of the maximum DC Current Gain (hFE), note the coordinate for the collector current ( $I_c$ ).
6. On the  $I_c$ - $V_{be}$  graph, find the point corresponding to this coordinate for  $I_c$  on the curve used in steps 1 to 4. Enter the voltage for this point in the **“On” Base-Emitter Voltage (Vbe\_iL) at 0.5 Max Gain Collector Current (low values range)** field.

➤ To enter **Vbe(sat)-Ic** data:

1. Using the  $I_c$  vs.  $V_{be}$  graph, locate the curve whose  $V_{be}(\text{Sat})@I_c/I_b=10$ . For example:

The top curve is  $V_{be}-I_c$  when  $V_{be}$  is saturated and  $I_c/I_b=10$ .



Point corresponding to highest value of collector current

Select a point in the high values range of the collector current.

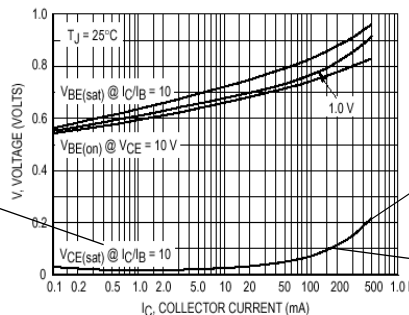
Figure 11. "On" Voltages

2. Find the highest point on the curve. Use the coordinates of this point to enter:
  - **Saturation Base-Emitter Voltage ( $V_{be2\_sat}$ )**
  - **Highest Value of Collector Current**
3. Select a point on the curve in the high values range of the collector current. Use the coordinates of this point to enter:
  - **Saturation Base-Emitter Voltage ( $V_{be1\_sat}$ )**
  - **Collector Current in the high values range**

➤ To enter **Vce(sat)-Ic** data:

1. Using the  $I_c$  vs.  $V_{be}$  graph, locate the curve whose  $V_{be}(\text{Sat})@I_c/I_b=10$ . For example:

The bottom curve is  $V_{be}-I_c$  when  $V_{ce}$  is saturated and  $I_c/I_b=10$ .



Point corresponding to highest value of collector current

Select a point in the high values range of the collector current.

Figure 11. "On" Voltages

2. Find the highest point on the curve. Use the coordinates of this point to enter:
  - **Saturation Base-Emitter Voltage ( $V_{ce2\_sat}$ )**
  - **Highest Value of Collector Current**

3. Select a point on the curve in the high values range of the collector current. Use the coordinates of this point to enter:

- **Saturation Base-Emitter Voltage ( $V_{ce1\_sat}$ )**
- **Collector Current in the high values range**

➤ To enter **Current-Gain Bandwidth Product Chart ( $f_T$ )** data:

1. In the databook, locate the “Current-Gain Bandwidth Product versus Frequency” graph — for example:

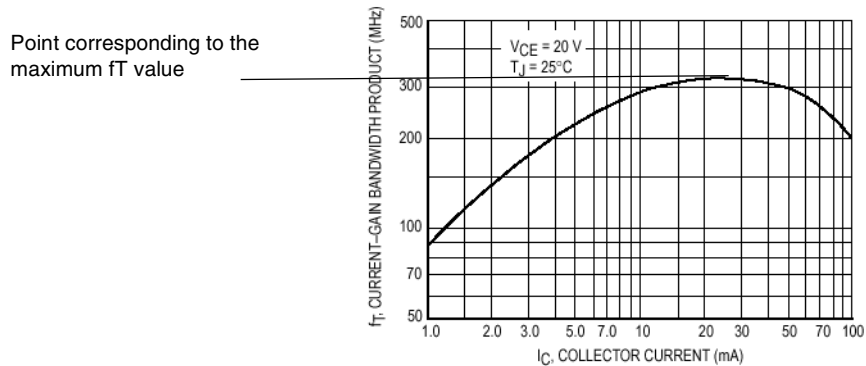


Figure 10. Current-Gain Bandwidth Product

2. Locate the maximum  $f_T$  value, or the highest point, of the curve. Enter this value in the **Maximum Value of Current-Gain Bandwidth Product** field.

➤ To enter **Temperature Coefficients Chart** data:

1. In the databook, locate the “Temperature Coefficients” chart — for example:

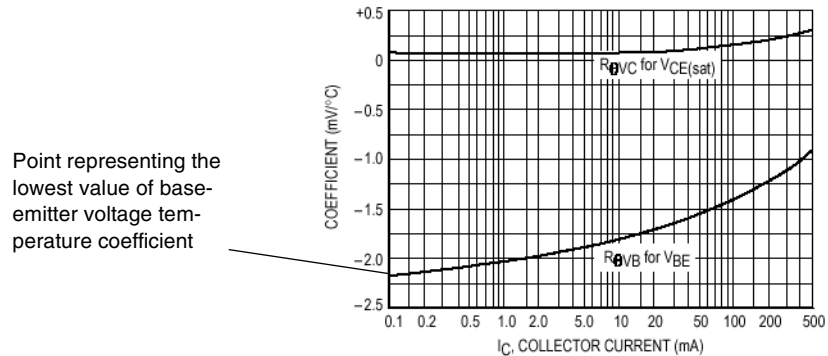


Figure 12. Temperature Coefficients



2. On the base-emitter temperature coefficient curve, find the minimum value, or the lowest point, and enter this value in the **Lowest Value of Base-Emitter Voltage Temperature Coefficient** field.

## 5.13.2 Diode Model Maker

1. From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
2. From the Model Maker list, select Diode and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The Diode Model screen appears.
3. Enter values on the Diode Model screen as described in the following sections.
4. When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

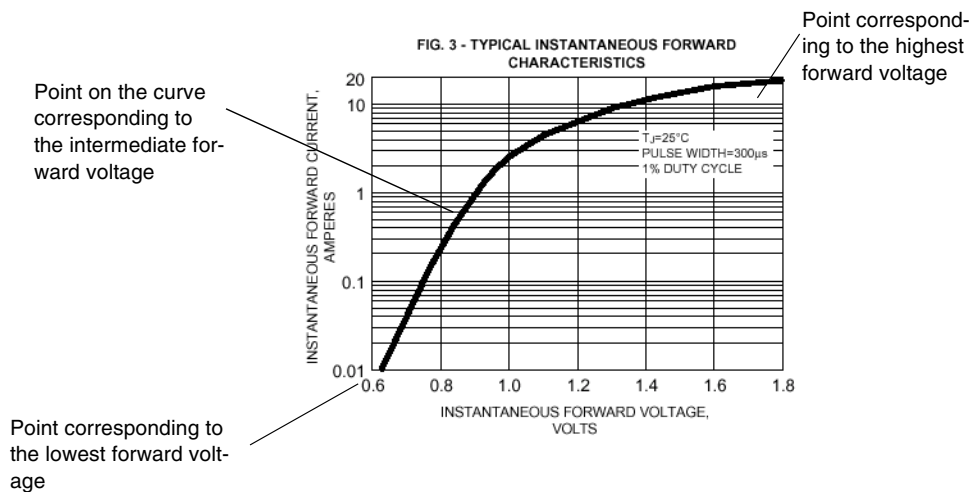
**Note** The Diode Model screen shows preset values for the 1N4001 model.

### Entering General, Maximum Rates, Forward and Reverse Characteristics Data

1. Click the General, Max Rates, Forward and Reverse Characteristics tab:

2. Look up data information for the diode in a databook.

- To enter **General** characteristics, enter the **Component Name**. This is usually found at the top of the data sheet.
- To enter **Reverse Characteristics** data:
  1. In the databook, find the “Maximum Ratings and Electrical Characteristics” table.
  2. Based on information in this table, enter the following values:
    - **Maximum repetitive peak reverse voltage (VRRM)**
    - **Maximum DC reverse current at rated DC blocking voltage (IR)**
    - **Typical reverse recovery time (trr).**
- To enter **Reverse Breakdown** data:
  1. In the databook, find the “Reverse Voltage vs. Reverse Current” chart.  
If no data are available, enable **Reverse Breakdown Data NOT available**.
  2. On the chart, locate the graph that indicates the ambient temperature of 25° C.
  3. Select a point on the graph that represents the mid-point of the horizontal direction, as indicated in the chart.
  4. Use the coordinates of this point to enter values for:
    - **Reverse Breakdown Voltage (BV)**
    - **Reverse Breakdown Current (IBV)**
- To enter **Instantaneous Forward Characteristics** data:
  1. In the databook, locate the “Typical Instantaneous Forward Characteristics” graph — for example:



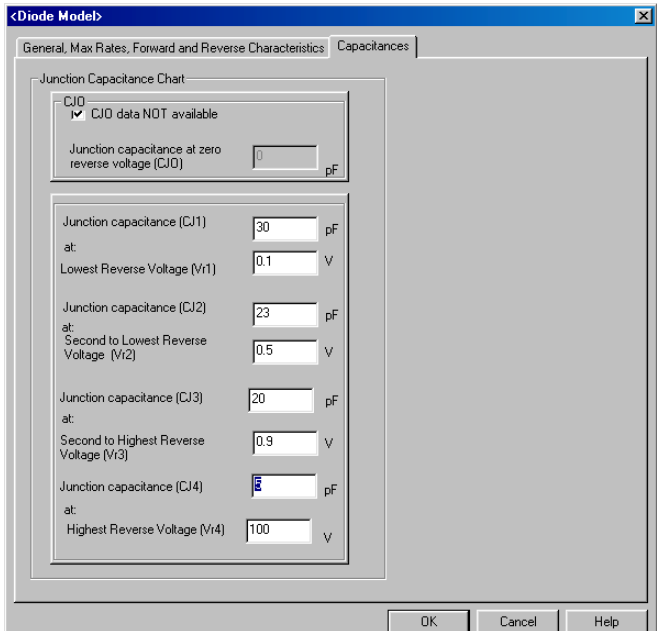
2. Find the point of lowest forward voltage, at beginning point of the curve. Use the coordinates of this point to enter values for:
  - **Lowest forward current (IF1)**
  - **Lowest forward voltage (VF1)**
3. Find the point of highest forward voltage, or the end point on the curve. Use the coordinates of this point to enter values for:
  - **Highest forward current (IFM)**
  - **Highest forward voltage (VFM).**
4. Using your eye or a ruler, find the second or intermediate point on the curve which you think best identifies the transition point in the curve.

**Note** Guidelines for selecting the intermediate point vary from one databook to another. If the graph is provided in logarithmic format, which is usually the case, a good way to find this point is to place a ruler along the beginning of the curve in the lower voltage area, which will look like a straight line. Where the curve begins to diverge from your ruler, use this point as your intermediate point. If the graph is provided in linear format, plot the data in logarithmic fashion and follow the ruler procedure.

5. Use the coordinates of this point to enter the values for:
  - **Intermediate forward current (IF2)**
  - **Intermediate forward voltage (VF2).**

# Entering Capacitances data

1. Click the Capacitances tab:



➤ To enter **Junction Capacitances** data:

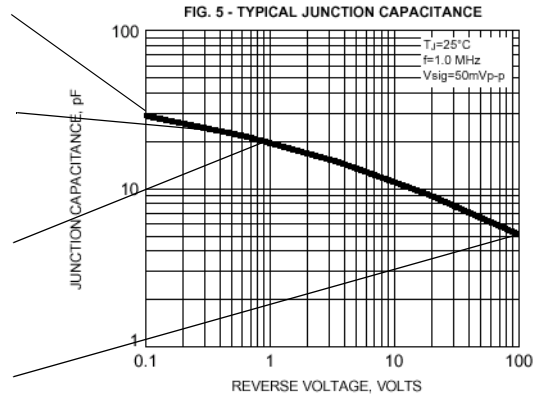
1. In the databook, find the “Typical Junction Capacitance” chart — for example:

Point corresponding to the lowest reverse voltage

Select a second point on the curve in the lower voltage area for the second to lowest reverse voltage.

Select a third point on the curve in the lower voltage area for the third to lowest/second to highest reverse voltage

Point corresponding to the highest reverse voltage



2. Find the junction capacitance at zero reverse voltage and enter it in the **Junction capacitance at zero reverse voltage (CJO)** field.

If this information is not given in the databook, enable **CJO data NOT available**.

3. Find the point of lowest reverse voltage, or the beginning point of the curve. Use the coordinates of this point to enter the values for:
  - **Junction capacitance (CJ1)**
  - **Lowest Reverse Voltage (Vr1).**

4. Find the point of highest reverse voltage, or the end point on the curve) and enter the coordinate values in the **Junction capacitance (CJ4)** and **Highest Reverse Voltage (Vr4)** fields.

5. Select two additional intermediate points on the graph, greater than the lowest reverse voltage but in the lower range of the reverse voltage.

6. Use the coordinate values of the second point to enter:
  - **Junction capacitance (CJ2)**
  - **Second to Lowest Reverse Voltage (Vr2).**

7. Use the coordinate values of the third point to enter:
  - **Junction Capacitance (CJ3)**
  - **Second to Highest Reverse Voltage (Vr3)** (Third to lowest reverse voltage)

### 5.13.3 MOSFET (Field Effect Transistor) Model Maker

1. From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
2. From the Model Maker list, select MOSFET and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The MOSFET Model screen appears.
3. Enter values on the MOSFET Model screen as described in the following sections.
4. When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

**Note** The MOSFET Model screen shows preset values for the BSS83 model.

#### Entering General and Output Characteristics Data

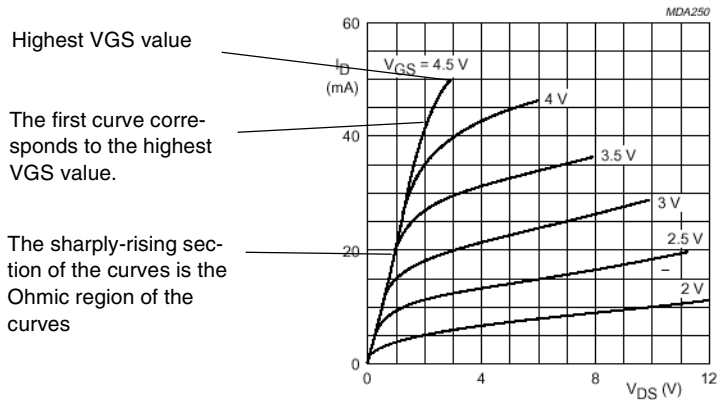
1. Click the General, Output Characteristics tab:

The screenshot shows the 'MOSFET Model' dialog box with the 'General, Output Characteristics' tab selected. The 'General' section contains fields for 'Component name' (BSS83), 'Channel Type of Mosfet' (NMOS), and 'Max Drain Current' (50 mA). The 'Output Characteristics in Ohmic region' section has fields for 'Vgs for the curve (Vgs\_ohmic)' (4.5 V), 'Drain Current (Ids\_ohmic)' (27 mA), and 'Vds when drain current is Ids\_ohmic (Vds\_ohmic)' (1.3 V). The 'Output Characteristics' section contains three columns for 'Ids\_Vds curve 1 (for lowest Vgs)', 'Ids\_Vds curve 2 (for intermediate Vgs)', and 'Ids\_Vds curve 3 (for highest Vgs)'. Each column has fields for 'Vgs for this curve', 'Drain Current', and 'Vds' at two different points (lower and higher Vds).

Ids_Vds curve 1 (for lowest Vgs)			Ids_Vds curve 2 (for intermediate Vgs)			Ids_Vds curve 3 (for highest Vgs)		
Vgs for this curve (Vgs_f0)	2	V	Vgs for this curve (Vgs_f1)	3	V	Vgs for this curve (Vgs_f2)	4.5	V
Drain Current (Ids_f00)	5	mA	Drain Current (Ids_f10)	20	mA	Drain Current	40	mA
at Vds_f00 (lower Vds)	2	V	at Vds_f01 (lower Vds)	3	V	at Vds_f20 (lower Vds)	3	V
Drain Current (Ids_f01)	10	mA	Drain Current (Ids_f11)	27	mA	Drain Current (Ids_f21)	47	mA
at Vds_f01 (higher Vds)	10	V	at Vds_f11 (higher Vds)	9	V	at Vds_f21 (higher Vds)	6	V

2. Look up data information for the MOSFET in a databook.
- To enter **General** data:
1. Enter the **Component Name**. This can usually be found in the top right corner of the datasheet.
  2. Enter the **Channel Type of MOSFET**. This is the title of the datasheet and is found at the top of the datasheet.

3. Find the “Ratings” table for the MOSFET.
4. From the data given in the table, enter the **Max drain current**
- To enter **Output Characteristics in Ohmic Region** data:
  1. From the MOSFET data information, find the  $I_d$  vs.  $V_{ds}$  graph — for example:



$T_j = 25^\circ\text{C}$ .

Fig.3  $V_{SB} = 0$ ; typical values.

2. Locate the curve with the highest  $V_{GS}$ . Enter this  $V_{GS}$  value in the  **$V_{GS}$  for the curve ( $V_{gs\_ohmic}$ )**.
3. Locate a point in the ohmic region of the same curve.
4. Enter the  $I_d$  value of this point in the **Drain Current ( $I_{ds\_Ohmic}$ )** field.
5. Enter the  $V_{ds}$  value of this point in the  **$V_{ds}$  when drain current is  $I_{ds\_Ohmic}$  ( $V_{ds\_Ohmic}$ )** field.
- To enter **Output Characteristics for Saturation Region** data:
  1. Using the same graph as above, locate the saturation region of the curves. The saturation region is the steady state situation of the curves where points along the curve fit on a

straight line. (The curve corresponding to the highest  $V_{GS}$  does not have a saturation region.) For example:

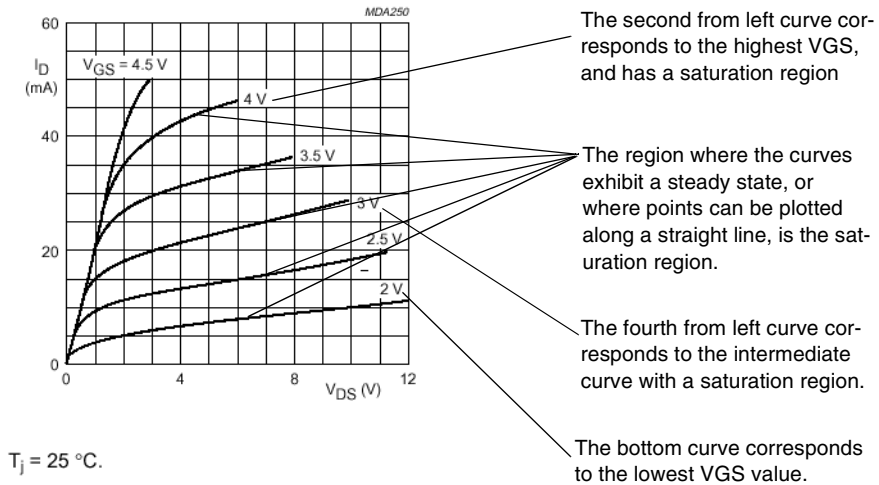


Fig.3  $V_{SB} = 0$ ; typical values.

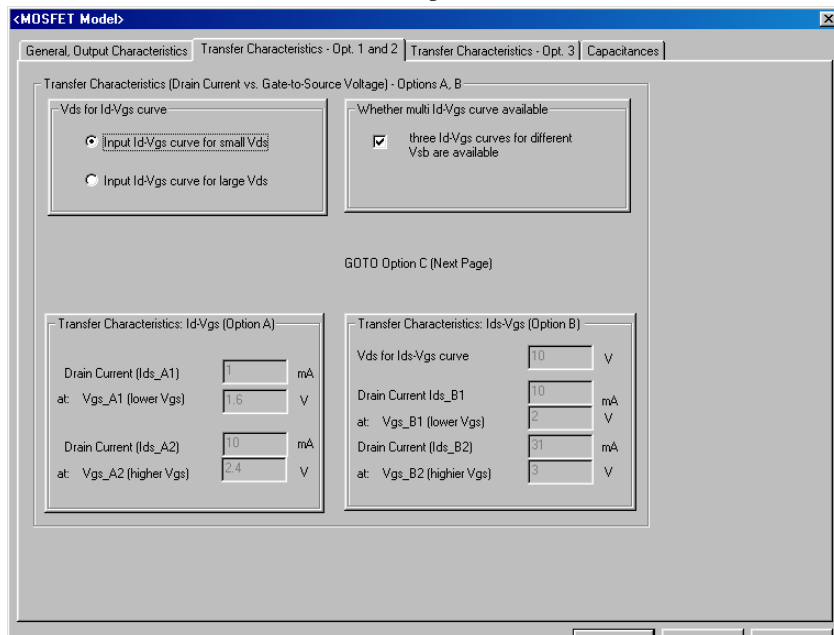
2. To enter data for the **Ids\_Vds curve 1 (for lowest Vgs)** fields, find the curve with the lowest  $V_{GS}$  value.
3. Enter the  $V_{GS}$  value of this curve in the **Vgs for this curve (Vgs\_f0)** field.
4. Using your eye or a ruler, locate the beginning and end points of the saturation region, or the area where the points fit along a straight line, for this curve.
5. Use the coordinates of the beginning point to enter:
  - **Drain Current (Ids\_f00)**
  - **Vds\_f00 (lower Vds).**
6. Use the coordinates of the end point to enter:
  - **Drain Current (Ids\_f01)**
  - **Vds\_f01 (higher Vds).**
7. To enter data for the **Ids\_Vds curve 3 (for highest Vgs)** fields, find the curve with the highest  $V_{GS}$  value, but which still has a saturation region. (This description excludes the topmost curve of the  $I_D$ - $V_{DS}$  graph.)
8. Repeat steps 4 through 7 to enter values for:
  - **Vgs for this curve (Vgs\_f2)**
  - **Drain Current**
  - **Vds\_f20 (lower Vds).**



- **Drain Current (Ids\_f21)**
  - **Vds\_f21 (higher Vds).**
9. To enter data for the **Ids\_Vds curve 2 (for intermediate Vgs)** fields, find the curve in the middle point between the curves corresponding to the lowest Vgs and the highest VGS with a saturation region.
  10. Repeat steps 4 through 7 to enter values for:
    - **Vgs for this curve (Vgs\_f1)**
    - **Drain Current (Ids\_f10)**
    - **Vds\_f01 (lower Vds).**
    - **Drain Current (Ids\_f11)**
    - **Vds\_f11 (higher Vds).**

## Entering Transfer Characteristics data

1. Click the Transfer Characteristics - Opt. 1 and 2 tab:



- To select Transfer Characteristics options:
1. In the databook, locate the Id vs. Vgs graph. Depending on the available data, under **Vds for Id-Vgs curve** and **Whether multi Id-Vgs curve available**, enable the appropriate options.
- Note** If the graph contains more than one Vsb curve, it implies that source and bulk (substrate) are not connected together.

If the latter option is not enabled, you will be prompted to enter data in the **Option A** or **Option B** fields on the same screen.

If the latter option is enabled (as it is in our example), the screen will prompt you to **Go to Option C (Next Page)**.

2. To go to Option C, click the Transfer Characteristics - Opt. 3 tab:

The screenshot shows the 'MOSFET Model' dialog box with the 'Transfer Characteristics - Opt. 3' tab selected. The dialog is titled 'Transfer Characteristics (Drain Current vs. Gate-to-Source Voltage) - Option C'. It contains three sections for defining Ids-Vgs curves for different Vsb values.

Vsb for this curve (Vsb_C1)	Drain Current (Ids_C11) at Vgs_C11 (lower Vgs)	Drain Current (Ids_C12) at Vgs_C12 (higher Vgs)
0 V	1 mA at 1.6 V	10 mA at 2.4 V

Vsb for this curve (Vsb_C2)	Drain Current (Ids_C21) at Vgs_C21 (lower Vgs)	Drain Current (Ids_C22) at Vgs_C22 (higher Vgs)
-4 V	1 mA at 2 V	10 mA at 2.75 V

Vsb for this curve (Vsb_C3)	Drain Current (Ids_C31) at Vgs_C31 (lower Vgs)	Drain Current (Ids_C32) at Vgs_C32 (higher Vgs)
-12 V	1 mA at 2.25 V	10 mA at 3 V

Buttons at the bottom: OK, Cancel, Help.

- To enter **Transfer Characteristics (Drain Current vs. Gate-to-Source Voltage)** data for all three options:

1. Look at the data in the  $I_{ds}$  vs.  $V_{gs}$  graph — for example:

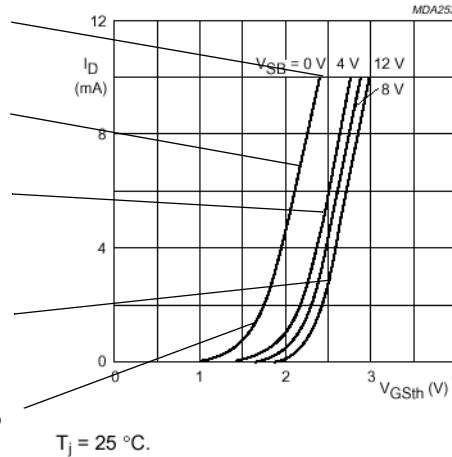
The highest point on the curves corresponds to the maximum  $I_d$  value.

The leftmost curve corresponds to the lowest  $V_{sb}$  value.

The second from left curve corresponds to the intermediate  $V_{sb}$  value.

The rightmost curve corresponds to the highest  $V_{sb}$  value.

Point corresponding to 10% of maximum  $I_d$  value



Multiple  $V_{sb}$  curves imply that source-bulk (substrate) are not connected together.

Fig.6  $V_{DS} = V_{GS} = V_{GS(th)}$ .

2. If you are using Option A, proceed to step 4.  
If you are using Option B, proceed to step 3.  
If you are using Option C, to enter data in the **Ids-Vgs Curve for lowest Vsb** fields, locate the curve with the lowest  $V_{sb}$ .
3. If you are using Option B, enter the  $V_{ds}$  value in the **Vds for Ids-Vgs curve** field.  
If you are using Option C, enter the  $V_{sb}$  value in the **Vsb for this curve (Vsb\_C1)** field.
4. Find the maximum  $I_d$ , or the highest point of the curve. Use the coordinates for this point to enter:  
for Option A:
  - **Drain Current (Ids\_A1)**
  - **Vgs\_A1 (lower Vgs)**
 for Option B:
  - **Drain Current Ids\_B1**
  - **Vgs\_B1 (lower Vgs)**

for Option C:

- **Drain Current (Ids\_C11)**
- **Vgs-C11 (lower Vgs)**

5. Find the point on the curve which corresponds to 10% of the maximum Id on the same curve. Use the coordinates of this point to enter:

for Option A:

- **Drain Current (Ids\_A2)**
- **Vgs\_A2 (higher Vgs)**

for Option B:

- **Drain Current (Ids\_B2)**
- **Vgs\_B2 (higher Vgs)**

for Option C:

- **Drain Current (Ids\_C12)**
- **Vgs\_C12 (higher Vgs)**

6. To complete the Option 3 screen, to enter data in the **Ids-Vgs Curve for highest Vsb**, find the curve with the highest Vsb value, and repeat steps 3 through 5 above to enter data for:

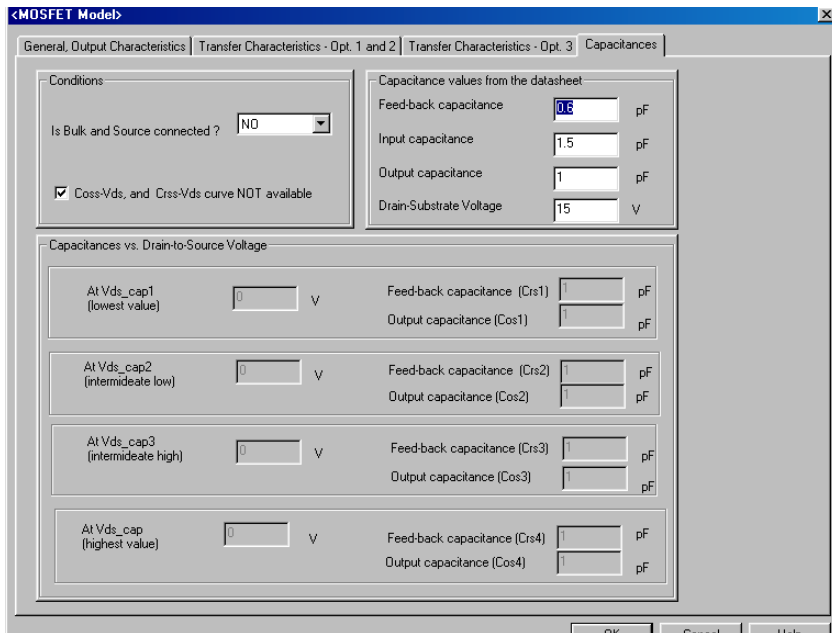
- **Vbs for this curve (Vsb\_C3)**
- **Drain Current (Ids\_C31)**
- **Vgs\_C21 (lower Vgs)**
- **Drain Current (Ids\_C32)**
- **Vgs\_C22 (highest Vgs)**

7. To enter data in the **Ids-Vgs Curve for Intermediate Vsb** fields, select the curve corresponding to a Vsb value in between the highest and lowest Vsb. Repeat steps 3 through 5 above to enter data for:

- **Vsb for this curve (Vsb\_C2)**
- **Drain Current (Ids\_C21)**
- **Vgs\_C21 (lower Vgs)**
- **Drain Current (Ids\_C22)**
- **Vgs\_C22 (highest Vgs)**

## Entering Capacitances Data

1. Click the Capacitances tab:



➤ To enter **Conditions**:

1. Determine whether the bulk and source of the model are connected, and select the appropriate answer from the **Is Bulk and Source connected?** drop-down list.

**Note** The substrate condition can be determined by two means. The first is to check the schematic of the device where the internal connections of the MOS transistor are shown. The second is to check the Id-Vgs graph. If the graph contains more than one Vsb curve, it suggests that source-bulk (substrate) are not connected together.

2. In the databook, locate the “Capacitances vs. Drain-to-Source Voltage” chart. If it is available, you may enter data in the **Capacitances vs. Drain-to-Source Voltage** fields. If it is not available, enable **Coss-Vds and Crss-Vds curve NOT available**, and use the datasheet to enter capacitances.

- 
- To enter **Capacitance values from the datasheet**:

1. In the databook, find the “Characteristics” table
2. From the table, enter data for:
  - **Feedback capacitance**
  - **Input capacitance**
  - **Output capacitance**
  - **Source-Gate Voltage**

## 5.13.4 Operational Amplifier Model Maker

1. From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
2. From the Model Maker list, select Operational Amplifier and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The Operational Amplifier Model screen appears.
3. Enter values on the Operational Amplifier Model screen as described in the following sections.
4. When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

**Note** The Operational Amplifier Model screen shows preset values for the  $\mu$ A741 model.

## Entering General and Input Data

1. Click the General Input tab:

The screenshot shows the '<Operational Amplifier Model>' dialog box with the 'General Input' tab selected. The 'General' section contains 'Component Name' set to 'uA741' and 'Transistor Type in input stage' set to 'NPN'. A note states: 'NOTE: Preset values are for uA741'. The 'Input' section contains the following parameters and values:

Parameter	Value	Unit
Input Capacitance (Ci)	1.4	pF
Input Offset Current (Iio)	20	nA
Input Bias Current (Iib)	80	nA
Input Offset Voltage (Vio)	1	mV
Common-mode Input Resistance (Rcm)	2	GOhm
Differential-mode Input Resistance (R1)	2	MOhm
Common-mode Rejection Ratio (CMRR)	90	dB

The 'Voltage Gain Avd' section has two radio buttons: 'in dB' (selected) and 'in V/mV'. The 'Voltage Gain (Avd)' is set to 106 dB.

Buttons at the bottom: OK, Cancel, Help.

2. Look up data information for the operational amplifier in a databook.

➤ To enter **General** data:

1. In the **Component Name** field, enter the appropriate name of the component. This is usually found in the top right corner of the datasheet.
2. In the **Transistor Type in input stage** field, select the type of transistor used in the input stage. This can be determined by looking at the schematic of the internal structure of the opamp.

**Note** This information is optional, as the opamp model can be based on any type of input transistor. If the type of the input transistor is not important, select the “Don’t Care” option.

➤ To enter **Input** data:

1. In the databook, find the two tables labelled “Electrical Characteristics at specified free-air temperature”.
2. Use the data from these tables to enter:
  - **Input Capacitance (I)**
  - **Input Offset Current (I1o)**
  - **Input Bias Current ((I1b)**
  - **Input Offset Voltage (V1o)**
  - **Common-mode Input Resistance (Rcm)**
  - **Differential-mode Input Resistance (R1)**
  - **Common-mode Rejection Ratio (CMRR)**
  - **Voltage Gain Avd**

**Note** The Common-Mode input resistance is usually very high. If its value is not available, choose 2 Gohm as the default.

**Note** While the typical value for the Common-Mode Rejection Ratio (CMRR) is provided in the “Electrical Characteristics” table, its variations with frequency are also provided in a chart called “Common-Mode Rejection Rate Vs. Frequency”. If you use this chart, use the CMRR value for the lowest frequency possible.

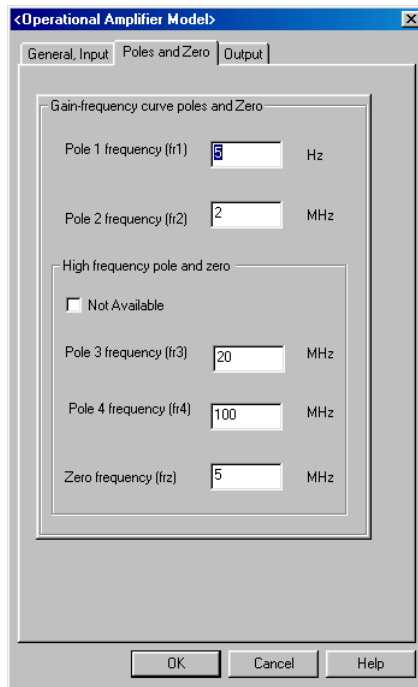
**Note** While the typical value for Large Signal Differential Voltage Amplification (Avd) is provided in the “Electrical Characteristics” table, you can also find it in a chart called “Open-Loop Large Signal Differential Voltage”. If you use this chart, use the Avd value at the lowest frequency.

**Note** Databooks provide Avd gain in either dB or V/mV. If the value is provided in V/mV, you can still enter the data in dB. However, you should convert the numerical values:  
 $value\ in\ dB = 20 * \log[1000 * (value\ in\ V/mv)]$



## Entering Poles and Zero Data

1. Click the Poles and Zero tab:



- To enter **Gain-frequency curve poles and Zero** data:
  1. In the databook, locate the “Avd-Open-Loop Single Differential Voltage Amplification vs. Frequency” chart — for example:

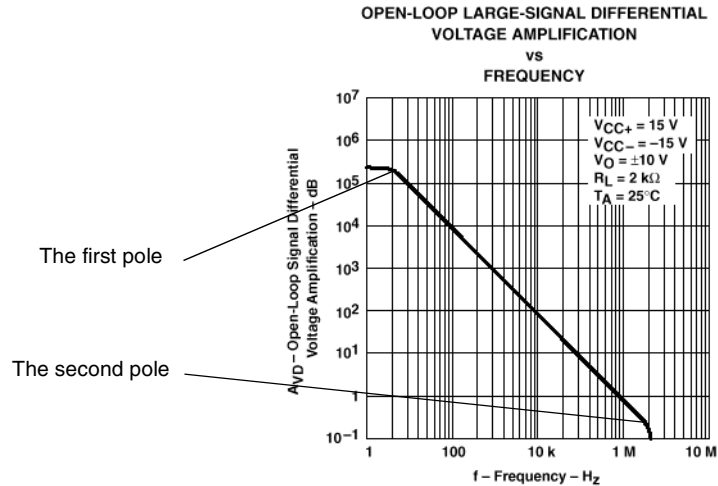
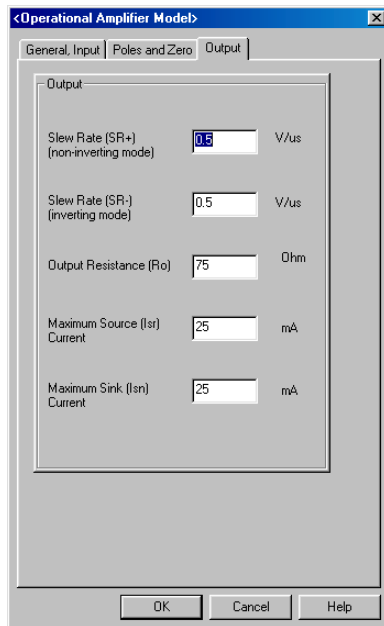


Figure 8

2. Find the first pole on the curve, or the point on the curve where the first horizontal line transitions into a slope. Enter the frequency value for this point in the **Pole 1 frequency (fr1)** field.
  3. Find the second pole on the curve, or the point where the slope transitions into a sharper slope. Enter the frequency value for this point in the **Pole 2 frequency (fr2)** field.
- To enter **High frequency pole and zero** data, find higher frequency poles using the curve mentioned above, web sites or books. If these pieces of information are not available, enable **Not Available**.

## Entering Output Data

1. Click the Output tab:



2. In the databook, locate the “Operating Characteristics” table.
3. Use the data from this table to enter:

- **Slew Rate (SR+) (non-inverting mode)**
- **Slew Rate (SR-) (inverting mode)**

**Note** Databooks may provide only one value for both inverted and non-inverted slew rates.

4. Refer to the “Electrical Characteristics” tables mentioned in the previous section. Use the data from these tables to enter:

- **Output Resistance (Ro)**
- **Maximum Source (Isr) Current**
- **Maximum Sink (Isn) Current**

**Note** Databooks normally provide the short circuit output current. This is the maximum value of the output current which the output node can provide if it is connected to the negative power supply, or can accept if it is shorted to the positive power supply. You should enter its value regardless of its sign.

## 5.13.5 Silicon Controlled Rectifier Model Maker

1. From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
2. From the Model Maker list, select SCR and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The SCR Model screen appears.
3. Enter values on the SCR Model screen as described in the following sections.
4. When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

**Note** The SCR Model screen shows preset values for the 2N6504 SCR.

### Entering Electrical and Maximum Forward Voltage Data

1. Click the Electrical Data, Max Forward Voltage tab:.

The screenshot shows the "<SCR Model>" dialog box with the "Electrical Data, Max Forward Voltage" tab selected. The "Time Data, Max Ratings" tab is also visible. The dialog contains the following fields and values:

Electrical Characteristics	
Holding Current	35 mA
Gate Trigger Current	40 mA
Gate Trigger Voltage	1.5 V
Peak Forward Blocking Current	10 uA
Peak Forward Blocking Voltage	50 V
1.05 Peak Reverse Blocking Voltage	52.5 V
Critical Rate of Rise of Off-State Voltage	50 V/us

Maximum Forward Voltage Chart	
Instantaneous Forward Current at: Minimum Value of Instantaneous Voltage	0.1 A, 0.85 V
Instantaneous Forward Current at: Intermediate Value of Instantaneous Voltage	3 A, 1 V
Instantaneous Forward Current at: Maximum Value of Instantaneous Voltage	100 A, 2.55 V

At the bottom of the dialog are buttons for "OK", "Cancel", and "Help".

2. Look up data information for the SCR in a databook.
  3. Enter the name of the model builder (optional).
  4. Enter the device type number. This is usually found in the top right corner of the datasheet.
- To enter **Electrical Characteristics** data:

1. Locate the “Electrical Characteristics” table — for example:

Use this information to enter data in the **Electrical Characteristics** fields under the Electrical Data, Max Forward Voltage tab.

Use this information to enter data in the **Electrical Characteristics** fields of under the Time Data, Max Ratings tab. See “Entering Time Data and Maximum Ratings Data” on page 5-65.

ELECTRICAL CHARACTERISTICS (T<sub>C</sub> = 25°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
*Peak Forward or Reverse Blocking Current (V <sub>AK</sub> = Rated V <sub>DRM</sub> or V <sub>RRM</sub> , Gate Open) T <sub>J</sub> = 25°C T <sub>J</sub> = 125°C	I <sub>DRM</sub> , I <sub>RRM</sub>	— —	— —	10 2	μA mA
*Forward “On” Voltage <sup>(1)</sup> (I <sub>TM</sub> = 50 A)	V <sub>TM</sub>	—	—	1.8	Volts
*Gate Trigger Current (Continuous dc) (Anode Voltage = 12 Vdc, R <sub>L</sub> = 100 Ohms) T <sub>C</sub> = 25°C T <sub>C</sub> = -40°C	I <sub>GT</sub>	— —	— 25	40 75	mA
*Gate Trigger Voltage (Continuous dc) (Anode Voltage = 12 Vdc, R <sub>L</sub> = 100 Ohms, T <sub>C</sub> = -40°C)	V <sub>GT</sub>	—	1	1.5	Volts
Gate Non-Trigger Voltage (Anode Voltage = Rated V <sub>DRM</sub> , R <sub>L</sub> = 100 Ohms, T <sub>J</sub> = 125°C)	V <sub>GD</sub>	0.2	—	—	Volts
*Holding Current (Anode Voltage = 12 Vdc, T <sub>C</sub> = -40°C)	I <sub>H</sub>	—	35	40	mA
*Turn-On Time (I <sub>TM</sub> = 25 A, I <sub>GT</sub> = 50 mAdc)	t <sub>gt</sub>	—	1.5	2	μs
Turn-Off Time (V <sub>DRM</sub> = rated voltage) (I <sub>TM</sub> = 25 A, I <sub>R</sub> = 25 A) (I <sub>TM</sub> = 25 A, I <sub>R</sub> = 25 A, T <sub>J</sub> = 125°C)	t <sub>q</sub>	— —	15 35	— —	μs
Critical Rate of Rise of Off-State Voltage (Gate Open, Rated V <sub>DRM</sub> , Exponential Waveform)	dv/dt	—	50	—	V/μs

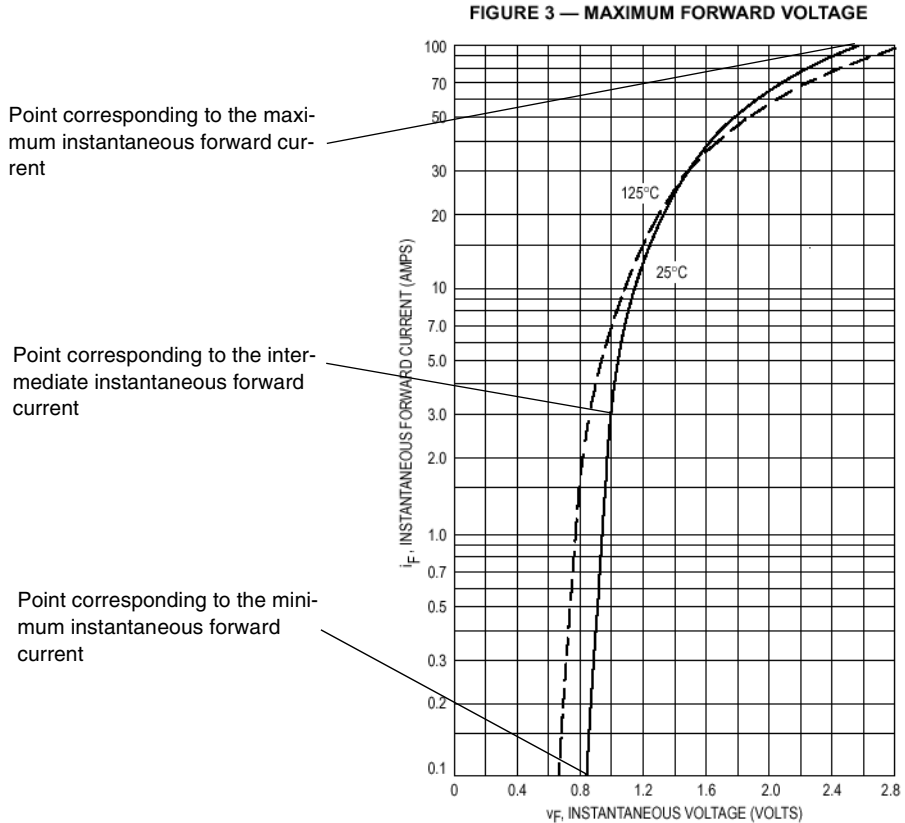
\*Indicates JEDEC Registered Data.

1. Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2%.

2. Based on the data provided this table, enter:
  - **Holding Current**
  - **Gate Trigger Current**
  - **Gate Trigger Voltage**
  - **Peak Forward Blocking Current**
  - **Critical Rate of Rise of Off-State Voltage**
3. In the databook, locate the “Maximum Ratings” table.
4. Based on the data in the table, enter the value of the **Peak Forward Blocking Voltage** field.
5. Multiply this value by 1.05 and enter the value in the **1.05 Peak Reverse Blocking Voltage** field.

➤ To enter **Maximum Forward Voltage Chart** data:

1. In the databook, locate the “Instantaneous Forward Current vs. Instantaneous Voltage” graph, and find the  $I_f$ - $V_f$  curve at 25°. For example:



2. On the curve, find the point at the minimum  $I_f$ , or the beginning point of the curve. Use the coordinates of this point to enter:
  - **Instantaneous Forward Current.**
  - **Minimum Value of Instantaneous Voltage.**
3. Find the point at the maximum  $I_f$ , or the end point of the curve. use the coordinates of this point to enter:
  - **Instantaneous Forward Current**
  - **Maximum Value of Instantaneous Voltage.**
4. Locate an intermediate point on the curve corresponding to the transition point. Since the graph is provided in logarithmic format, you can do this by using a ruler to draw a line starting at the beginning point and following the straight line of the curve in the lower

voltage area. Where the curve begins to diverge from your ruler, use this point as your intermediate point. Use the coordinates of this point to enter:

- **Instantaneous Forward Current**
- **Intermediate Value of Instantaneous Voltage**

## Entering Time Data and Maximum Ratings Data

1. Click the Time Data, Max Ratings tab:

- To enter **Electrical Characteristics** data, refer to the “Electrical Characteristics” table mentioned in the previous section, and enter data in the **Turn-On Time** and **Turn-Off Time** fields.
- To enter **Maximum Ratings Chart** data:
  1. Refer to the “Maximum Ratings” table mentioned in the previous section.
  2. Find the Forward Current and enter this value in the **Forward Current** field.
  3. For the **Reverse Current** field, find the reverse current (IRC) when the device is in off-state and enter this value, or, if this value is not provided, enter 0.
  4. For the **Reverse Voltage** field, find the reverse voltage (VRC) when the device is in off-state or, if this value is not provided, enter 0.
  5. For the **Identifier** field, enter 1 if Reverse Current and Reverse Voltage values are available, or 0 if they are unavailable.

- For the first **Parameter related to “off-state”** field, enter 0 if the Reverse Current and Reverse Voltage values are available, or, if they are not provided, enter the Forward Current value.
- For the second **Parameter related to “off-state”** field, enter 0 if the Reverse Current and Reverse Voltage values are available, or, if they are not provided, enter the Peak Reverse Blocking Voltage value.

## 5.13.6 Zener Model Maker

- From the Model tab of the Component Properties screen, click **Start Model Maker**. The Select Model Maker screen appears.
- From the Model Maker list, select Zener and, to continue, click **Accept**. (Click **Cancel** to return to the Model tab.) The Zener Model screen appears.
- Enter values on the Zener Model screen as described in the following sections.
- When all values are entered, click **OK** to complete the model, or click **Cancel** to cancel.

**Note** The Zener Model screen shows preset values for the 1N4728A model.

### Entering Electrical Characteristics Data

- Click the Electrical Characteristics tab:

**<Zener Model>**

Electr. Characteristics | Temp. Coeff., Capacitance

Component Name  
(Defaults are for 1N4728A) **1N4728A**

Revers Characteristics		Forward Characteristic	
Nominal Zener Voltage (Vzt) at:	3.3 V	Forward Current (If1) at:	1 mA
Zener Test Current (Izt)	76 mA	Lowest Forward Voltage (Vf1)	0.73 V
Nominal Zener Impedance at Zzt and Izt (Zzt)	4 Ohms	Forward Current (If2) at:	10 mA
Leakage Current (Iir) at:	100 uA	Intermediate Forward Voltage (Vf2)	0.8 V
Reverse Test Voltage (Vr)	1 V	Highest Forward Current (Ifm) at:	200 mA
Zener Impedance near Breakdown Knee (Zzk)	300 Ohms	Highest Forward Voltage (Vfm)	1.05 V
Zener Current near Breakdown Knee (Izk)	1 mA		

OK Cancel Help

- Look up data information for the Zener diode in a databook.



3. From the databook, locate the “Electrical Characteristics” table — for example:

Use the information from this table to enter data in the **Reverse Characteristics** fields.

\*ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$  unless otherwise noted)  $V_F = 1.2\text{ V Max}$ ,  $I_F = 200\text{ mA}$  for all types.

JEDEC Type No. (Note 1)	Nominal Zener Voltage $V_Z @ I_{ZT}$ Volts (Notes 2 and 3)	Test Current $I_{ZT}$ mA	Maximum Zener Impedance (Note 4)			Leakage Current		Surge Current @ $T_A = 25^\circ\text{C}$ $I_r - \text{mA}$ (Note 5)
			$Z_{ZT} @ I_{ZT}$ Ohms	$Z_{ZK} @ I_{ZK}$ Ohms	$I_{ZK}$ mA	$I_r$ $\mu\text{A Max}$	$V_R$ Volts	
1N4728A	3.3	76	10	400	1	100	1	1380
1N4729A	3.6	69	10	400	1	100	1	1260
1N4730A	3.9	64	9	400	1	50	1	1190
1N4731A	4.3	58	9	400	1	10	1	1070
1N4732A	4.7	53	8	500	1	10	1	970
1N4733A	5.1	49	7	550	1	10	1	890
1N4734A	5.6	45	5	600	1	10	2	810
1N4735A	6.2	41	2	700	1	10	3	730
1N4736A	6.8	37	3.5	700	1	10	4	660
1N4737A	7.5	34	4	700	0.5	10	5	605
1N4738A	8.2	31	4.5	700	0.5	10	6	550

4. Enter the name of the component in the **Component Name** field.

➤ To enter **Reverse Characteristics** data, use the information from the table for the following fields:

- **Nominal Zener Voltage (Vzt)**
- **Zener Test Current (Izt)**
- **Nominal Zener Impedance at Zzt and Izt (Zzt)**
- **Leakage Current (Ir)**
- **Reverse Test Voltage (Vr)**
- **Zener Impedance near Breakdown Knee (Zzk)**
- **Zener Current near Breakdown Knee (Izk)**

**Note** In the example, the databook only provides the maximum Zener impedance. To find a typical value for Zzk, use 0.75 times the maximum value of Zzk. To find the typical value of Zzt, you can use the Zz-Iz graph. Find or estimate a curve at the nominal zener voltage given in the table, and choose the point which corresponds to the test current given in the table. Use the Zz coordinate of this point to enter as the typical value.

➤ To enter **Forward Characteristics** data:

1. In the databook, locate the  $I_f$ - $V_f$  graph, and find the maximum curve at 25° — for example:

The sixth from left curve is the maximum curve at 25°.

Point corresponding to maximum forward voltage

Point corresponding to intermediate forward voltage

Point corresponding to minimum forward voltage

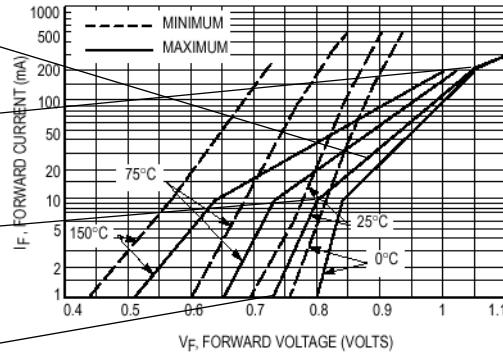
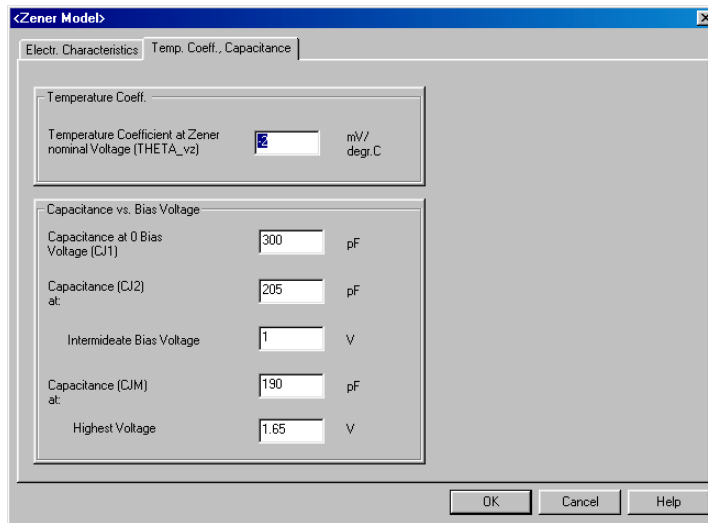


Figure 10. Typical Forward Characteristics

2. Find the point on the curve with the lowest forward voltage, or the beginning point. Use the coordinates of this point to enter:
  - **Forward Current ( $I_{f1}$ )**
  - **Lowest Forward Voltage ( $V_{f1}$ )**
3. Find the knee point on the curve, or the point where the slope changes drastically. Use the coordinates of this point to enter:
  - **Forward Current ( $I_{f2}$ )**
  - **Intermediate Forward Voltage ( $V_{f2}$ )**
4. Find the point of maximum forward voltage, or the highest point on the curve. Use the coordinates of this point to enter:
  - **Highest Forward Current ( $I_{fm}$ )**
  - **Highest Forward Voltage ( $V_{fm}$ )**

## Entering Temperature Coefficient and Capacitance Data

1. Click the Temp. Coeff., Capacitance tab:



To enter **Temperature Coefficient** data:

1. From the databook, find the Temperature Coefficient versus Zener Voltage graph — for example:

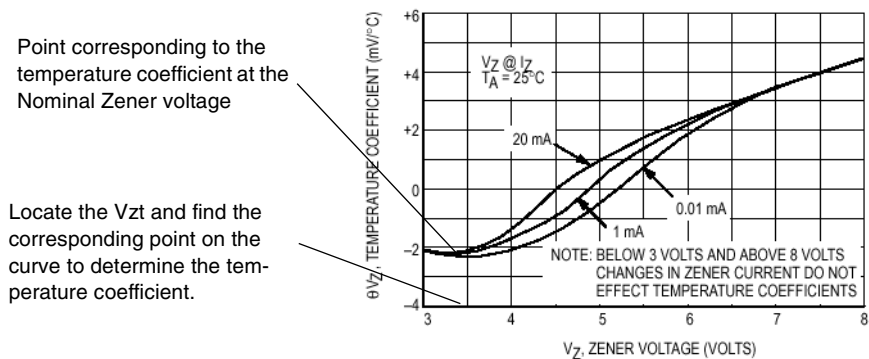


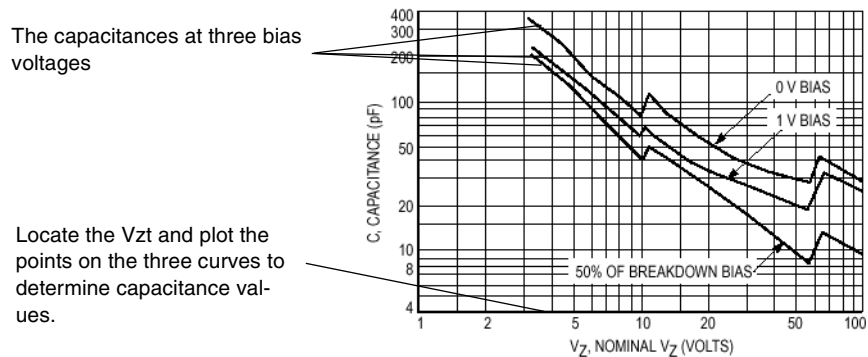
Figure 4. Effect of Zener Current

2. Locate the curve corresponding to the model's test current as given in the "Electrical Characteristics" table. (If it is not on the graph, estimate its placement.)

- Find the point corresponding to the model's  $V_Z$ , as provided in the “Electrical Characteristics” table. Enter the Temperature Coefficient for this point in the **Temperature Coefficient at Zener Nominal Voltage (THETA\_vz)** field.

➤ To enter **Capacitance vs. Bias Voltage** data:

- In the databook, locate the “Capacitance versus Nominal  $V_Z$ ” graph — for example:



- On each of the three curves in the graph, locate the point corresponding to the  $V_{zt}$  provided in the “Electrical Characteristics” table.
- For the curve at zero bias voltage, use this point to enter the capacitance value in the **Capacitance at 0 Bias Voltage (CJ1)** field.
- For the intermediate curve, enter its bias voltage in the **Intermediate Bias Voltage** field and enter the capacitance value for the point you have marked in the **Capacitance (CJ2)** field.
- For the curve with the highest voltage, enter its bias voltage in the **Highest Voltage** field. (In our example, this value is 50% of the nominal Zener voltage ( $V_{zt}$ ) of the model, as provided in the “Electrical Characteristics” table.) Enter the capacitance value for the point you have marked in the **Capacitance (CJM)** field.

## 5.14 Creating a Model Using Code Modeling



This section explains how to model a component using a high-level, industry-standard programming language: C. The component can then be added to the Multisim database. To use code modeling you must have a C compiler such as Microsoft Visual C++, Version 4.1 or greater, and be familiar with programming and compiling C code. This section is not designed for Multisim users without programming exposure.

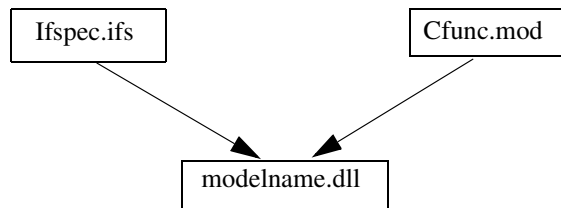
Multisim has built-in models for most types of devices, but it is impossible to provide models for every possible device. The behavior of some devices may be extremely difficult to model as groups of SPICE components, but may be easier to describe in terms of high-level, behavioral equations. As a result, the behavior of these devices can be modeled using code modeling.

### 5.14.1 What is Code Modeling?

Code modeling is the behavioral modeling of devices whose governing equations are known.

**Note** This section serves as a basic guide to code modeling and includes helpful examples. However, code modeling is a complex process, so be aware that you need time and practice to gain proficiency.

A code model consists of a set of interface definitions and a C function implementation describing the device's behavior. The naming and location of these files is important. The model is created by combining two files (`Ifspec.ifs` and `Cfunc.mod`). The resulting file, which is given the same name as the folder containing its source files, is placed in the `codemodl` folder.



---

## 5.14.2 Creating a Code Model

➤ To create a code model:

1. Set up your environment variables for Microsoft Visual C++ by running `VcVars32.bat` (installed, by default, in the `c:/Program Files/DevStudio/Vc/Bin` folder). This step should be repeated each time you restart your computer and want to create a code model.
2. Create a folder under the `codemodl` folder (within the folder where you installed Multi-sim). Give the folder the same name as the model you are creating. For example, from the `codemodl` subdirectory, create the folder `C:/Program Files/EWB/codemodl/testmodl`.
3. Create an interface file called `Ifspec.ifs` in the folder you just created. The interface file describes the number and types of connections and parameters of the device. For example, create the file `C:/Program Files/EWB/codemodl/testmodl/Ifspec.ifs`.
4. Create an implementation file called `Cfunc.mod` in the subdirectory. The implementation file gives the equations that govern the behavior of the device. For example, create the file `C:/Program Files/EWB/codemodl/testmodl/Cfunc.mod`.
5. To compile the files into a dynamically-linked library (DLL), go to the `codemodl` folder and execute the command `MakeDev "folder"`, where "folder" is the name of the folder containing the `.ifs` and `.mod` files. For example, execute  

```
C:/Program Files/EWB/codemodl
cd
MakeDev "testing"
```

Errors and warnings may appear.
6. Place the resulting `.dll` file, which has the same name as the folder containing its source files, in the `codemodl` folder (that is, above the folder with the model's name). For example, the file created is called `C:/Program Files/EWB/codemodl/testing.dll`.

**Note** All of the code modeling files follow the C code syntax. For an overview of the general rules for the C code syntax, see any C code reference manual.

## 5.14.3 The Interface File (Ifspec.ifs)

The interface file sets out, in tables, the names used by the model, the electrical connections to the devices (ports), and the user-defined variables (parameters) that provide finer control over the behavior of the model. These tables are explained in this section, and examples given for each table. An example of an interface file is shown in “5.14.3.4 Example Interface File” on page 5-78. The interface file, along with the implementation file, needs to be compiled into a DLL to complete the code model.

### 5.14.3.1 Name Table

The model name, description text, and C implementation function name are defined in the name table. The model name must be the same as the subdirectory containing the code model files. It is recommended that the model name be eight characters.

The name table has the following syntax:

```
NAME_TABLE:
```

```
C_Function_Name:function_name  
Spice_ModelName:model_name  
Description:      "text"
```

where:

**function\_name** is a valid C identifier which is the name of the main entry point (function) for the code model. It may or may not be the same as the SPICE model name. To reduce the chance of name conflicts, we recommend you use the prefix "UCM\_" for user code model, or use a prefix based on your own initials. The following prefixes are used by the XSPICE simulator core and should not be used for user code models:

A2VERI	D_NOR	EW_RES	N1
A2VHDL	D_OPEN_C	EW_SCR	NCO
ADC_BRDG	D_OPEN_E	EW_SWITCH	ONESHOT
ASRC	D_OR	EW_VLT	POLY
ASWITCH	D_OSC	FTE	POT
BJT	D_PULLDN	GAIN	PPT
BSIM	D_PULLUP	HLP	PWL
CAP	D_RAM	HYST	R_2_V
CCCS	D_SOURCE	ICM	RDELAY
CCVS	D_SRFF	IDN	RES
CKT	D_SRLATC	ILIMIT	RGAIN
CLIMIT	D_STATE	IND	S_XFER
CM	D_TFF	INDUCTOR	SINE
CMETER	D_TRISTA	INP	SLEW
CORE	D_VERI	INT	SMP
CP	D_VHDL	IPC	SQUARE
CSW	D_WGEN	ISRC	SUMMER
D_2_R	D_XNOR	JFET	SW
D_AND	D_XOR	LCOUPLE	TRA
D_BUFFER	DAC_BRDG	LIMIT	TRIANGLE
D_CHIP	DAC_HIZ	LMETER	URC
D_DFF	DEV	MES	VCCS
D_DLATCH	DIO	MFB	VCVS
D_DT	DIVIDE	MIF	VERIZA
D_FDIV	ENH	MOS1	VHDL2A
D_INV	EVT	MOS2	VSRC
D_JKFF	EW_CAP	MOS3	XCAP
D_NAND	EW_IND	MULT	ZENER

**model\_name** is a valid SPICE identifier which will be used on SPICE deck .model records to refer to this code model. It may or may not be the same as the C function name.

**text** is a string describing the purpose and function of the code model.

For example:

NAME\_TABLE:

```
Spice_Model_Name: capacitor
C_Function_Name: cm_capacitor
Description: "Capacitor with voltage initial condition"
```



### 5.14.3.2 Port Table

The device ports are defined in the port tables. The port table has the following syntax:

PORT\_TABLE:

```
Port_Name:      name
Description:    text
Default_Type:   default
Allowed_Type:   [type type type]
Vector:         vector
Vector_Bounds: size
Direction:     dataflow
Null_Allowed:  null
```

where:

- name is a valid SPICE identifier giving the name of the port.
- text is a string describing the purpose and function of the port.
- default specifies the type used for the port when no type is explicitly specified. Must be one of the items listed in "type".
- type lists the allowed types to which the port can be connected, with names separated by commas or spaces (for example, [d, g, h]).

Type Name	Valid Directions	Description
d	in, out	digital
g	in, out	conductance (voltage input, current output)
gd	in, out	differential conductance (voltage input, current output)
h	in, out	resistance (current input, voltage output)
hd	in, out	differential resistance (current input, voltage output)
i	in, out	current
id	in, out	differential current
v	in, out	voltage
vd	in, out	differential voltage
vnam	in	current through named voltage source

- vector specifies whether or not port is a vector and can be considered a bus. Choose from:
  - yes - this port is a vector
  - no - this port is not a vector

size	for port that are vectors only, specifies upper and lower bounds on vector size. Lower bound specifies minimum number of elements, upper bound specifies maximum number of elements. For unconstrained range, or ports that are not a vector, use a hyphen (“-”).
data-flow	specifies the dataflow direction through the port. Choose from: <ul style="list-style-type: none"> <li>• in</li> <li>• out</li> <li>• inout</li> </ul>
null	specifies whether or not it is an error to leave the port unconnected. Choose from: <ul style="list-style-type: none"> <li>• yes - this port may be left unconnected</li> <li>• no - this port must be connected</li> </ul>

For example:

```

PORT_TABLE:

Port_Name:      cap
Description:    "capacitor terminals"
Direction:      inout
Default_Type:   hd
Allowed_Types:  [hd]
Vector:         no
Vector_Bounds:  -
Null_Allowed:   no

```

### 5.14.3.3 Parameter Table

The device parameters are defined in the parameter tables. The parameter table has the following syntax:

```

PARAMETER_TABLE:

Parameter_Name: name
Description:     text
Data_Type:      type
Vector:         vector
Vector_Bounds:  size
Default_Value:  default
Limits:         range
Null_Allowed:   null

where:

```

name	is a valid SPICE identifier which will be used on SPICE deck .model cards to refer to this parameter.
text	is a string describing the purpose and function of the parameter.

- type** is the parameter data type. Corresponds to the underlying C data type (e.g. "double"), not the conceptual type of the parameter (e.g. "voltage"). Choose from:
- boolean (if C data type is "Boolean\_t" with valid values MIF\_TRUE and MIF\_FALSE)
  - complex (if C data type is "Complex\_t" with double members real and imag)
  - int (if C data type is "int")
  - real (if C data type is "double")
  - string (if C data type is "char\*")
  - pointer (if C data type is "void\*")
- vector:** specifies whether parameter is vector or scalar. Choose from:
- yes - parameter is vector
  - no - parameter is scalar
- size:** for parameters that are vectors only, specifies upper and lower bounds on vector size. Lower bound specifies minimum number of elements, upper bound specifies maximum number of elements. For unconstrained range, or parameters that are not a vector, use a hyphen ("-"). Alternatively, specifies the name of the port whose vector size is to be used for this parameter.
- default** if Null\_Allowed is "yes", a default value to be used if the SPICE deck .model line does not supply a value for the parameter. Value must correspond to Data\_Type (numeric, boolean, complex or string literal).
- range** is a limited range of values (for "int" and "real" type parameters only).
- null** specify whether or not parameter is allowed to be null. Choose from:
- yes - the corresponding SPICE deck .model card may omit a value for this parameter, and the default value will be used or, if no default value, an undefined value will be passed to the code model
  - no - this parameter must have a value. XSPICE will flag an error if the corresponding SPICE deck .model card omits a value for this parameter.

For example:

PARAMETER\_TABLE:

Parameter_Name:	c	ic
Description:	"capacitance"	"voltage initial condition"
Data_Type:	real	real
Default_Value:	-	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

---

### 5.14.3.4 Example Interface File

Here is an example interface file:

```
/* =====
FILE      ifspec.ifs
MEMBER OF process XSPICE
Copyright 1991
Georgia Tech Research Corporation
Atlanta, Georgia 30332
All Rights Reserved

PROJECT A-8503

AUTHORS
    9/12/91  Bill Kuhn

MODIFICATIONS
    <date> <person name> <nature of modifications>

SUMMARY
This file contains the definition of a capacitor code model with volt-
age type initial conditions.

INTERFACES
None.

REFERENCED FILES
None.

NON-STANDARD FEATURES
None.
===== */

NAME_TABLE:

Spice_Model_Name:      capacitor
C_Function_Name:       cm_capacitor
Description:            "Capacitor with voltage initial condition"

PORT_TABLE:

Port_Name:             cap
Description:            "capacitor terminals"
Direction:             inout
```

```

Default_Type:      hd
Allowed_Types:    [hd]
Vector:           no
Vector_Bounds:    -
Null_Allowed:     no

```

PARAMETER\_TABLE:

Parameter_Name:	c	ic
Description:	"capacitance"	"voltage initial condition"
Data_Type:	real	real
Default_Value:	-	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

## 5.14.4 The Implementation File (Cfunc.mod)

At each simulation iteration for a circuit using the code model, Multisim's XSpice simulation engine calls the implementation file. An example of an implementation file is shown in "5.14.4.2 Example Implementation File" on page 5-88. The implementation file, along with the interface file, needs to be coupled into a DLL to complete the code model.

The code model function then generates the code-modeled device's output. This output is based on the following:

- The input that XSpice presents to the code model function.
- The state of the model, which is stored and returned by XSpice.

The implementation file includes one or more of the macros, shown in "5.14.4.1 Implementation File C Macros" on page 5-80, that provide the API (Application Programming Interface) between XSpice and the code model.

This section lists the macros from which you can select. The example file shown in "5.14.4.2 Example Implementation File" on page 5-88 gives an example of how to implement a macro. The implementation file, along with the interface file, needs to be compiled into a DLL to complete the code model.

# 5.14.4.1 Implementation File C Macros

## AC\_GAIN(outputname, inputname)

Type	Complex_t
Args	y[i], x[i]
Applies to	Analog code models only (event-driven or digital code models should do nothing during AC analysis).
Description	Assigns a value to this macro to specify the gain from outputname to inputname at the current frequency. The code model function is called once for each frequency point simulated.

## ANALYSIS

Type	enum
Args	none
Applies to	All code models, since their behavior typically changes depending on the type of analysis being performed, and this macro can be used to specify appropriate output macros.
Description	Returns the type of analysis being performed: MIF_AC for AC MIF_DC for DC operating point MID_TRAN for transient

## ARGS

Type	Mif_Private_t
Args	none
Applies to	All code models.
Description	The code model function's parameter list. Must be present and should not be modified.

**CALL\_TYPE**

<b>Type</b>	enum
<b>Args</b>	none
<b>Applies to</b>	Only code models that are mixed-mode (analog and event-driven or digital).
<b>Description</b>	If the analog portion of the simulator requested the code model call, set to MIF_ANALOG. If the digital portion of the simulator requested the code model call, set to MIF_EVENT. Needed if a code model's computation effort can be reduced based on the type of call made.

**INIT**

<b>Type</b>	Boolean_t
<b>Args</b>	none
<b>Applies to</b>	All code models.
<b>Description</b>	If this is the first call to the code model function during the current analysis or batch of analyses, set to MIF_TRUE. Otherwise, set to MIF_FALSE. Needed to let the code model perform startup activities (for example, allocated memory) at the start of simulation only.

**INPUT(inputname)**

<b>Type</b>	double or void *
<b>Args</b>	name [i]
<b>Applies to</b>	Analog/mixed-mode code models.
<b>Description</b>	Only analog inputs are allowed (for event-driven, use INPUT_STATE and INPUT_STRENGTH). Returns the value on the node or branch connected to inputname. Type/units of input value is specified when input type is specified in the lfspec.lfs file.

**INPUT\_STATE(inputname)**

<b>Type</b>	enum
<b>Args</b>	name [i]
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Only event-driven/digital inputs are allowed (for analog, use INPUT). Returns the digital value (ZERO, ONE or UNKNOWN) at node at inputname. When a single output is connected to that node, this will equal the value of the last output event. When multiple outputs are connected, conflict resolution is performed.

**INPUT\_STRENGTH(inputname)**

<b>Type</b>	enum
<b>Args</b>	name [i]
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Only event-driven/digital inputs are allowed (for analog, use INPUT). Returns the digital strength (STRONG, RESISTIVE, HI_IMPEDANCE or UNDETERMINED) of node at inputname. When a single output is connected to that node, this will equal the strength of the last output event. When multiple outputs are connected, conflict resolution is performed.

**INPUT\_TYPE(inputname)**

<b>Type</b>	char *
<b>Args</b>	name [i]
<b>Applies to</b>	All code models.
<b>Description</b>	Any inputs allowed. Returns the type string (i.e.: "v" for voltage, "i" for digital, "hd" for differential conductance, etc.) which describes the current usage of inputname. Needed to distinguish between "simulation time" usage of an input or output with more than one allowed type. For example, used for an input which has allowed types [v, i] and behaves differently when the input is voltage vs. current.



**LOAD(inputname)**

<b>Type</b>	double
<b>Args</b>	name [i]
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Only event-driven/digital inputs are allowed. Assign a value to LOAD to set the input load due to inputname on the connected node. The load is given as a capacitance (normalized to 1ohm resistance) which is summed with all the other loads on the event-driven node to yield the total delay of the node.

**MESSAGE(outputname)**

<b>Type</b>	char *
<b>Args</b>	name [i]
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Only event-driven/digital outputs are allowed. A message string to be placed on an event-driven node can be assigned to MESSAGE. Allows a code model to issue a message associated with a node.

**OUTPUT(outputname)**

<b>Type</b>	double or void *
<b>Args</b>	name [i]
<b>Applies to</b>	Analog/mixed-mode code models.
<b>Description</b>	Only analog outputs are allowed (for event-driven, use OUTPUT_STATE and OUTPUT_STRENGTH and OUTPUT_DELAY). Assigns a value to the node or branch connected to outputname. Type/units of output value specified when output type is specified in the Ifspec.Ifz file.

**OUTPUT\_CHANGED(outputname)**

Type	Boolean_t
Args	name [i]
Applies to	Digital/mixed-mode code models.
Description	Only event-driven/digital inputs are allowed. Set to MIF_TRUE by default. Assign MIF_FALSE to indicate no change on that output. Allows the code model to specify that the event-driven output did not change and thereby speed up simulation.

**OUTPUT\_DELAY(outputname)**

Type	none
Args	double
Applies to	Digital/mixed-mode code models.
Description	Only event-driven/digital inputs are allowed (for analog, use OUTPUT). Sets the delay after which the transition event specified by OUTPUT_STATE occurs.

**OUTPUT\_STATE(outputname)**

Type	none
Args	Digital_State_t
Applies to	Digital/mixed-mode code models.
Description	Only event-driven/digital outputs are allowed (for analog, use OUTPUT). Assigns the digital value (ZERO, ONE or UNKNOWN) to node at output-name by creating an event which is a transition to that value. When a single output is connected to that node, this will equal the value of the last output event. When multiple outputs are connected, conflict resolution is performed.

**OUTPUT\_STRENGTH(outputname)**

<b>Type</b>	none
<b>Args</b>	Digital_State_t
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Only event-driven/digital outputs are allowed (for analog, use OUTPUT). Assigns the digital strength (STRONG, RESISTIVE, HI_IMPEDANCE or UNDETERMINED) at node at outputname. When a single output is connected to that node, this will equal the strength of the last output event. When multiple outputs are connected, conflict resolution is performed.

**OUTPUT\_TYPE(inputname)**

<b>Type</b>	char *
<b>Args</b>	name [i]
<b>Applies to</b>	Digital/mixed-mode code models.
<b>Description</b>	Any output allowed. Returns the type string (i.e.: "v" for voltage, "i" for digital, "hd" for differential conductance, etc.) which describes the current usage of outputname. Needed to distinguish between "simulation time" usage of an input or output with more than one allowed type. For example, used for an input which has allowed types [v, i] and behaves differently when the input is voltage vs. current.

**PARAM(paramname)**

<b>Type</b>	CD
<b>Args</b>	name [i]
<b>Applies to</b>	Any code model.
<b>Description</b>	Applies to all parameters. Returns the value paramname. Needed to access model parameters specified in the netlist.

**PARAM\_NULL(paramname)**

Type	Boolean_t
Args	name [i]
Applies to	Only parameters allowed to be unspecified (Null allowed in the param table of the lfspec.lfs file is yes).
Description	Returns MIF_TRUE if paramname was not specified in the netlist and MIF_FALSE if it was specified. Allows the code model to tell if a parameter value equals its default because the default value was actually specified.

**PARAM\_SIZE(paramname)**

Type	int
Args	name
Applies to	Vector type parameters only.
Description	Returns the number of elements in a vector type parameter. Needed to iterate over the vector parameter if the number of vector elements is not fixed.

**PARTIAL**

Type	double
Args	y[i], x[i]
Applies to	Analog/mixed-mode code models.
Description	Partial derivative of output y with respect to input x.

**PORT\_NULL**

<b>Type</b>	Boolean_t
<b>Args</b>	name[i]
<b>Applies to</b>	Any code model.
<b>Description</b>	Has this port been specified as unconnected?

**PORT\_SIZE**

<b>Type</b>	int
<b>Args</b>	name
<b>Applies to</b>	Any code model.
<b>Description</b>	Size of port vector.

**RAD\_FREQ**

<b>Type</b>	double
<b>Args</b>	<none>
<b>Applies to</b>	Analog/mixed-mode code models.
<b>Description</b>	Current analysis frequency in radians per second.

**T (<n>)**

<b>Type</b>	double
<b>Args</b>	<none>
<b>Applies to</b>	All code models.
<b>Description</b>	History of the previous nth analysis time (TIME = T[0]). Maximum of 8.

TEMPERATURE

Type	double
Args	<none>
Applies to	All code models.
Description	Current analysis temperature.

TIME

Type	double
Args	<none>
Applies to	All code models.
Description	Current analysis time (same as T[0]).

5.14.4.2 Example Implementation File

Here is an example implementation file:

```
/* =====  
FILE      cfunc.mod  
  
MEMBER OF process XSPICE  
  
Copyright 1991  
Georgia Tech Research Corporation  
Atlanta, Georgia 30332  
All Rights Reserved  
  
PROJECT A-8503  
  
AUTHORS  
    9/12/91  Bill Kuhn  
  
MODIFICATIONS  
    <date> <person name> <nature of modifications>
```

#### SUMMARY

This file contains the definition of a capacitor code model with voltage type initial conditions.

#### INTERFACES

cm\_capacitor()

#### REFERENCED FILES

None.

#### NON-STANDARD FEATURES

None.

```
===== */

#define VC 0

void cm_capacitor (ARGS)
{
    Complex_t    ac_gain;
    double        partial;
    double        ramp_factor;
    double        *vc;

    /* Get the ramp factor from the .option ramptime */
    ramp_factor = cm_analog_ramp_factor(MIF_INSTANCE);

    /* Initialize/access instance specific storage for capacitor voltage */
    if (INIT) {
        cm_analog_alloc(MIF_INSTANCE, VC, sizeof(double));
        vc = cm_analog_get_ptr(MIF_INSTANCE, VC, 0);
        *vc = PARAM(ic) * cm_analog_ramp_factor(MIF_INSTANCE);
    }
    else {
        vc = cm_analog_get_ptr(MIF_INSTANCE, VC, 0);
    }

    /* Compute the output */
    if (ANALYSIS == DC) {
        OUTPUT(cap) = PARAM(ic) * ramp_factor;
        PARTIAL(cap, cap) = 0.0;
    }
    else if (ANALYSIS == AC) {
```

---

```

        ac_gain.real = 0.0;
        ac_gain.imag = -1.0 / RAD_FREQ / PARAM(c);
        AC_GAIN(cap, cap) = ac_gain;
    }
    else if (ANALYSIS == TRANSIENT) {
        if (ramp_factor < 1.0) {
            *vc = PARAM(ic) * ramp_factor;
            OUTPUT(cap) = *vc;
            PARTIAL(cap, cap) = 0.0;
        }
        else {
            cm_analog_integrate(MIF_INSTANCE, INPUT(cap) / PARAM(c),
                                vc, &partial);
            partial /= PARAM(c);
            OUTPUT(cap) = *vc;
            PARTIAL(cap, cap) = partial;
        }
    }
}

```

### 5.14.4.3 Additional Example ifspec File

```

/* =====
FILE    ifspec.ifs

MEMBER OF process XSPICE

Copyright 1991
Georgia Tech Research Corporation
Atlanta, Georgia 30332
All Rights Reserved

PROJECT A-8503

AUTHORS
9/12/91  Bill Kuhn

MODIFICATIONS
<date> <person name> <nature of modifications>

SUMMARY
This file contains the definition of a capacitor code model with volt-
age type initial conditions.

INTERFACES

```



```

None.

REFERENCED FILES
None.

NON-STANDARD FEATURES
None.

===== */

NAME_TABLE:

Spice_Model_Name:      capacitor
C_Function_Name:       cm_capacitor
Description:            "Capacitor with voltage initial condition"

PORT_TABLE:

Port_Name:             cap
Description:            "capacitor terminals"
Direction:             inout
Default_Type:          hd
Allowed_Types:         [hd]
Vector:                no
Vector_Bounds:         -
Null_Allowed:          no

PARAMETER_TABLE:

Parameter_Name:        c                ic
Description:            "capacitance"    "voltage initial condition"
Data_Type:              real             real
Default_Value:          -                0.0
Limits:                 -                -
Vector:                 no               no
Vector_Bounds:          -                -
Null_Allowed:           no               no

/* $Id: cfunc.tpl,v 1.1 91/03/18 19:01:04 bill Exp $ */
/
*.....1.....2.....3.....4.....5.....6.....7
.....8
=====

```

---

Additional Example cfunc File  
FILE d\_jkff/cfunc.mod

Copyright 1991  
Georgia Tech Research Corporation, Atlanta, Ga. 30332  
All Rights Reserved

PROJECT A-8503-405

#### AUTHORS

21 Jun 1991        Jeffrey P. Murray

#### MODIFICATIONS

12 Aug 1991        Jeffrey P. Murray  
30 Sep 1991        Jeffrey P. Murray  
29 Jan 1992        Jeffrey P. Murray

#### SUMMARY

This file contains the functional description of the d\_jkff code model.

#### INTERFACES

FILE	ROUTINE CALLED
CMutil.c	void cm_toggle_bit();

CMevt.c	void *cm_event_alloc(MIF_INSTANCE)
	void *cm_event_get_ptr(MIF_INSTANCE)

#### REFERENCED FILES

Inputs from and outputs to ARGS structure.

#### NON-STANDARD FEATURES

NONE

```
=====*/  
  
/*== INCLUDE FILES =====*/  
  
/*== CONSTANTS =====*/  
  
/*== MACROS =====*/
```

```

/*=== LOCAL VARIABLES & TYPEDEFS =====*/

/*=== FUNCTION PROTOTYPE DEFINITIONS ===*/

/*=====

FUNCTION cm_toggle_bit()

AUTHORS
27 Sept 1991      Jeffrey P. Murray

MODIFICATIONS
NONE

SUMMARY
Alters the state of a passed digital variable to its
    complement. Thus, a ONE changes to a ZERO. A ZERO changes
    to a ONE, and an UNKNOWN remains unchanged.

INTERFACES
FILE              ROUTINE CALLED

    N/A              N/A

RETURNED VALUE
No returned value. Passed pointer to variable is used
    to redefine the variable value.

GLOBAL VARIABLES
NONE

NON-STANDARD FEATURES
NONE

=====*/

/*=== CM_TOGGLE_BIT ROUTINE ===*/

static void cm_toggle_bit(Digital_State_t *bit)
{
    /* Toggle bit from ONE to ZERO or vice versa, unless the

```

---

```

        bit value is UNKNOWN. In the latter case, return
        without changing the bit value.                                */

    if ( UNKNOWN != *bit ) {
        if ( ONE == *bit ) {
            *bit = ZERO;
        }
        else {
            *bit = ONE;
        }
    }
}

/*=====

FUNCTION cm_eval_jk_result

AUTHORS
30 Sept 1991      Jeffrey P. Murray

MODIFICATIONS
NONE

SUMMARY
Evaluates the J and K input states, plus the last state of
the flip flop, and returns the expected output value.

INTERFACES
FILE              ROUTINE CALLED
CMutil.c          void cm_toggle_bit();

RETURNED VALUE

    A Digital_State_t.

GLOBAL VARIABLES
NONE

NON-STANDARD FEATURES
NONE

=====*/

/*=== CM_EVAL_JK_RESULT ROUTINE ===*/

```

```

static Digital_State_t cm_eval_jk_result(Digital_State_t j_input,
    Digital_State_t k_input,
    Digital_State_t old_output)
{
    Digital_State_t    output; /* returned output value */

    switch (j_input) {

    case ZERO:
        switch (k_input) {
            case ZERO:
                output = old_output;
                break;
            case ONE:
                output = ZERO;
                break;
            case UNKNOWN:
                output = UNKNOWN;
                break;
        }
        break;

    case ONE:
        switch (k_input) {
            case ZERO:
                output = ONE;
                break;
            case ONE:
                output = old_output;
                cm_toggle_bit(&output);
                break;
            case UNKNOWN:
                output = UNKNOWN;
                break;
        }
        break;

    case UNKNOWN:
        output = UNKNOWN;
        break;
    }

    return output;
}

```

---

```

}

/*=====

FUNCTION cm_d_jkff()

AUTHORS
21 Jun 1991      Jeffrey P. Murray

MODIFICATIONS
12 Aug 1991      Jeffrey P. Murray
 30 Sep 1991      Jeffrey P. Murray
 29 Jan 1992      Jeffrey P. Murray

SUMMARY
This function implements the d_jkff code model.

INTERFACES
FILE              ROUTINE CALLED

    CMutil.c              void cm_toggle_bit();

    CMevt.c               void *cm_event_alloc(MIF_INSTANCE)
                          void *cm_event_get_ptr(MIF_INSTANCE)

RETURNED VALUE
Returns inputs and outputs via ARGS structure.

GLOBAL VARIABLES
NONE

NON-STANDARD FEATURES
NONE

=====*/

/*=== CM_D_JKFF ROUTINE ===*/

/*****
*      The following is the model for the      *
*      digital jk-type flip flop for the      *
*      ATE SSE Version 2.0 system.            *
*      *                                       *
*      Created 6/21/91                      J.P.Murray      *
*****/

```

```

*****/

void cm_d_jkff(ARGS)

{
/* generic loop counter index */
int i;
/* current clk value */
Digital_State_t*clk,
/* previous clk value */
*clk_old,
/* current set value for dff */
*set,
/* previous set value for dff */
*set_old,
/* current reset value for dff */
*reset,
/* previous reset value for dff */
*reset_old,
/* current output for dff */
*out,
/* previous output for dff */
*out_old,
/* current j input value */
j_input,
/* current k input value */
k_input,
/* temp storage for state values */
temp;

/** Setup required state variables **/

if(INIT) { /* initial pass */

/* allocate storage */
cm_event_alloc(MIF_INSTANCE,0,sizeof(Digital_State_t));
cm_event_alloc(MIF_INSTANCE,1,sizeof(Digital_State_t));
cm_event_alloc(MIF_INSTANCE,2,sizeof(Digital_State_t));
cm_event_alloc(MIF_INSTANCE,3,sizeof(Digital_State_t));

```

```

        clk = clk_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,0,0);
        set = set_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,1,0);
        reset = reset_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,2,0);
        out = out_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,3,0);

        /* declare load values */
        LOAD(j) = PARAM(jk_load);
        LOAD(k) = PARAM(jk_load);
        LOAD(clk) = PARAM(clk_load);
        if ( !PORT_NULL(set) ) {
            LOAD(set) = PARAM(set_load);
        }
        if ( !PORT_NULL(reset) ) {
            LOAD(reset) = PARAM(reset_load);
        }
    }
    else { /* Retrieve previous values */

        /* retrieve storage for the outputs */
        clk = (Digital_State_t *) cm_event_get_ptr(MIF_INSTANCE,0,0);
        clk_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,0,1);
        set = (Digital_State_t *) cm_event_get_ptr(MIF_INSTANCE,1,0);
        set_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,1,1);
        reset = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,2,0);
        reset_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,2,1);
        out = (Digital_State_t *) cm_event_get_ptr(MIF_INSTANCE,3,0);
        out_old = (Digital_State_t *)
cm_event_get_ptr(MIF_INSTANCE,3,1);
    }

    /****** load current input values if set or reset
are not connected, set to zero... *****/
    *clk = INPUT_STATE(clk);
    if ( PORT_NULL(set) ) {
        *set = *set_old = ZERO;
    }
    else {

```



```

        *set = INPUT_STATE(set);
    }
    if ( PORT_NULL(reset) ) {
        *reset = *reset_old = ZERO;
    }
    else {
        *reset = INPUT_STATE(reset);
    }

    /***** Determine analysis type and output appropriate values
    *****/

    /***** DC analysis...output w/o delays *****/
    if (0.0 == TIME) {
        temp = PARAM(ic);

        /** Modify output if set or reset lines are active **/
        if ( (*set==ONE) && (*reset==ZERO) ) temp = ONE;
        if ( (*set==ZERO) && (*reset==ONE) ) temp = ZERO;
        if ( (*set==ONE) && (*reset==ONE) ) temp = UNKNOWN;

        *out = *out_old = temp;

        if ( !PORT_NULL(out) ) {
            OUTPUT_STATE(out) = temp;
        }

        cm_toggle_bit(&temp);

        if ( !PORT_NULL(Nout) ) {
            OUTPUT_STATE(Nout) = temp;
        }
    }

    else {          /***** Transient Analysis *****/

        /***** Find input that has changed... *****/

        /**** Test set value for change ****/

        /* either set or set release */
        if ( *set != *set_old ) { switch ( *set ) {

```

---

```

        case ONE:
            if ( ONE != *reset) {

                /*set will change output */
                if (*out_old != ONE) {

                    /* output goes to ONE */
                    *out = ONE;

                    if ( !PORT_NULL(out) ) {
                        OUTPUT_STATE(out) = ONE;
                        OUTPUT_DELAY(out) = PARAM(set_delay);
                    }
                    if ( !PORT_NULL(Nout) ) {
                        OUTPUT_STATE(Nout) = ZERO;
                        OUTPUT_DELAY(Nout) = PARAM(set_delay);
                    }
                }
            }
            else {

                /* output already set */
                *out = *out_old;

                if ( !PORT_NULL(out) ) {
                    OUTPUT_CHANGED(out) = FALSE;
                }
                if ( !PORT_NULL(Nout) ) {
                    OUTPUT_CHANGED(Nout) = FALSE;
                }
            }
        }
    }
    else {
        if (*out_old != UNKNOWN) { /* set will change out-
put */

            /* output goes to UNKNOWN */
            *out = UNKNOWN;

            if ( !PORT_NULL(out) ) {
                OUTPUT_STATE(out) = UNKNOWN;
                OUTPUT_DELAY(out) = PARAM(set_delay);
            }
            if ( !PORT_NULL(Nout) ) {
                OUTPUT_STATE(Nout) = UNKNOWN;
                OUTPUT_DELAY(Nout) = PARAM(set_delay);
            }
        }
    }
    else {

```

```

        *out = *out_old;      /* output already unknown
*/
        if ( !PORT_NULL(out) ) {
            OUTPUT_CHANGED(out) = FALSE;
        }
        if ( !PORT_NULL(Nout) ) {
            OUTPUT_CHANGED(Nout) = FALSE;
        }
    }
}
break;

case ZERO:
    if ( ONE != *reset) {
        /* output remains at current value */
        *out = *out_old;
        if ( !PORT_NULL(out) ) {
            OUTPUT_CHANGED(out) = FALSE;
        }
        if ( !PORT_NULL(Nout) ) {
            OUTPUT_CHANGED(Nout) = FALSE;
        }
    }
    else {
        if (*out_old != ZERO) { /* set will change output
*/
            /* output returns to reset condition */
            *out = ZERO;

            if ( !PORT_NULL(out) ) {
                OUTPUT_STATE(out) = ZERO;
                OUTPUT_DELAY(out) = PARAM(set_delay);
            }
            if ( !PORT_NULL(Nout) ) {
                OUTPUT_STATE(Nout) = ONE;
                OUTPUT_DELAY(Nout) = PARAM(set_delay);
            }
        }
        else {
            *out = *out_old;      /* output already reset */
            if ( !PORT_NULL(out) ) {
                OUTPUT_CHANGED(out) = FALSE;
            }
            if ( !PORT_NULL(Nout) ) {
                OUTPUT_CHANGED(Nout) = FALSE;
            }
        }
    }
}

```

---

```
    }  
  }  
  break;  
  
case UNKNOWN:  
  
  if ( ONE == *reset ) {  
    /* output goes to ZERO */  
    *out = ZERO;  
  
    if ( !PORT_NULL(out) ) {  
      OUTPUT_STATE(out) = ZERO;  
      OUTPUT_DELAY(out) = PARAM(set_delay);  
    }  
  }  
}
```

# Chapter 6

## Instruments

### 6.1 About this Chapter

This chapter explains how to use the various virtual instruments provided as part of Multisim 2001. It explains both the general procedures for attaching and configuring the instruments, and the specific steps in using each instrument.

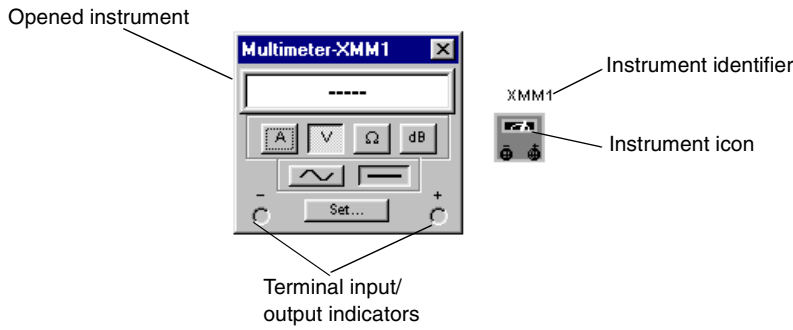


Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 6.2 Introduction to the Multisim Instruments

Multisim provides a number of virtual instruments. You use these instruments to measure the behavior of your circuits. These instruments are set, used and read just like their real-world equivalents. They look and feel just like the instruments you've seen and used in a lab. Using virtual instruments is one of the best and easiest ways of examining your circuit's behavior and showing the results of a simulation. These instruments can be placed in any level of circuit or subcircuit (for information on subcircuits, see "3.11 Subcircuits and Hierarchy" on page 3-31) but they are active only for the currently active circuit or subcircuit.

Virtual instruments have two views: the instrument icon you attach to your circuit, and the opened instrument, where you set the instrument's controls and display screen.

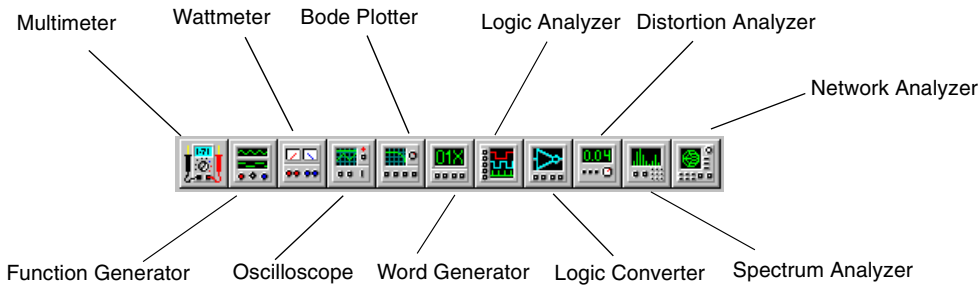


The instrument's icon indicates how the instrument is connected into the circuit. Once simulated, a black dot inside the terminal input/output indicators shows that the instrument is connected to a node.

➤ To add an instrument to a circuit:



1. By default the instruments toolbar is displayed on the workspace. If the instrument toolbar is not displayed, click the Instruments button on the Design Bar. The Instruments toolbar appears, including one button for each instrument.



2. From the Instruments toolbar, click the button of the instrument you want to use.
3. Move the cursor to the location on the circuit window where you want to place the instrument and click. The instrument is placed with the connections landing on the grid.

The instrument icon and the instrument identifier appear. The instrument identifier identifies the type of instrument and its instance. For example, the first multimeter you place on a circuit is called “XMM1”, the second is “XMM2”, and so on. This numbering is unique within each circuit. That is, if you create a second circuit, the first multimeter placed in it is “XMM1”, and so on.

**Note** Not all Multisim editions support multiple instances of an instrument.

4. To wire the instrument into the circuit, click on a terminal on the instrument’s icon and drag a wire to the desired location in the circuit (a pin, wire, or junction). All the rules for component wiring, described in Chapter 3, “Schematic Capture”, apply to instruments as well.

**Note** To change the color of the Instrument icon, right-click on it and choose **Color** from the pop-up menu that appears. Choose the desired color and click **OK**.

For educational purposes, you may wish to prevent students from being able to place instruments on the circuit window. You can restrict students’ access to the Instruments tool bar by using restrictions, as described in “13.3 Using Restrictions” on page 13-4.

➤ To use the instrument:

1. To view and modify an instrument’s controls, double-click its icon. The instrument controls appear. Make any necessary changes to the control settings, just as you would on their real-world equivalents. The control settings are different for each instrument, so if you are unfamiliar with them or need instruction, refer to the section on that particular instrument in this chapter.

It is *critical* that the control settings be appropriate for your circuit. If the settings are incorrect, this may cause the simulation results to appear incorrect or difficult to read.

**Note** Not all areas of the open instrument are modifiable. A hand appears when your cursor is on a control that can be modified.



2. To “activate” the circuit, click the Simulate button on the Design Bar and choose **Run** from the pop-up menu that appears. Multisim begins to simulate the circuit’s behavior and the signals, as measured at the points to which you have connected the instrument, are displayed.

The simulation results depend on the circuit’s construction. During simulation, messages about the simulation results and any problems with the simulation are written to the simulation error log/audit trail. The error log/audit trail appears automatically when you stop the simulation. If you want to keep an eye on the progress of the simulation, you can display the error log/audit trail during simulation. To display it, from the **View** menu choose **Show Simulation Error Log/Audit Trail**. For more detailed information about simulation, see Chapter 7, “Simulation”.

While the circuit is activated, you can adjust instrument settings. You cannot change the circuit by changing values (except for variable components), or perform any schematic capture functions such as rotate or component replacement.

- To pause or resume the simulation, click the Simulate button on the Design Bar and choose **Pause** from the pop-up menu that appears. The simulation is paused.
- To stop the simulation, click the Simulate button on the Design Bar and choose **Run** from the pop-up menu that appears. The simulation ends, with the final results shown on the instrument face and in the audit trail.

You can also run, stop, pause, or resume using commands from the **Simulate** menu.

## 6.3 Working with Multiple Instruments

A single circuit can have multiple instruments attached to it, including (for some editions) multiple instances of the same instrument. In addition, each circuit window can have its own set of instruments. Setting up many different instruments or multiple instances of one instrument is done in exactly the same way as setting up one instrument.

Instruments that sample for an amount of time cause a transient analysis to be run. If you use multiples of such instruments, only one transient analysis is run. The settings of this analysis are derived from considering all the concurrent instruments and choosing settings that will satisfy each. For example, if you have two oscilloscopes with two different time-bases (resolutions), Multisim uses the time-base of the oscilloscope with the smallest time-base (highest resolution). As a result, both instruments will sample at a higher resolution than they would individually.

The results from each instrument are recorded separately in the error log/audit trail.



## 6.4 Default Instrument Analysis Settings

Multisim lets you set default settings for instruments that are based on a transient analysis (such as the oscilloscope, spectrum analyzer and logic analyzer).

- To set the default instrument settings:
  1. Choose **Simulate/Default Instrument Settings**. A streamlined Default Instrument Settings screen appears displaying the most commonly used functions.
  2. Click **More**. The full screen appears:

Set initial conditions: Zero, User-Defined, Calculate DC Operating Point, or Automatically Determine Initial Conditions.

Start time of transient analysis must be greater than or equal to 0 and less than End time.

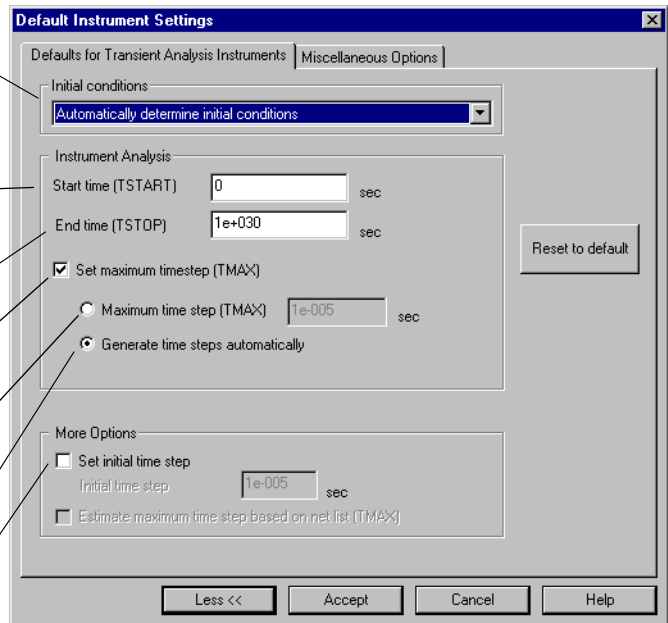
End time of transient analysis must be greater than Start time.

Enable to enter the maximum time step the simulation can handle.

Enable to manually set time steps.

Enable to generate time steps automatically.

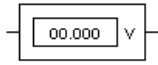
Enable to set a time interval for simulation output and graphing.



3. Enter settings as desired and click **Accept**, or click **Cancel** to cancel. These settings will be in effect the next time you run a simulation.

You can control many aspects of the simulation, such as resetting error tolerances, selecting simulation techniques, and viewing the results. The options you choose will determine the efficiency of the simulation. See “8.28 Analysis Options” on page 8-78 for details on the analysis options and their default values. You set these options through the Miscellaneous Options tab.

## 6.5 Voltmeter



The voltmeter offers advantages over the multimeter for measuring voltage in a circuit. The advantage of using the voltmeter is that you can use an unlimited number of voltmeters in a circuit and you can rotate their terminals to suit your layout. The side with the heavier border is the negative terminal.

### 6.5.1 Resistance (1.0 $\Omega$ - 999.99 T $\Omega$ )

The voltmeter is preset to a very high resistance (1 M $\Omega$ (+)) which generally has no effect on a circuit. If you are testing a circuit that itself has very high resistance, you may want to increase the voltmeter's resistance to get a more accurate reading. (However, using a voltmeter with very high resistance in a low-resistance circuit may result in a mathematical round-off error.)

### 6.5.2 Mode (DC or AC)

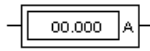
The voltmeter can measure DC or AC voltage. In DC mode, any AC component of the signal is eliminated so that only the DC component of the signal is measured. In AC mode, any DC component is eliminated so that only the AC component is measured. When set to AC, the voltmeter displays the root-mean-square (RMS) value of the signal.

### 6.5.3 Connecting a Voltmeter

Connect the voltmeter in parallel with the load, attaching the probes to connectors on either side of the load you want to measure. When a circuit is activated and its behavior is simulated, the voltmeter displays the voltage across the test points. (The voltmeter may also display interim voltages before the final steady-state voltage is reached.)

**Note** If a voltmeter is moved after the circuit has been simulated, activate the circuit again to get a reading.

## 6.6 Ammeter



The ammeter offers advantages over the multimeter for measuring current in a circuit. The advantage of using the ammeter is that you can use an unlimited number of ammeters in a circuit and you can rotate their terminals to suit your layout. The side with the heavier border is the negative terminal.

### 6.6.1 Resistance (1.0 p $\Omega$ - 999.99 $\Omega$ )

The ammeter's resistance is preset to 1 m $\Omega$ , which presents little resistance to a circuit. If you are testing a circuit that has low resistance, you can lower the ammeter's resistance even further to get a more precise measurement. (However, using an ammeter with very low resistance in a high-resistance circuit may result in a mathematical round-off error.)

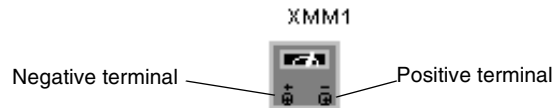
### 6.6.2 Mode (DC or AC)

The ammeter is preset to DC mode, which measures only the DC component of a signal. If you want to measure the current from an AC source, change the mode to AC. When set to AC, the ammeter displays the root-mean-square (RMS) value of the alternating signal.

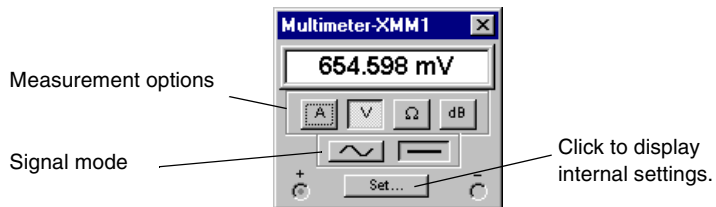
### 6.6.3 Connecting an Ammeter

Like a real ammeter, the simulated ammeter must be connected in series at nodes you want to measure. The negative terminal is on the side with the heavy border. If an ammeter is moved after the circuit has been simulated, activate the circuit again to get a reading.

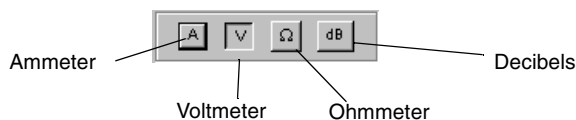
## 6.7 Multimeter



Use the multimeter to measure AC or DC voltage or current, and resistance or decibel loss between two nodes in a circuit. The multimeter is auto-ranging, so a measurement range does not need to be specified. Its internal resistance and current are preset to near-ideal values, which can be changed (see “6.6.3 Internal Settings” on page 6-11).



### 6.7.1 Measurement Options

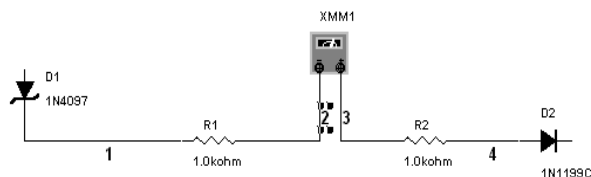




### 6.7.1.1 Ammeter

This option measures current flowing through the circuit in a branch between two nodes.

Insert the multimeter in series with the load to measure current flow, just like a real ammeter (as shown in diagram below).

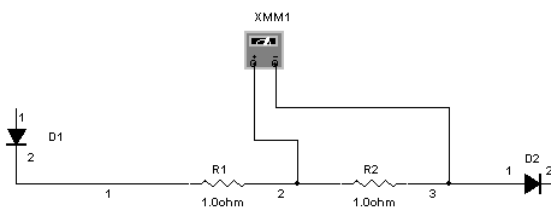


- To measure current at another node in the circuit, connect another multimeter in series at that load and activate the circuit again. When used as an ammeter, the multimeter's internal resistance is very low (1n Ohm).
- To change the resistance, click **Set**. See “6.6.3 Internal Settings” on page 6-11 for details.



### 6.7.1.2 Voltmeter

This option measures voltage between two nodes. Select **V** and attach the voltmeter's probes in parallel with the load (as shown in diagram below).

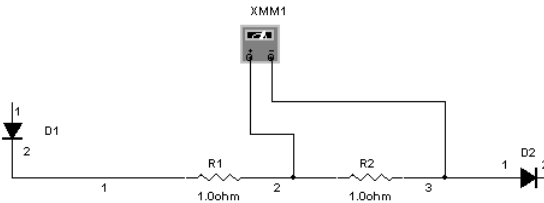


When used as a voltmeter, the multimeter has a high internal resistance of 1 mohm, which can be changed by clicking **Set**. See “6.6.3 Internal Settings” on page 6-11 for details.

### 6.7.1.3 Ohmmeter



This option measures resistance between two nodes. The nodes and everything that lies between them are referred to as the “component network”. To measure the resistance, select this option and attach the multimeter’s probes in parallel with the component network (as shown in the diagram below).



To get an accurate measurement, make sure that:

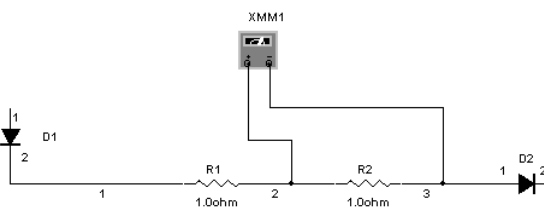
- there is no source in the component network
- the component or component network is grounded
- the multimeter is set to DC (for more details, see “6.6.2 Signal Mode (AC or DC)” on page 6-11)
- there is nothing else in parallel with the component or component network.

The ohmmeter generates a 1-mA current, which can be changed by clicking **Set**. See “6.6.3 Internal Settings” on page 6-11 for details. If the ohmmeter has been attached to different nodes, re-activate the circuit to get a reading.

### 6.7.1.4 Decibels



This option measures decibel voltage loss between two nodes in a circuit. To measure the decibels, select this option and attach the multimeter’s probes in parallel with the load (as shown in diagram below).



The Decibel standard for calculating dB is preset to 1 V, but can be changed by clicking **Set**. See “6.6.3 Internal Settings” on page 6-11 for details. Decibel loss is calculated as follows:

$$dB = 20 * \log_{10} \left( \frac{V_{out}}{V_{in}} \right)$$

## 6.7.2 Signal Mode (AC or DC)



The **Sine-wave** button measures the root-mean-square (RMS) voltage or current of an AC signal. Any DC component of the signal will be eliminated, so only the AC component of the signal is measured.



The **Straight-wave** button measures the current or voltage value of a DC signal.

**Note** To measure the RMS voltage of a circuit with both AC and DC components, connect an AC voltmeter as well as a “DC” voltmeter across the appropriate nodes and measure the AC and DC voltage.

The following formula can be used to calculate RMS voltage when both AC and DC components are in the circuit. Be advised that this is not a universal formula and should be used in conjunction with Multisim only.

$$\text{RMS voltage} = \sqrt{(V_{dc}^2 + V_{ac}^2)}$$

## 6.7.3 Internal Settings

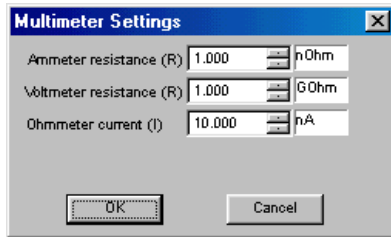
Ideal meters have no effect on the circuit being measured. An ideal voltmeter would have infinite resistance, so no current could flow through it while it is attached to a circuit. An ideal ammeter would present no resistance to a circuit. Real meters do not achieve this ideal, so their readings will very closely match theoretical, calculated values for a circuit, but never with absolute precision.

The multimeter in Multisim, like a real multimeter, is nearly ideal. It uses very small and very large numbers that approximate zero and infinity to calculate near-ideal values for the circuit. For special cases, however, the meter’s behavior can be changed by changing these values used to model its effect on the circuit. (The values must be higher than 0.)

For example, if testing the voltage of a circuit with very high resistance, increase the voltmeter’s resistance. If measuring the current of a circuit with very low resistance, decrease the ammeter’s resistance even further.

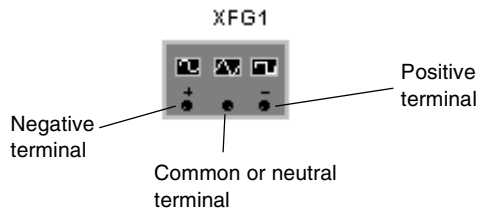
**Note** Very low ammeter resistance in a high-resistance circuit may result in a mathematical roundoff error.

- To display the default internal settings:
1. Click **Set**. The Multimeter Settings screen appears.



2. Change the desired options.
3. To save your changes, click **OK**. To cancel them, click **Cancel**.

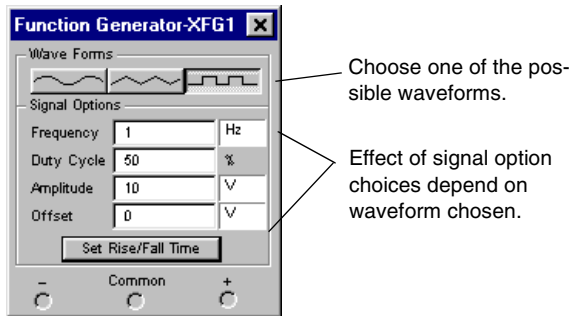
## 6.8 Function Generator



The function generator is a voltage source that supplies sine, triangular or square waves. It provides a convenient and realistic way to supply stimulus signals to a circuit. The waveform can be changed and its frequency, amplitude, duty cycle and DC offset can be controlled. The function generator's frequency range is great enough to produce conventional AC as well as audio- and radio-frequency signals.



The function generator has three terminals through which waveforms can be applied to a circuit. The common terminal provides a reference level for the signal.



- To reference a signal from ground, connect the common terminal to the ground component. The positive terminal (+) provides a waveform in the positive direction from the neutral common terminal. The negative terminal (-) provides a waveform in the negative direction.

## 6.8.1 Waveform Selection

You can select three different types of waveforms as the output.

- To select the waveform, click the Sine-, Triangular- or Square-wave button.

## 6.8.2 Signal Options

### 6.8.2.1 Frequency (1Hz — 999 MHz)

This setting determines the number of cycles per second the function generator generates.

### 6.8.2.2 Duty Cycle (1% — 99%)

This setting determines the ratio of on-period to off-period. It affects the shape of triangular and square waves as shown below. A sine wave is not affected by the duty cycle setting.

### 6.8.2.3 Amplitude (1 $\mu$ V — 999 kV)

This setting controls the signal's voltage, measured from its DC level to its peak. If the leads are connected to the common and either the positive or the negative terminal, the wave's peak-to-peak measurement is twice its amplitude. If the output comes from the positive and negative terminals, the wave's peak-to-peak measurement is four times its amplitude.

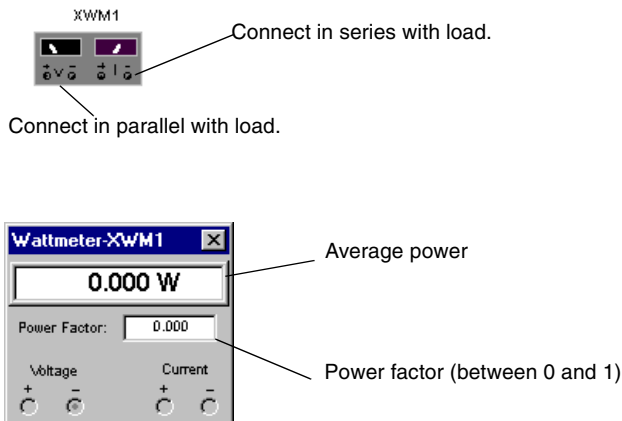
### 6.8.2.4 Offset (-999 kV and 999 kV)

This option controls the DC level about which the alternating signal varies. An offset of 0 positions the waveform along the oscilloscope's x-axis (provided its Y POS setting is 0). A positive value shifts the DC level upward, while a negative value shifts it downward. Offset uses the units set for Amplitude.

## 6.8.3 Rise Time

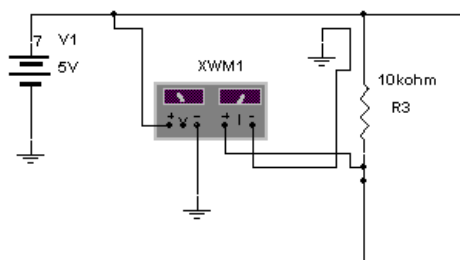
This option sets the time over which the square waveform is built (and, therefore, the angle of the waveform). Only available for square waveforms.

## 6.9 Wattmeter

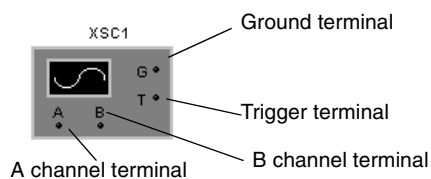


The wattmeter measures power. It is used to measure the magnitude of the active power, that is, the product of the voltage difference and the current flowing through the current terminals in a circuit. The results are shown in watts. The wattmeter also displays the power factor, calculated by measuring the difference between the voltages and the current, and multiplying

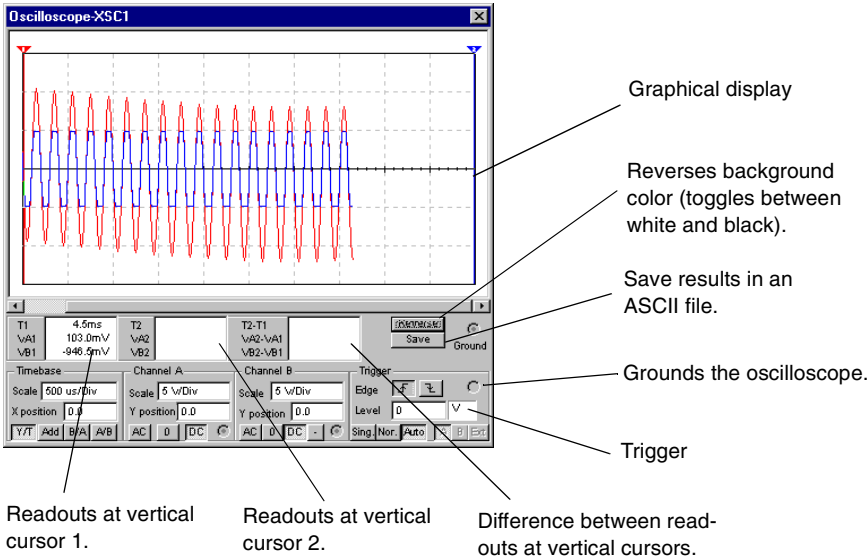
them together. The power factor is the cosine of the phase angle before the voltage and current.



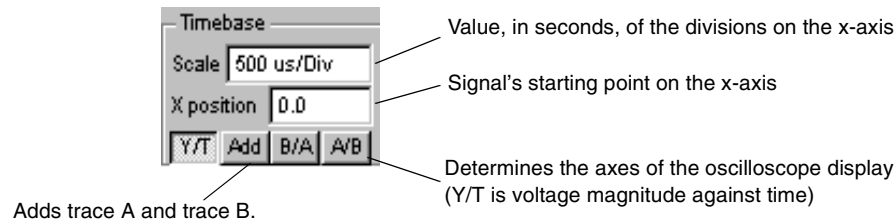
## 6.10 Oscilloscope



The dual-channel oscilloscope displays the magnitude and frequency variations of electronic signals. It can provide a graph of the strength of one or two signals over time, or allow comparison of one waveform to another.



### 6.10.1 Time Base (0.1 ns/Div — 1s/Div)



The time base setting controls the scale of the oscilloscope's horizontal or x-axis when comparing magnitude against time (Y/T).

- To get a readable display, adjust the time base in inverse proportion to the frequency setting on the function generator or AC source—the higher the frequency, the lower (or more magnified) the time base.

For example, if you want to see one cycle of a 1 kHz signal, the time base should be around 1 millisecond.

### 6.10.1.1 X Position (-5.00 — 5.00)

This setting controls the signal's starting point on the x-axis. When **X Position** is 0, the signal starts at the left edge of the display. A positive value (for example, 2.00) shifts the starting point to the right. A negative value (for example, -3.00) shifts the starting point to the left.

### 6.10.1.2 Axes (Y/T, A/B, and B/A)

The axes of the oscilloscope display can be switched from showing waveform magnitude against time (Y/T) to showing one input channel against the other (A/B or B/A). The latter settings display frequency and phase shifts, known as Lissajous patterns, or they can display a hysteresis loop. When comparing channel A's input against channel B's (A/B), the scale of the x-axis is determined by the volts-per-division setting for channel B (and vice versa).

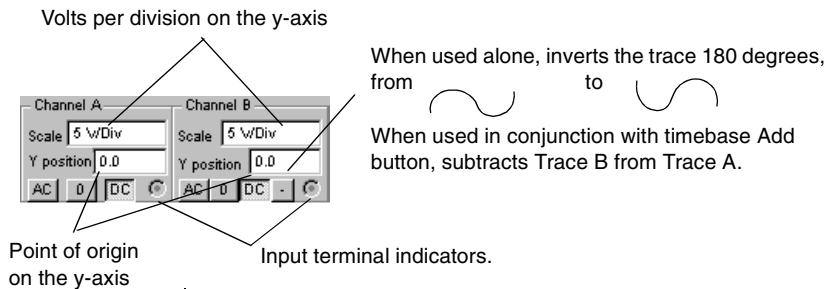
**Tip** To analyze waveforms in detail, use the Design Bar button to stop and start. Continue the simulation when ready.

## 6.10.2 Grounding

It is not necessary to ground the oscilloscope, as long as the circuit to which it is attached is grounded.

## 6.10.3 Channel A and Channel B Settings

### 6.10.3.1 Volts per Division ( $10\mu$ V/Div — 5 kV/Div)



This setting determines the scale of the y-axis. It also controls the x-axis scale when A/B or B/A is selected.

To get a readable display, adjust the scale in relation to the channel's expected voltage. For example, an input AC signal of 3 volts fills the oscilloscope's display vertically when the y-axis is set to 1 V/Div. If the volts-per-division is increased, the waveform will become smaller. If the volts-per-division is decreased, the waveform's top will be cut off.

### 6.10.3.2 Y Position (-3.00 — 3.00)

This setting controls the point of origin for the y-axis. When **Y position** is set to 0.00, the point of origin is the intersection with the x-axis. Increasing **Y position** to 1.00, for example, moves 0 (the point of origin) up to the first division above the x-axis. Decreasing **Y position** to -1.00 moves 0 down to the first division below the x-axis.

Changing the **Y position** setting for channels A and B may help distinguish their waveforms for comparison.

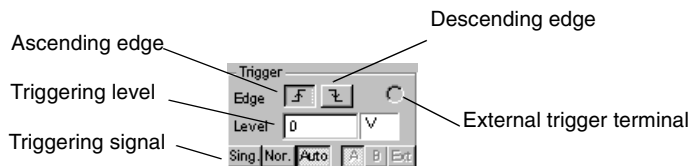
### 6.10.3.3 Input Coupling (AC, 0, and DC)

With AC coupling, only the AC component of a signal is displayed. AC coupling has the effect of placing a capacitor in series with the oscilloscope's probe. As on a real oscilloscope using AC coupling, the first cycle displayed is inaccurate. Once the signal's DC component has been calculated and eliminated during the first cycle, the waveforms will be accurate.

With DC coupling, the sum of the AC and DC components of the signal is displayed. Selecting 0 displays a reference flat line at the point of origin set by **Y position**.

**Note** Do not place a coupling capacitor in series with an oscilloscope probe. The oscilloscope will not provide a path for current, and the analysis will consider the capacitor improperly connected. Instead, choose AC coupling.

## 6.10.4 Trigger



These settings determine the conditions under which a waveform is first displayed on the oscilloscope.

### 6.10.4.1 Trigger Edge

- To start displaying the waveform on its positive slope or rising signal, click the “ascending edge” button.
- To start with the negative slope or falling signal, select the “descending edge” button.

### 6.10.4.2 Trigger Level (-999kV — 999 kV)

The trigger level is the point on the oscilloscope’s y-axis that must be crossed by the waveform before it is displayed.

**Tip** A flat waveform will not cross the trigger level. To see a flat signal, make sure the triggering signal is set to **Auto**.

### 6.10.4.3 Trigger Signal

Triggering can be internal, with reference to the input signal for channel A or B, or external, with reference to a signal through the external trigger terminal situated below the ground terminal on the **Oscilloscope** icon. If a flat signal is expected, or if signals are to be displayed as soon as possible, select **Auto**.

The **Sing** button is used to make the oscilloscope trigger one pass when the trigger level is met. Once the trace gets to the end of the scope face, the trace will not change until you click **Sing** again.

The **Nor** button is used to make the oscilloscope refresh every time the trigger level is met.

## 6.10.5 Using Cursors and Readouts

- To display the exact values of the wave, drag the vertical cursor until the desired portion appears.

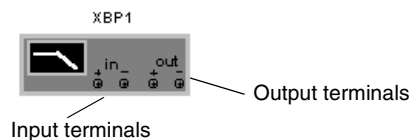
The boxes below the display show the time and the voltage at the probe connections, where the vertical cursor intersects the sine wave, and the difference between the two positions.

Once a circuit has been activated and its behavior simulated, you may move the oscilloscope's probes to other nodes without re-activating the circuit. Moving the probes automatically redraws the waveforms for the new nodes. If you fine-tune the oscilloscope's settings either during or after simulation, the display redraws automatically.

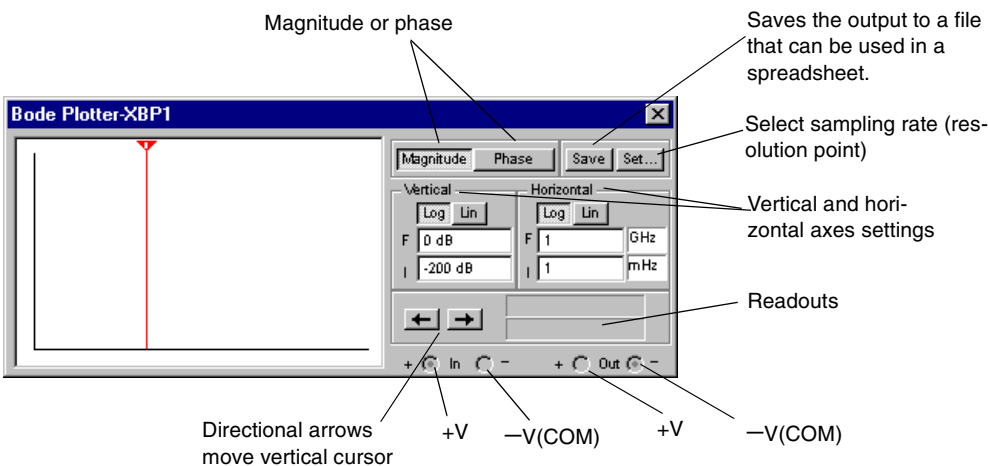
**Note** If the oscilloscope settings or analysis options are changed to provide more detail, the waveforms may appear choppy or uneven. If so, activate the circuit again to get more detail. You can also increase the precision of a waveform by increasing the simulation time step using the default instrument settings, as explained in “6.3 Default Instrument Analysis Settings” on page 6-5.



# 6.11 Bode Plotter



The Bode plotter produces a graph of a circuit's frequency response and is most useful for analyzing filter circuits. The Bode plotter is used to measure a signal's voltage gain or phase shift. When the Bode plotter is attached to a circuit, a spectrum analysis is performed.



The Bode plotter generates a range of frequencies over a specified spectrum. The frequency of any AC sources in the circuit does not affect the Bode plotter. However, an AC source must be included somewhere in the circuit.

The initial and final values of the vertical and horizontal scales are preset to their maximum value. These values can be changed to see the plot on a different scale. If the scale is expanded or the base changed after simulation is complete, you may need to activate the circuit again to get more detail in the plot. Unlike most test instruments, if the Bode plotter's probes are moved to different nodes, it is best to re-activate the circuit to ensure accurate results.

## 6.11.1 Magnitude or Phase

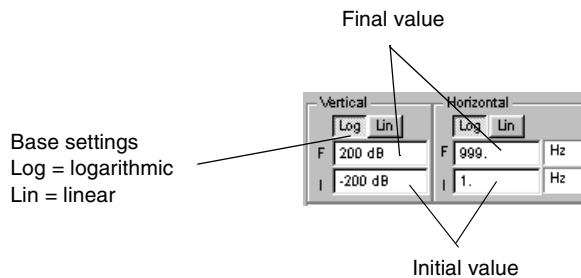
Magnitude measures the ratio of magnitudes (voltage gain, in decibels) between two nodes, V+ and V-.

Phase measures the phase shift (in degrees) between two nodes.

Both gain and phase shift will be plotted against frequency (in hertz).

- If V+ and V- are single points in a circuit:
  1. Attach the positive IN terminal and the positive OUT terminal to connectors at V+ and V-.
  2. Attach the negative IN and OUT terminals to a ground component.
- If V+ (or V-) is the magnitude or phase across a component, attach both IN terminals (or both OUT terminals) on either side of the component.

## 6.11.2 Vertical and Horizontal Axes Settings



### 6.11.2.1 Base Settings

A logarithmic base is used when the values being compared have a large range, as is generally the case when analyzing frequency response. For example, if measuring a signal's voltage gain, the decibel value is calculated as follows:

$$dB = 20 * \log_{10} \left( \frac{V_{out}}{V_{in}} \right)$$

The base scale can be changed from logarithmic (Log) to linear (Lin) without the circuit being activated again. (Only when using a logarithmic scale is the resulting graph referred to as a Bode plot.)

### 6.11.2.2 Horizontal Axis Scale (1 mHz — 999.9 GHz)

The horizontal or x-axis always shows frequency. Its scale is determined by the initial (I) and final (F) settings for the horizontal axis. Since a frequency response analysis requires a large frequency range, a logarithmic scale is often used.

**Note** When setting the horizontal axis scale, the initial (I) frequency must be smaller than the final (F) frequency. Multisim will not let you set I larger than F.

### 6.11.2.3 Vertical Axis Scale

The units and scale for the vertical axis depend on what is being measured and the base being used, as shown in the table below.

When Measuring...	Using the Base...	Minimum Initial Value is...	Maximum Final Value is...
Magnitude (gain)	Logarithmic	-200 dB	200 dB
Magnitude (gain)	Linear	0	10e+09
Phase	Linear	-720°	720°

When measuring voltage gain, the vertical axis shows the ratio of the circuit’s output voltage to its voltage. For a logarithmic base, the units are decibels. For a linear base, the vertical axis shows the ratio of output voltage to input voltage. When measuring phase, the vertical axis always shows the phase angle in degrees. Regardless of the units, you can set initial (I) and final (F) values for the axis using the Bode plotter’s controls.

**Note** When setting the vertical axis scale, the initial (I) frequency must be smaller than the final (F) frequency. Multisim will not let you set I larger than F.

### 6.11.3 Readouts

Move the Bode plotter's vertical cursor to get a readout of the frequency and magnitude or phase at any point on the plot. The vertical cursor is stored at the left edge of the Bode plotter display.

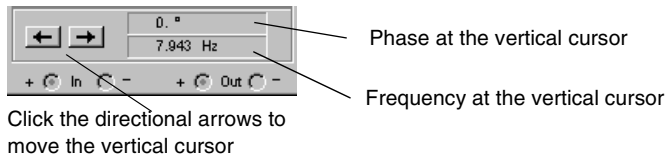
➤ To move the vertical cursor:

- click the arrows near the bottom of the Bode plotter

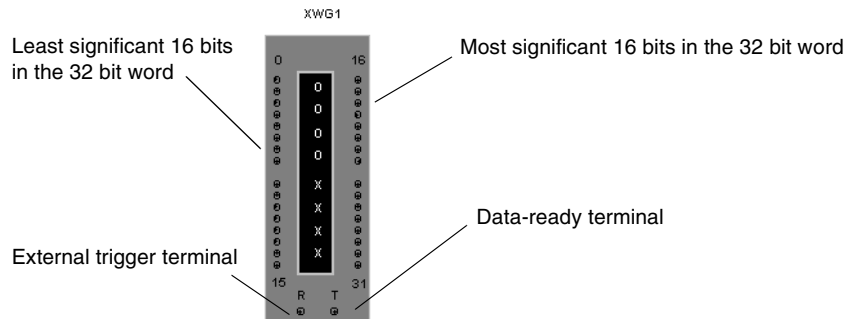
or

- drag the vertical cursor from the left edge of the Bode plotter display to the point on the plot you want to measure.

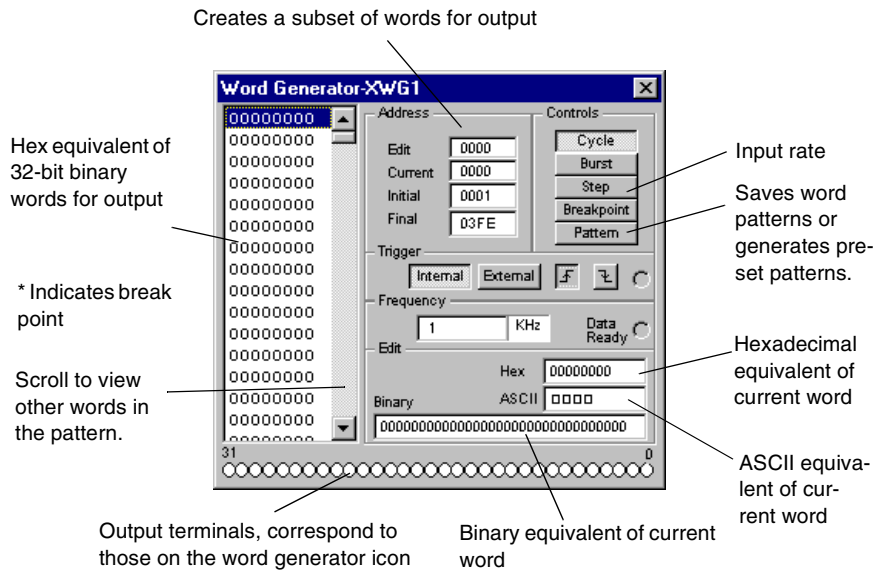
The magnitude (or phase) and frequency at the intersection of the vertical cursor and the trace are shown in the boxes beside the arrows.



## 6.12 Word Generator



Use the word generator to send digital words or patterns of bits into circuits to provide stimulus to digital circuits:



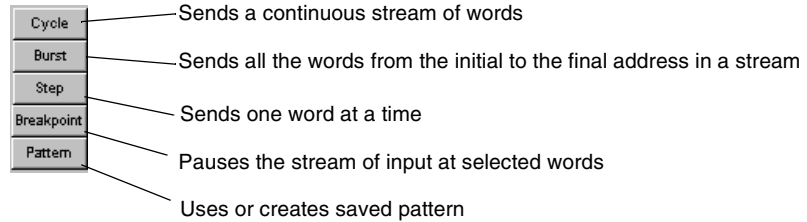
## 6.12.1 Entering Words

The left side of the word generator instrument face displays rows of 8-digit hexadecimal numbers, ranging from 00000000 to FFFFFFFF (0 to 4,294,967,265, in decimal). Each horizontal row represents a binary 32-bit word. When the word generator is activated, a row of bits is sent in parallel to the corresponding terminals at the bottom of the instrument.

- To change a bit value in the word generator, select the number you want to modify and type the new value in the **Hex**, **ASCII** or **Binary** fields, using the appropriate number format.

As the words are transmitted by the word generator, the value of each bit appears in the circles representing the output terminals at the bottom of the instrument.

## 6.12.2 Controls

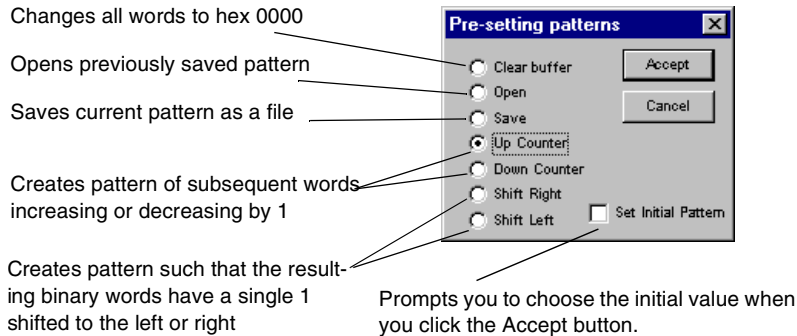


- To inject the 32-bit words into a circuit, click **Step**, **Burst** or **Cycle**. The current word appears in the box labeled **Current**.
- To transmit one word at a time into the circuit, click **Step**.
- To send all words in sequence, click **Burst**. Clicking **Cycle** sends a continuous stream of words that can be stopped by clicking **Cycle** again, or by pressing CTRL+T.  
Use **Breakpoint** when you want to pause and restart the stream of words at a specified word.
- To insert a breakpoint, select the word in the scroll list where you want the input to stop, then click **Breakpoint**. An asterisk marks a breakpoint in the scroll list.
- To remove a breakpoint, click on an existing breakpoint (\*) in the scroll list, then click **Breakpoint**.

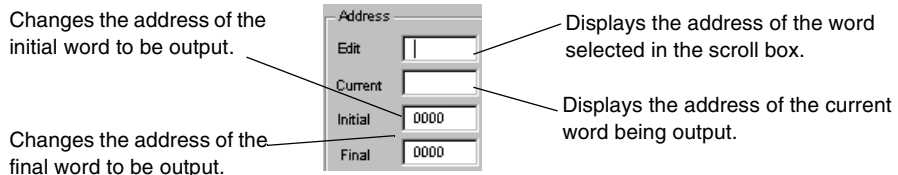
More than one breakpoint can be used. Breakpoints affect both **Cycle** and **Burst**.

## 6.12.3 Creating, Saving and Reusing Word Patterns

Click **Pattern** to display a set of options that allow you to save word patterns entered in the word generator to a file and load previously saved word patterns. This function can also be used to generate useful patterns or to clear the display.



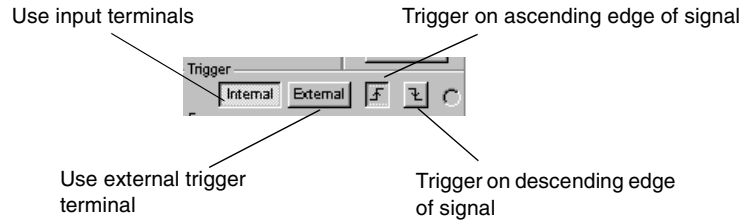
## 6.12.4 Addressing



Each word in the word generator's scroll window has an address, expressed as a 4-character hexadecimal number. When a word in the scroll box is changed, its address appears in the **Edit** box. As the word generator outputs words, each word's address appears in the **Current** box.

- To create a subset of the words to be output, enter first and last addresses in the **Initial** and **Final** fields.

## 6.12.5 Triggering

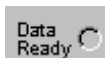


- To use the word generator clock to trigger input from the word generator's input field to the circuit, click **Internal**. To use input through the external terminal instead, with each input cycle causing one word to be transmitted, click **External**.  
Use the "ascending/descending edge" buttons to control whether the input signal triggers the word generator on its ascending or descending edge.

## 6.12.6 Frequency and Data Ready



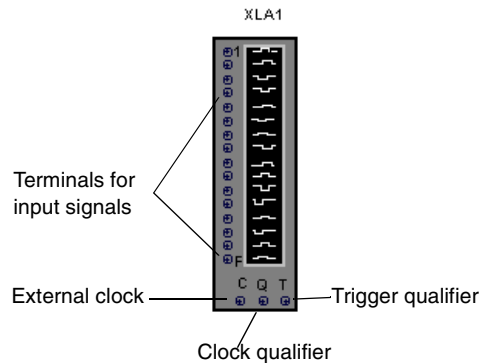
Set the clock frequency of the word generator in Hz, kHz or MHz. Each word is placed on the output terminals for the duration of one clock cycle.



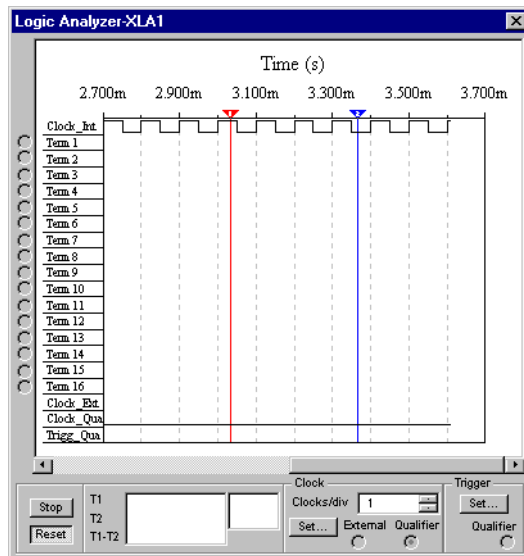
Enabling this option lets the circuit know that data from the word generator is ready.



## 6.13 Logic Analyzer



The logic analyzer displays the levels of up to 16 digital signals in a circuit. It is used for fast data acquisition of logic states and advanced timing analysis to help design large systems and carry out troubleshooting.



The 16 circles on the left side of the icon correspond to the terminals and horizontal rows across the instrument face. When the terminal is connected with a node, its circle is displayed with a black dot and the node's name and color are displayed. Otherwise the terminal circle is displayed without a black dot.

When a circuit is activated, the logic analyzer records the input values on its terminals. When the triggering signal is seen, the logic analyzer displays the pre- and post-trigger data. Data is displayed as square waves over time. The top row displays values for channel 0 (generally the first bit in a digital word), the next row displays values for channel 1, and so on. The binary value of each bit in the current word is displayed in the terminals on the left side of the instrument face. The time axis is displayed as the top axis of the signal display screen. The screen also displays the internal clock signal, external clock signal, external clock qualify signal and trigger qualify signal.

- To specify the number of samples stored before and after triggering, click **Set** in the **Clock** box or use the default instrument settings, as explained in “6.3 Default Instrument Analysis Settings” on page 6-5.

The logic analyzer stores data until it reaches the pre-trigger number of samples. Then, it begins discarding samples as new samples appear until it sees the trigger signal. After the trigger signal, samples are stored up to the value of the post-trigger samples.

The time position automatically displays the time position values of the two crosshair cursors, T1 and T2, when sampling stopped. It also automatically moves the first crosshair cursor T1 to the position of time zero, when sampling stopped.

- To change the threshold voltage, use the default instrument settings, as explained in “6.3 Default Instrument Analysis Settings” on page 6-5.

## 6.13.1 Stop & Reset

Selecting **Simulate/Pause** while the circuit is simulating, pauses both the circuit and the Logic Analyzer. By using the scroll bar at the bottom of the instrument face, you can view the results of the Logic Analyzer at any period from the first instance the trigger condition was met as long as the Logic Analyzer was not reset. If you did reset the Logic Analyzer, you will be able to view the results from the time the instrument was reset and the trigger condition is met.

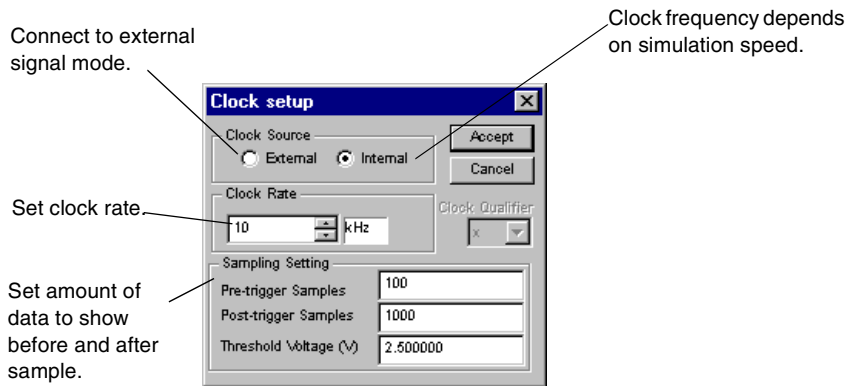
- To restart the instrument, click **Reset**. This action will clear all information in the instrument and start storing new information from the instance the trigger condition is met after you clicked **Reset**.
- To dump stored data when the logic analyzer is not triggered, click **Stop**. If the logic analyzer is already triggered and displaying data, clicking **Stop** will stop the instrument and allow the simulation to continue. After you have stopped the instrument, you must reset the Logic Analyzer to begin storing data again.
- To clear the logic analyzer’s display, click **Stop**, then **Reset**. Clearing the display does not disrupt the simulation. The instrument will start storing data from the instance the trigger condition is met after you pushed the Reset.

## 6.13.2 Clock

The clock informs the logic analyzer when to read an input sample. The clock can be internal or external.

➤ To adjust the clock settings:

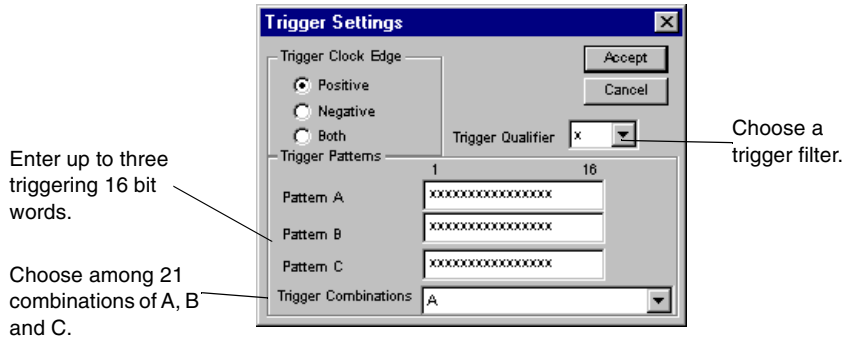
1. Click **Set** in the **Clock** area of the logic analyzer. The Clock Setup screen appears.



2. Select external or internal clock mode.
3. Set the internal clock rate. The clock qualifier is an input signal that filters the clock signal. If it is set to "x", then the qualifier is disabled and the clock signal determines when samples are read. If it is set to "1" or "0", the samples are read only when the clock signal matches the selected qualifier signal.
4. Set how much data to show before (**Pre-trigger samples**) and after (**Post-trigger samples**) the sample.
5. Click **Accept**.

### 6.13.3 Triggering

The logic analyzer can be made to trigger upon reading a specified word or combination of words or when meeting the increase edge or decrease edge of the clock signal.



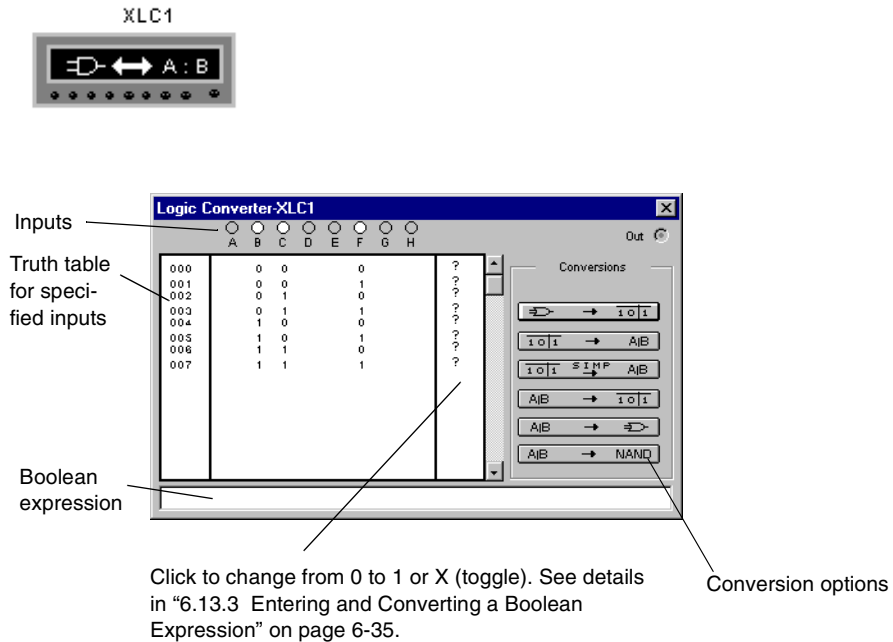
- To specify up to three trigger words or word combinations:
  1. Click **Set** in the **Trigger** box of the logic analyzer.
  2. Select **Positive**, **Negative** or **Both** positive and negative clock edge.
  3. Click in the box labeled **Pattern A**, **Pattern B**, or **Pattern C** and enter a binary word. An “x” means either 1 or 0.
  4. From the **Trigger combinations** drop-down list, select the desired combination. (See below for a list of combinations.)
  5. From the **Trigger qualifier** drop-down list, select the desired trigger qualifier. The trigger qualifier is an input signal that filters the triggering signal. If it is set to “x”, then the qualifier is disabled and the trigger signal determines when the logic analyzer is triggered. If it is set to “1” or “0”, the logic analyzer is triggered only when the triggering signal matches the selected trigger qualifier.
  6. Click **Accept**.

The possible trigger combinations are:

A	B	C
A or B	A or C	B or C
A OR B OR C	A AND B	A AND C
B AND C	A AND B AND C	NO B
A NO C	B NO C	A THEN B
A THEN C	B THEN C	(A OR B) THEN C
A THEN (B OR C)	A THEN B THEN C	A THEN (B WITHOUT C)


## 6.14 Logic Converter

The logic converter is able to perform several transformations of a circuit representation or digital signal. This is a useful tool for digital circuit analysis, but has no real-world counterpart. It can be attached to a circuit to derive the truth table or Boolean expression the circuit embodies, or to produce a circuit from a truth table or Boolean expression.





- Click the circles or the label below them to display the inputs for that terminal.

## 6.14.1 Deriving a Truth Table from a Circuit

- To derive a truth table from a circuit schematic:
  1. Attach the input terminals of the logic converter to up to eight nodes in the circuit.
  2. Connect the single output of the circuit to the output terminal in the **Logic Converter** icon.
  3. Click the **Circuit to Truth Table**  button. The truth table for the circuit appears in the logic converter's display.

## 6.14.2 Entering and Converting a Truth Table

- To construct a truth table:
  1. Click the number of input channels you want, from A to H, across the top of the logic converter. The display area below the terminals fills up with the necessary combinations of ones and zeros to fulfill the input conditions. The values in the output column on the right are initially set to 0.
  2. Edit the output column to specify the desired output for each input condition.
- To change an output value, click on it to move among the three possible settings: “0”, “1” and “x” (an “x” indicates that either 1 or 0 is acceptable).
- To convert a truth table to a Boolean expression, click the **Truth Table to Boolean Expression**  button. The Boolean expression is displayed at the bottom of the logic converter.
- To convert a truth table to a simplified Boolean expression, or to simplify an existing Boolean expression, click the **Simplify**  button.

The simplification is performed by the Quine-McCluskey method, rather than the more familiar Karnaugh mapping technique. Karnaugh mapping works for only small numbers of variables and requires human intuition, while Quine-McCluskey has proved to be exhaustive for any number of variables but is too cumbersome for manual solutions.

**Note** Simplifying a Boolean expression requires substantial memory. If not enough memory is available, Multisim may not be able to complete this operation.

### 6.14.3 Entering and Converting a Boolean Expression

A Boolean expression can be entered in the box at the bottom of the logic converter using either sum-of-products or product-of-sums notation.

- To convert a Boolean expression to a truth table, click the **Boolean Expression to Truth**

**Table** button.

- To convert a Boolean expression to a circuit, click the **Boolean Expression to Circuit**

button.

The logic gates that fulfill the Boolean expression appear on the circuit window. The components are selected so you can move them to a different location on the circuit window or put them in a subcircuit. Deselect the components by clicking on an empty spot on the circuit.

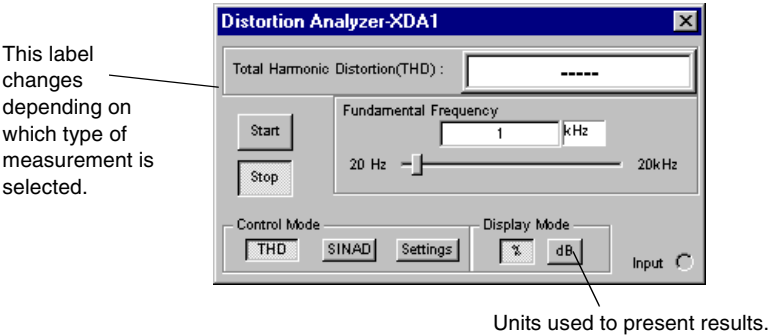
- To see a circuit that fulfills the conditions of the Boolean expression using only NAND gates, click the **Boolean Expression to NAND** button.

## 6.15 Distortion Analyzer

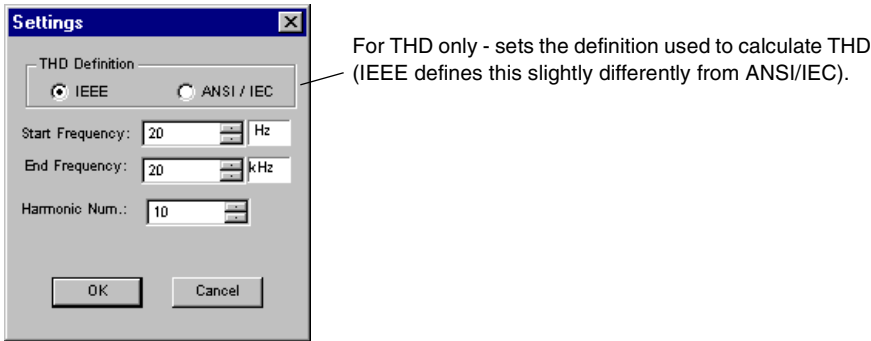
**Note** If the simplified version option is selected, the Distortion Analyzer is hidden. For details on the simplified version option, see “13.3.1.2 Simplified Version” on page 13-6.



A typical distortion analyzer provides distortion measurements for signals in the range of 20 Hz to 100 kHz, including audio signals.



The types of measurements performed are either Total Harmonic Distortion (THD) or Signal Plus Noise and Distortion (SINAD). To set the way results are to be displayed for either type of measurement, click **Settings**.



### 6.15.1 Harmonic Distortion

Harmonic distortion produces signals at harmonics of the test frequency. For example, for a 1 kHz signal, the harmonics may be at 2 kHz, 3 kHz, 4 kHz, etc.

A very sharp tunable notch is required to measure harmonic distortion. The filter is tuned to the test frequency such as 1 kHz, which will remove the 1kHz signal, leaving only the harmonics or the distortion. The distortion harmonics are measured and the resulting value is compared to the amplitude of the test signal.



## 6.15.2 SINAD

This type of measurement measures the ratio of (signal plus noise and distortion)/(noise and distortion).

## 6.16 Spectrum Analyzer

**Note** If the simplified version option is selected, the Spectrum Analyzer is hidden. For details on the simplified version option, see “13.3.1.2 Simplified Version” on page 13-6.



The spectrum analyzer is used to measure amplitude versus frequency. It performs a similar function in the frequency domain as an oscilloscope performs in the time domain. It operates by sweeping through a range of frequencies. The amplitude of the signal at the input of the receiver is plotted against the frequency of the signal. This instrument is capable of measuring a signal's power at various frequencies, and helps determine the existence of the frequency components' signal.

The spectrum analyzer is part of the RF Design Module. For more details, see the “RF” chapter.

## 6.17 Network Analyzer

**Note** If the simplified version option is selected, the Network Analyzer is hidden. For details on the simplified version option, see “13.3.1.2 Simplified Version” on page 13-6.



The network analyzer is used to measure the scattering parameters (or S-parameters) of a circuit, commonly used to characterize a circuit intended to operate at higher frequencies. These S-parameters are used to derive matching cells using other Multisim analyses. The network analyzer also calculates H, Y, Z parameters.

The circuit is idealized as a two-port network. To properly use the network analyzer, the circuit must be left open at its input and output ports. During simulation the network analyzer completes the circuit being analyzed by inserting its sub-circuits. You need to remove these sub-circuits from the circuit before performing other analysis and simulation.

The network analyzer is part of the RF Design Module. For more details, see Chapter 14, “RF”.



# Chapter 7

## Simulation

### 7.1 About this Chapter

This chapter explains the various types of simulation available in Multisim 2001, the application for which each type is appropriate, how the types of simulation are used separately and together, and finally some of the underlying logic of Multisim 2001 simulation.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 7.2 Introduction to Simulation

Simulation is a mathematical way of emulating the behavior of a circuit. With simulation, you can determine a circuit's performance without physically constructing the circuit or using actual test instruments. Although Multisim makes simulation intuitively easy-to-use, you should be aware that the technology underlying the speed and accuracy of the simulation, as well as its ease of use, is extremely complex. For that reason, explaining how Multisim performs its simulation is beyond the scope of this manual.

#### 7.2.1 What Type of Simulation Should I Use?

The type of simulation that is appropriate for a circuit depends on the type of circuit and how you plan to physically implement it. For example, analog, digital and mixed analog/digital circuits to be built as a PCB are, in general, best simulated with a SPICE simulation. Digital circuits to be implemented in Programmable Logic Devices are usually simulated at the behavioral language level, most commonly with VHDL or Verilog HDL. (See Chapter 10, "HDLs and Programmable Logic" for details.)

For very complex digital devices (LSI or VLSI chips) such as microprocessors or memory, SPICE models are not usually practical, and in these cases, VHDL or Verilog HDL is the preferred solution.

## 7.2.2 What Kind of Simulation Does Multisim Support?

Multisim offers multiple simulators, optimized to meet the needs of various types of circuit designs and implementation. These simulators include:

- SPICE (including specialized RF simulation)
- VHDL
- Verilog HDL
- Co-simulation of all three together

When designing with programmable logic devices such as Field Programmable Gate Arrays (FPGAs) or Complex Programmable Logic Devices (CPLDs), VHDL or Verilog HDL simulators have traditionally been used separately from each other and from SPICE. With Multisim, these simulators can be also used in combination. For example, to simulate a PCB designed using Multisim's schematic capture front end, Multisim uses SPICE for most of the simulation (that is, components will use SPICE models) and VHDL or Verilog HDL for modeling the most complex digital parts (including programmable devices), all brought together in the co-simulation mode. Multisim's simulation engine checks which type of model (SPICE, VHDL, Verilog HDL, etc.) is used, as indicated in the component database, and calls the appropriate simulator. It then controls the passing of information between these various simulators, all without requiring your intervention.

To simulate a system- or board-level design, these simulators are used together in a co-simulation environment. This means that, for example, a chip that is modeled using VHDL or Verilog HDL (whether a CPLD/FPGA or a complex digital chip such as a microcontroller) can be a component in a PCB design. Multisim will simulate most of the board using SPICE, but automatically simulate the VHDL- or Verilog HDL-modeled chip with VHDL or Verilog HDL simulation. This co-simulation environment is described in this chapter. Communications between the multiple simulation engines in co-simulation mode are extremely complex, yet remain very easy for you to use. All the results are combined together, so they can be displayed on a common set of instruments and analyses, as if all the devices were modeled using the same technology.

## 7.3 Using Multisim Simulation

This chapter explains the simulation of PCB-level circuits. This, by default, is primarily the function of the Multisim SPICE simulator. In cases where the PCB circuit makes use of a complex digital chip modeled with VHDL or Verilog HDL (including a programmable logic device), Multisim automatically simulates that device with the correct VHDL or Verilog HDL simulator. This process is invoked automatically during simulation of the PCB-level circuit, is performed simultaneously with the SPICE simulation, and is transparent to you.

To view the results of your simulation, you will need to use either a virtual instrument or be running an analysis (explained in the next chapter) in order to display the simulation output. This output will include the combined results of all Multisim simulation engines (SPICE, VHDL, Verilog HDL), all brought together conveniently in common displays (instruments or Grapher).



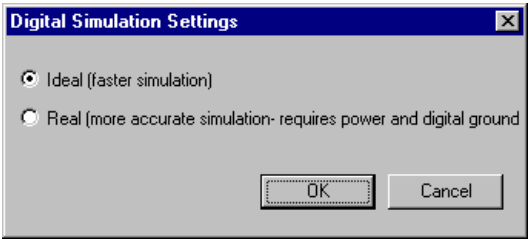
**Note** For design entry, simulation and source code debugging of individual VHDL or Verilog HDL modelled chips (on their own, not as part of a PCB circuit), likely as part of the programmable logic design flow, see “7.5 VHDL Simulation” on page 7-13 and “7.6 Verilog HDL Simulation” on page 7-13 for an introduction. For details, see Chapter 10, “HDLs and Programmable Logic”, which explains designing with VHDL and Verilog HDL.

### 7.3.1 Simulating Circuits with Digital Components

When simulating circuits with digital components, you have the option of simulating for speed or for accuracy. The “Ideal” option simulates your circuit quickly by not taking into account variances in digital power and internal tolerances. The time to simulate digital components is faster but the signal is not as accurate.

The “Real” option simulates your circuit accurately, but slower than the “Ideal” option, by accounting for all variances. When using Real simulation settings, you are required to add digital power and digital ground to your circuit.

- To select a digital simulation option:
  1. Choose **Simulate/Digital Simulation Setting**.



2. Select either Ideal (faster) or Real (more accurate) depending on the speed and accuracy you require.
3. Click **OK**.

### 7.3.2 Start/Stop/Pause Simulation



- To simulate a circuit, click the Simulate button on the Design Bar and choose **Run** from the pop-up menu that appears. Multisim begins to simulate the circuit's behavior.  
During simulation, messages about the simulation results and any problems with the simulation are written to the simulation error log/audit trail. The error log/audit trail appears automatically when you stop the simulation. If you want to keep an eye on the progress of the simulation, you can display the error log/audit trail during simulation. To display it, from the **View** menu choose **Show/ Simulation Error Log/Audit Trail**.



- To pause the simulation while it is running, click the Simulate button on the Design Bar and choose **Pause** from the pop-up menu that appears. To resume the simulation from the same point as when you paused, click the Simulate button and choose **Pause** again.
- To stop a simulation, click the Simulate button on the Design Bar and choose **Run** from the pop-up menu that appears. If you restart the simulation after stopping it, it will restart from the beginning (unlike **Pause**, which allows you to restart from the point you paused).
- Alternatively, you can run simulations by choosing **Run** and **Pause** from the **Simulation** menu using the same instructions as above.



- A final option available to you for starting and stopping simulations is to use the simulation switch. From the **View** menu, choose **Show Simulation Switch** to display a switch for “activating” your circuit. The switch can be used to start, stop, and pause the simulation.

### 7.3.3 Interactive Simulation

In a capacity unique to Multisim, simulation is interactive. You can simply change the values of “interactive” components (those whose behavior can be controlled through the keyboard) and see the simulation results instantly. Interactive components include such devices as the potentiometer, variable capacitor, variable inductor, and multiple switcher. For example, changing a 100 kohm resistor to the next smaller resistor may alter the results more than desired, but with Multisim, you could use a variable resistor, reducing its value gradually, all the time seeing the simulation result change, until you reach the correct result.

### 7.3.4 Circuit Consistency Check

When you simulate your circuit or perform an analysis, a circuit consistency check is performed to determine if the circuit is “legal” — for example, if a ground is present. Errors are written to the error log. This function speeds your simulation process, since it alerts you to items that may cause simulation errors and allows you to correct them before simulating. Keep in mind that the types of problems found by the circuit consistency check are those that will cause simulation errors. The circuit consistency check does not necessarily indicate a circuit’s viability.

### 7.3.5 Miscellaneous SPICE Simulation Capabilities

Multisim offers the following SPICE-specific simulation capabilities.

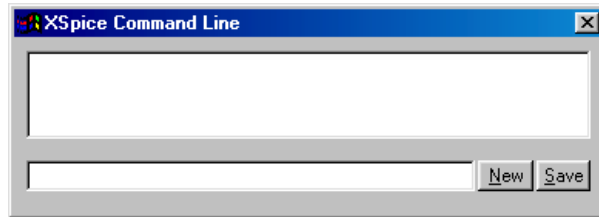
#### 7.3.5.1 Component Tolerances

As explained in “4.5 Component Nominal Values and Tolerances” on page 4-33, Multisim allows you to use “real world components” that introduce variances to simulate the performance of actual, physical components. Real world component settings are based on component tolerances, and therefore vary from their nominal values. This affects simulation results. For example, a 1 Kohm resistor with a 10% variance could vary 100 ohms either way.

### 7.3.5.2 Menu-Driven Simulation from Netlist Without Schematic

You can also run simulations from a command line.

- To open the command line interface, choose **View/Show Command Line**. The following window appears:



You can enter commands directly in this window. The most important commands are: SOURCE, PLOT, OP, SAVE, WRITE, TAN, SET and ANAC.

## 7.4 Multisim SPICE Simulation: Technical Detail

This section explains the basic technical methodology of circuit simulation in a SPICE-based simulator, including an outline of the stages of circuit simulation and an explanation of the methods of formulation and solution used in the circuit simulation. It is not necessary to comprehend this information to make use of Multisim's simulation, but you may find it interesting. In addition, if you would like some advanced technical hints for user controllable simulation settings, see "7.3.7 User Setting: Maximum Integration Order" on page 7-11.

To fully understand the information in this section, you should be acquainted with the theory of electronic circuit simulation and the mathematics involved.

### 7.4.1 BSpice/XSpice Support

Multisim incorporates SPICE3F5 and XSpice at the core of its simulation engine, with customized enhancements designed by Electronic Workbench specifically for optimizing simulation performance with digital and mixed-mode simulation. Both SPICE3F5 and XSpice are industry-accepted, public-domain standards, unlike non-standard, proprietary formats such as PSPICE. SPICE3F5 is the most recent edition of the SPICE (Simulation Program with Integrated Circuit Emphasis) core designed by the University of California at Berkeley. SPICE3F5 has evolved from the original program developed and released in 1972. It is commonly called BSpice. XSpice is a set of unique enhancements made to SPICE, under contract to the US Air Force, which included specialized modeling subsystems.



## 7.4.2 Circuit Simulation Mechanism

After you create a circuit schematic and begin simulation, the solution of the circuit and generation of the data you see on instruments such as the oscilloscope is the role of the simulator. More specifically, the simulator is the part of Multisim that calculates a numerical solution to a mathematical representation of the circuit you created.

For this calculation to occur, each component in a circuit is represented by a mathematical model. Mathematical models link the schematic in the circuit window with the mathematical representation for simulation. The accuracy of the component models is one of the key items that determines the degree to which simulation results match real-world circuit performance.

The mathematical representation of a circuit is a set of simultaneous, nonlinear differential equations. The main task of the simulator is to solve these equations numerically. A SPICE-based simulator transforms the nonlinear differential equations into a set of nonlinear algebraic equations. These equations are further linearized using the modified Newton-Raphson method. The resulting set of linear algebraic equations is efficiently solved using the sparse matrix processing LU factorization method.

## 7.4.3 Four Stages of Circuit Simulation

The simulator in Multisim, like other general-purpose simulators, has four main stages: input, setup, analysis and output (described below).

Stage	Description
Input stage	Simulator reads information about your circuit (after you have built a schematic, assigned values and chosen an analysis).
Setup stage	Simulator constructs and checks a set of data structures that contain a complete description of your circuit.
Analysis stage	The circuit analysis specified in the input stage is performed. This stage occupies most of the CPU execution time and is actually the core of circuit simulation. The analysis stage formulates and solves circuit equations for the specified analyses and provides all the data for direct output or postprocessing.
Output stage	You view the simulation results. You can view results on Os such as the oscilloscope, on graphs that appear when you run an analysis, or in the log file/audit trail.

## 7.4.4 Equation Formulation

In a circuit, each common point created by wires and connectors is called a node. The simulator calculates the voltage at each node. Each branch joining two nodes will have a separate current flowing through it.

To calculate a circuit solution, a circuit is represented internally as a system of equations, in the form:

$$A * X = B$$

where:

- A = modified nodal admittance matrix with dimension  $n \times n$
- X = vector of unknowns with dimension  $n$
- B = vector of constants, also with dimension  $n$
- n = number of unknowns.

The system of equations is formulated using a general circuit analysis method called the Modified Nodal Approach (MNA).

The unknowns ( $n$ ) include each node voltage (excluding ground), as well as the voltage source currents. B contains the voltage and current source constants, and the entries in the admittance matrix (A) are determined by Ohm's law and Kirchhoff's current and voltage laws.

The modified nodal admittance matrix is deemed sparse because it contains more zeros than non-zeros. Making use of a linked list, the solution of circuit equations can be performed by employing non-zero terms only. This method is called Sparse Matrix Technique. Generally, a sparse matrix approach requires less memory consumption and achieves faster simulation.

## 7.4.5 Equation Solution

Multisim solves circuit equations for linear and nonlinear circuits using a unified algorithm. The solution of a linear DC circuit is treated as a special case of general nonlinear DC circuits.

LU factorization is used to solve the system of sparse modified nodal matrix equations described previously (a set of simultaneous linear equations). This involves decomposing the matrix A into two triangular matrices (a lower triangular matrix, L, and an upper triangular matrix, U) and solving the two matrix equations using a forward substitution and a backward substitution.

Several efficient algorithms are used to avoid numerical difficulties due to the modified nodal formulation, to improve numerical calculation accuracy and to maximize the solution efficiency. These include:

- A partial pivot algorithm that reduces the round-off error incurred by the LU factorization method.
- A reordering algorithm that improves the matrix condition.
- A reordering algorithm that minimizes nonzero terms for the equation solution.

A nonlinear circuit is solved by transforming it into a linearized equivalent circuit at each iteration and iteratively solving the linear circuit using the above-described method. Nonlinear circuits are transformed into linear ones by linearizing all nonlinear components in the circuit using the modified Newton-Raphson method.

A general nonlinear dynamic circuit is solved by transforming the circuit into a discretized equivalent nonlinear circuit at each time point and solving it using the method for a nonlinear DC circuit described above. A dynamic circuit is transformed into a DC circuit by discretizing all dynamic components in the circuit using an appropriate numerical integration rule.

## 7.4.6 Numerical Integration

To approximate the value of the integral of the differential equations used in the time-domain solution, Multisim optionally uses two numerical integration methods:

- the Trapezoidal (default) method
- the Gear (order from 1 to 6) method.

When the trapezoidal method is applied, the following approximation is used to discretize the differential equations:

$$V_{n+1} = V_n + \frac{h}{2} \left( \frac{dV_{n+1}}{dt} + \frac{dV_n}{dt} \right)$$

where

- $V_{n+1}$  = present unknown voltage value
- $V_n$  = previous time-point solution
- $h$  = time step length
- $n$  = time interval.

The first-order Gear integration is the popular Backward Euler method. The second-order variable step size Gear integration formula is:

$$\frac{dV_{n+1}}{dt} = \frac{2h_n + h_{n-1}}{h_n(h_n + h_{n-1})}V_{n+1} + \frac{h_n + h_{n-1}}{h_n - h_{n-1}}V_n + \frac{h_n}{h_{n-1}(h_n + h_{n-1})}V_{n-1}$$

where

- $V_{n+1}$  = present unknown solution
- $V_n$  = previous first time-point solution
- $V_{n-1}$  = previous second time-point solution
- $h_n$  = present time step
- $h_{n-1}$  = previous time step

## 7.4.7 User Setting: Maximum Integration Order

You can change the maximum order for integration method using the MAXORD analysis option (see “8.28 Analysis Options” on page 8-78). Using a higher order (3 through 6) Gear method theoretically leads to more accurate results, but slows down the simulation. Be aware that the maximum order for integration method is the maximum order that could be used, but that the simulator selects the most appropriate order based on the circuit.

Due to the nature of the nonlinear components, each time point may involve solving the admittance matrix several times before converging to a solution. The point solution is reached when the difference between consecutive voltage values is less than the tolerance calculated internally in terms of the absolute and relative tolerances specified in the analysis options.

## 7.4.8 Convergence Assistance Algorithms

Multisim uses two modified Newton-Raphson continuation algorithms, Gmin stepping and Source stepping, to help find the solution during a DC Operating Point analysis of general nonlinear circuits.

### 7.4.8.1 Gmin Stepping

Gmin stepping is a multi-step iterative algorithm. This algorithm simply adds a conductance, Gmin, to the diagonal elements of the modified nodal admittance matrix so that a solution will converge more quickly. The basic concept is to keep the matrix well-conditioned.

Initially, a large Gmin value is applied and an approximate solution is found quickly. The initial value is set by the Gmin value times  $10^{\text{GminSteps}}$  Gmin. The Gmin value is taken from the GMIN (Gmin Minimum Conductance) analysis option and the number of steps from GMIN-STEPS (both options are described in more detail in “8.28 Analysis Options” on page 8-78). The conductance is then reduced by a factor of ten and the circuit is solved again by setting the previous solution as the initial guess of the next iteration. When Gmin is reduced to zero, a final solution of the circuit is performed and the correct answer is obtained. This actually divides one single-step solution of the simple nonlinear iteration into a multi-step solution, which uses the same algorithm but has many smaller steps.

### 7.4.8.2 Source Stepping

Source stepping is a convergence assistance algorithm. This algorithm solves a nonlinear circuit problem by setting a fraction of the source vector as a parameter variable to aid the convergence of the DC solution. Similar to the Gmin stepping method, Source stepping converts a single nonlinear circuit problem into a multi-step nonlinear circuit problem. Starting from a zero source vector, the source vector is slowly ramped up to its full DC value. At each source step, a simple nonlinear iteration solution is performed. The ramp rate is controlled by the SRCSTEPS (“Steps in source stepping algorithm”) analysis option (see “8.28 Analysis Options” on page 8-78).

## 7.5 RF Simulation



RF simulation is included with Multisim Power Professional and is available as part of the Professional Edition in an optional RF Design module. This section is simply a brief introduction to the simulation portion of the RF Design module.

You simulate an RF circuit the same way you simulate a board/system-level circuit in Multisim, as described in “7.2 Using Multisim Simulation” on page 7-3. This is because Multisim’s RF Design module simulates RF circuits using an optimized SPICE engine (as opposed to VHDL, Verilog HDL, etc.). There is no need to tell Multisim that your circuit is an RF circuit. RF simulation uses the SPICE simulation engine, but has been optimized to accurately simulate circuits designed to operate at higher frequencies, or at faster clock speeds (which generate RF characteristics). This optimization uses parts specifically designed and modeled to simulate accurately at these higher frequencies.

For detailed information on RF simulation and the RF Design module, see Chapter 14, “RF”.

## 7.6 VHDL Simulation



Multisim employs a specialized VHDL simulator which simulates, not at the SPICE level using schematic design entry, but at the behavioral language level. VHDL is one of the two most widely used behavioral languages, commonly used for designing and modeling:

- programmable logic devices such as CPLDs and FPGAs
- complex digital chips, such as memory, CPUs, microcontrollers, and other devices which could not be reasonably modeled using SPICE.

Even if you are not using such devices today, you will likely find it increasingly necessary to do so in the future. Multisim offers the perfect environment for experienced and novice VHDL users alike.

The Multisim VHDL simulator can be used in two ways:

- As part of the board/system design process, when components are modeled in VHDL instead of SPICE. Multisim automatically invokes the VHDL simulator as needed (this is called co-simulation). In this method, you do not need extensive VHDL knowledge, but can simply take advantage of the broader library of simulatable models for complex digital chips.

If you have VHDL simulation, it is invoked automatically by Multisim when you begin simulation (as described in “7.2 Using Multisim Simulation” on page 7-3) and a component that is part of your system- or board-level circuit is modeled in VHDL.

You need not do anything different to simulate in this co-simulation mode than you do to simulate a circuit with only SPICE-modeled parts. Just begin simulation as normal and Multisim takes care of the rest, recognizing when VHDL models exist and reacting accordingly.

- As part of the programmable logic design process, in which you write, simulate and debug VHDL source code. This is a much more involved process and does require knowledge of the VHDL language on your part. For details on this type of VHDL simulation, see Chapter 10, “HDLs and Programmable Logic”.

## 7.7 Verilog HDL Simulation



Multisim employs a specialized Verilog HDL simulator which simulates, not at the SPICE level using schematic design entry, but at the behavioral language level. Verilog HDL is, along with VHDL, one of the most widely used behavioral languages, commonly used for designing and modeling:

- programmable logic devices such as CPLDs and FPGAs
- complex digital chips, such as memory, CPUs, microcontrollers, and other devices which could not be reasonably modeled using SPICE.

Even if you are not using such devices today, you will likely find it increasingly necessary to do so in the future. Multisim offers the perfect environment for experienced and novice Verilog HDL users alike.

The Multisim Verilog HDL simulator can be used in two ways:

1. As part of the board/system design process, when components are modeled in Verilog HDL instead of SPICE. Multisim automatically invokes the Verilog HDL simulator as needed (this is called co-simulation). In this method, you do not need extensive Verilog HDL knowledge, but can simply take advantage of the broader library of simulatable models for complex digital chips.

If you have Verilog HDL simulation, it is invoked automatically by Multisim when you begin simulation (as described in “7.2 Using Multisim Simulation” on page 7-3) and a component that is part of your system- or board-level circuit is modeled in Verilog HDL.

For details on Verilog HDL simulation, see Chapter 10, “HDLs and Programmable Logic”. You need not do anything different to simulate in this co-simulation mode than you do to simulate a circuit with only SPICE-modeled parts. Just begin simulation as normal and Multisim takes care of the rest, recognizing when Verilog HDL models exist and reacting accordingly.

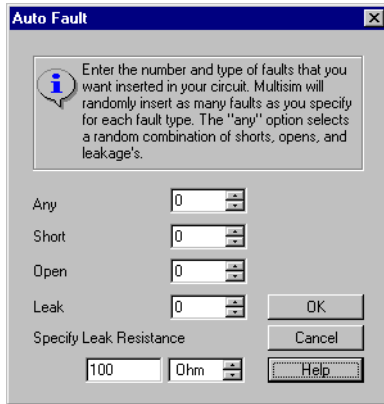
2. As part of the programmable logic design process, in which you write, simulate and debug Verilog HDL source code. For details on this type of Verilog HDL simulation, see Chapter 10, “HDLs and Programmable Logic”.

## 7.7.1 Using the Auto Fault Option

When you use the Auto Fault option, you specify the number of faults or, optionally, the number of faults per different type of fault, that you want Multisim to assign to placed components in the circuit.

➤ To use the auto fault option:

1. Choose **Simulate/Auto Fault Option**. The Auto Fault screen appears:



2. Use the up/down arrow keys or enter numerical values directly in the **Short**, **Open**, and **Leak** fields, or enter a numerical value in the **Any** field to let Multisim randomly select the type of faults to assign.
3. If you specify a number of leaks, enter a number and unit of measurement in the **Specify Leak Resistance** fields.
4. Click **OK** to apply the faults, or **Cancel** to cancel, and return to the circuit window.



# Chapter 8

## Analyses

### 8.1 About this Chapter

This chapter explains how to use the various analyses included in Multisim 2001. It explains how to work with analyses in general, the specific settings and options for each individual analysis, and how to view and manipulate analyses results.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 8.2 Introduction to Multisim Analyses

Multisim offers you many analyses, all of which utilize simulation to generate the data for the analysis you want to perform. These analyses can range from quite basic to extremely sophisticated, and can often require one analysis to be performed (automatically) as part of another.

For each analysis, you will need to decide upon some settings that will tell Multisim what you want the analysis to do.

In addition to the analyses provided by Multisim, you can also create user-defined analyses based on SPICE commands you enter.

When you activate an analysis, the results are displayed on a plot in Multisim's Grapher (unless you specify otherwise; see "8.22 Viewing the Analysis Results: Grapher" on page 8-64 for more information on the Grapher) and saved for use in the Postprocessor (not available in all editions of Multisim; see Chapter 9, "Postprocessor" for more information). Some results are also written to an audit trail, which you can view (see "8.21 Viewing the Analysis Results: Error Log/Audit Trail" on page 8-63 for more information).

For educational purposes, you can choose which analyses will be available to students in a circuit. See “13.3 Using Restrictions” on page 13-4 for more information. If the simplified version is selected in Global Restrictions, some of the analysis will be hidden. For more information on setting the simplified version see “13.3.1 Setting Global Restrictions” on page 13-4.

## 8.3 Working with Analyses

You need to know how to work with analyses in general as well as the specific options for each individual analysis. For each analysis, you can set:

- the analysis parameters (all have default values)
- how output variables are to be handled (required)
- a title for the analysis (optional)
- custom values for analysis options (optional).

Analysis settings are saved with the circuit.

The next section describes the general procedures for performing analyses, and the following sections describe the details of each particular analysis.

### 8.3.1 General Instructions



➤ To perform an analysis:

1. Click the Analyses button on the Design Bar, or choose **Simulate/Analyses**. A menu appears with the list of analyses available. If you have the simplified version option selected, the list of analyses will display only the basic types of analyses. For more information about the simplified version, see “13.3.1.2 Simplified Version” on page 13-6.
2. Select the desired analyses. Depending on the analysis selected, the screen that appears will include some or all of the following tabs:
  - the Analysis Parameters tab, where you set the parameters for this analysis
  - the Output Variables tab, where you specify what is to be done with specific analysis output (not present in all analyses)
  - the Miscellaneous Options tab, where you choose a title for the plot produced by the analysis, and set any custom values for analysis options
  - the Summary tab, where you see a consolidated view of all the settings for the analysis.

By default, most of the tabs display a streamlined version of the tab with just the most commonly used fields. Click **More** on the analysis screen to display the full screen displaying all available options. The options and settings available in these tabs are described in the following sections of this chapter. The examples in those sections show the full screen.

- To save the settings as the defaults for future use, click **Accept** on the analysis screen.
- To run the simulation with the current settings, click **Simulate** on the analysis screen.
- To run several analyses in batch, see “8.18 Batched Analyses” on page 8-60.

**Tip** To stop an analysis, press ESC.

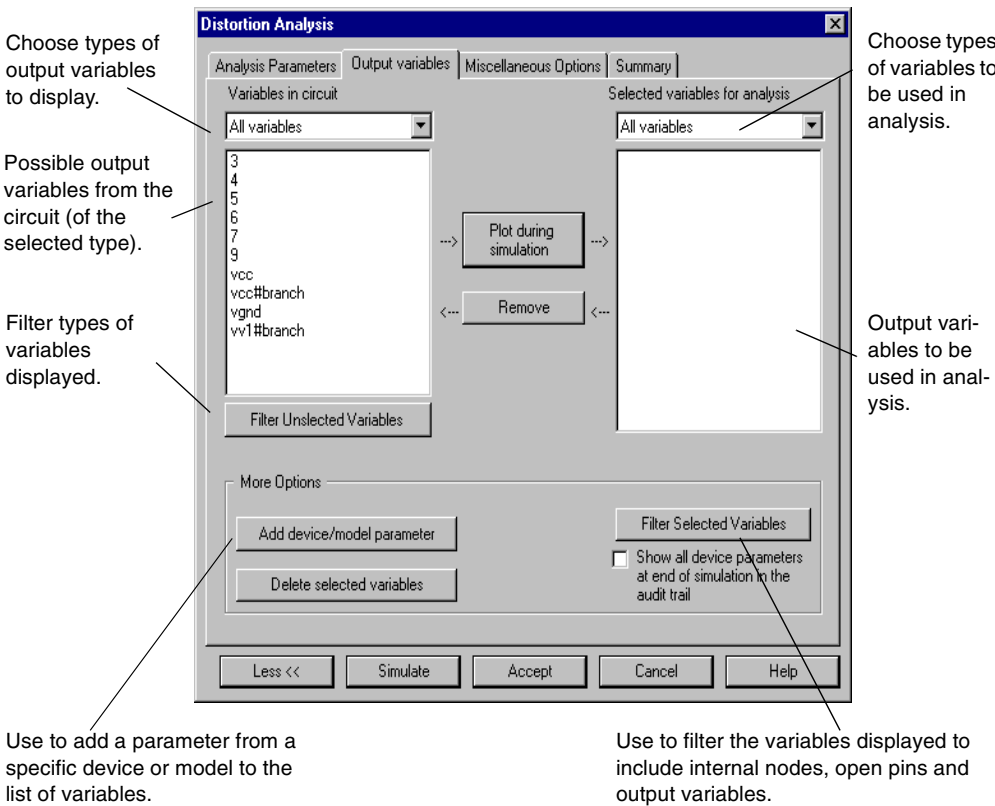
## 8.3.2 The Analysis Parameters Tab

The options available on the analysis parameters tab are different for each analysis, and so are described in different subsections of this chapter, one per analysis. Each analysis description includes guidelines for both normal and advanced use of the analysis.

Some lists of items are accompanied by a **Filter variables displayed** function. This lets you filter the items shown in that list, choosing whether or not to include internal nodes, submodules, open pins and device parameters.

### 8.3.3 The Output Variables Tab

This tab displays, on the left, all the possible output variables for the current circuit. You choose the variables you want to use in the analysis. By default, the tab displays just the most commonly used fields. Click **More** to display the full screen displaying all available options as shown in the example below.



### 8.3.3.1 Choosing How Output Variables are to be Handled

- To include the output variable in the plot, select a variable from the list on the left and click **Plot During Simulation**.
- To select for than one variable at a time, press SHIFT while selecting variables from the list on the left and click **Plot During Simulation**.
- To remove an item from the right hand list, select it and click **Remove**.

Using the Output Variables tab, you can also filter the variables list, filter the variables displayed, as well as add a wide range of device or model parameters.

By default, all variables are initially included in the **Variables in Circuit** list.

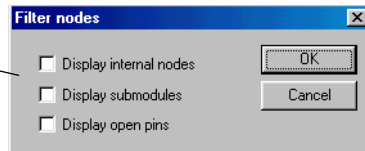
### 8.3.3.2 Filtering the Variable Lists

- To filter the variables list according to general variable type:
  1. Click the **Variables in Circuit** drop-down list.
  2. Click the general variable type (such as voltages, currents, device/model parameters) to include in the list.

You can filter the variables displayed to include internal nodes (such as nodes inside a BJT model or inside a SPICE subcircuit), open pins, as well as output variables from any submodules contained in the circuit.

- To filter the variables displayed:
  1. Click **Filter Selected Variables**. The Filter Nodes screen appears:

Enable the desired settings.



2. Enable one or more settings.
3. Click **OK**.

### 8.3.3.3 Adding Parameters to the Variable List

- To add a parameter from a specific device or model to the list of variables:
  1. Click **Add device/model parameter**. The Add Device/Model parameter screen appears, allowing you to specify which parameter is to be added. For example:

Choose whether to add either a device or model parameter.

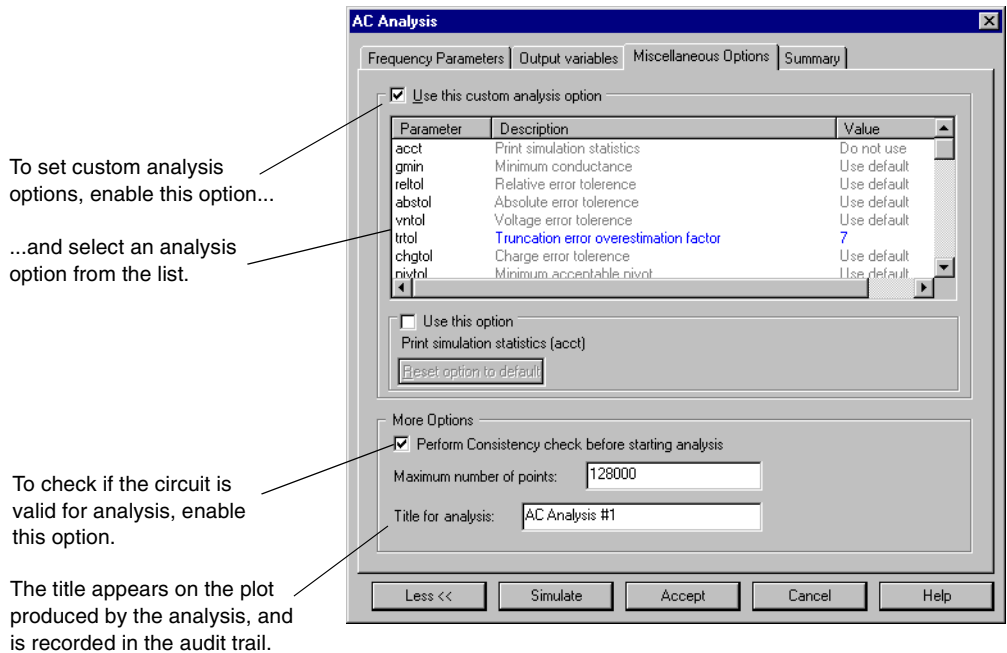
2. From the **Parameter Type** list, select whether you want to add a device parameter or model parameter. These let you set how various internal parameters of a component or model change during the analysis.
3. From the **Device Type** drop-down list, select a device type from the devices in the circuit.
4. From the **Name** drop-down list, select a specific instance of the device type.
5. From the **Parameter** drop-down list, select a parameter from all available device/model parameters. A brief description of the selected parameter appears in the **Description** list.
6. Click **OK** to add the selected parameter to the **Variables in Circuit** list. This variable can now be included in the analysis.
7. To save the addition, click **Accept**. To cancel the addition, click **Cancel**.

The parameter appears in the left-hand list under the Output Variables tab on the analysis screen, which is automatically set to show only device/model parameters.

- To delete a parameter added in this way, select it and click **Delete selected variables**.
- To show the values of all the components and models in the circuit at the end of the simulation, enable **Show all output parameters at end of simulation in the audit trail**.

### 8.3.4 The Miscellaneous Options Tab

The options on this tab provide you with additional flexibility, but do not require that you set them. Use this tab to set a title for the analysis results, to check if the circuit is valid for analysis, and to set custom analysis options. By default, the tab displays just the most commonly used fields. Click **More** to display the full screen displaying all available options as shown in the example below.



- To check if the circuit is a valid circuit for analysis, enable **Perform consistency check before starting analysis**. This option automatically identifies inconsistencies such as open capacitors, empty circuit files and ungrounded circuits.

Normally analyses run without further intervention. If an analysis does not perform as necessary, you may need to set custom analysis options.

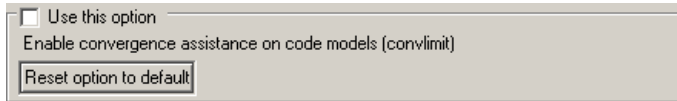
- To change the analysis title from its default, enter text in the **Title for Analysis** field.
- To set custom analysis options:

1. Enable **Use this custom analysis option**.

**Note** You should have a general knowledge of the SPICE simulation engine before altering the default settings under this option.

2. In the list of analysis options (analysis options shown in blue are set to the non-SPICE default value), select the analysis options whose value you want to change. The screen at the bottom of the list changes to reflect the choices available to you.

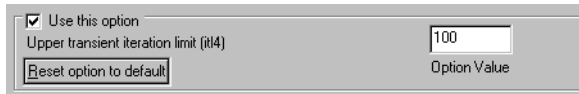
For those analysis options that you simply turn on or off, the screen looks like this:



☐ Use this option  
Enable convergence assistance on code models (convlimit)

To control whether or not the option takes effect, enable or disable the **Use this option** option. To reset the default value for the option, click **Reset option to default**.

For those analysis options for which you set values, the screen looks like this:



☒ Use this option  
Upper transient iteration limit (itl4)   
 Option Value

To use the default value, enable **Use default value**. To set a specific value, disable **Use default value** and type a value in the **Option Value** field. To re-set the value to its default setting, click **Reset option to default**.

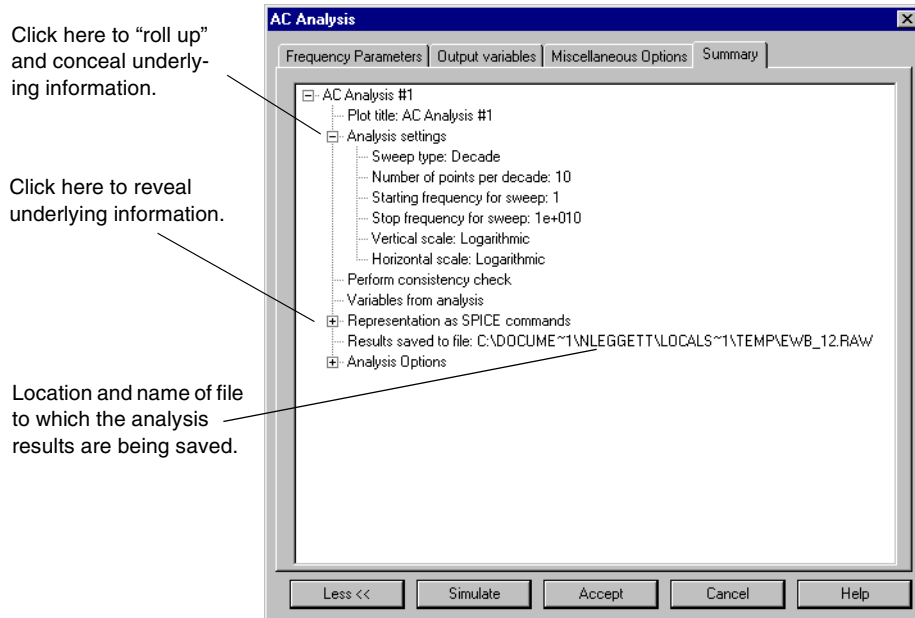
3. To save your changes, click **OK**. To close without saving, click **Cancel**.

For a complete list of the available analysis options, see “8.28 Analysis Options” on page 8-78.



### 8.3.5 The Summary Tab

This tab offers a quick overview of all the various settings for your analysis. It does not require you to set any options, but you can use it to view summary information about your analysis. By default, the tab displays just the most commonly used fields. Click **More** to display the full screen as shown in the example below.



You navigate through this display as you do with Windows Explorer. The "+" beside an item indicates that it has additional information underneath it, which can be revealed by clicking the "+". The "-" beside an item indicates that all its information is being revealed. That information can be hidden by clicking the "-".

This window also shows you the SPICE representation of your analysis options, as well as the name of the file to which the analysis results are being saved (.raw file). This file is used for Postprocessing.

## 8.3.6 Incomplete Analyses

For a variety of reasons, the simulator in Multisim is occasionally unable to complete a simulation or an analysis.

Multisim uses the modified Newton-Raphson method to solve nonlinear circuits. When a circuit includes nonlinear components, multiple iterations of a set of linear equations are used to account for the non-linearities. The simulator makes an initial guess at the node voltages, then calculates the branch currents based on the conductances in the circuit. The branch currents are then used to recalculate the node voltages and the cycle is repeated. This cycle continues until all of the node voltages and branch currents fall within user-defined tolerances, that is, convergence occurs. You can specify tolerances and iteration limits for the analysis through the analysis options described in “8.28 Analysis Options” on page 8-78.

If the voltages or currents do not converge within a specified number of iterations, an error message is produced and the simulation is aborted (typical messages include “Singular matrix,” “Gmin stepping failed,” “Source stepping failed” and “Iteration limit reached”).

## 8.4 DC Operating Point Analysis

### 8.4.1 About the DC Operating Point Analysis

The DC operating point analysis determines the DC operating point of a circuit. For DC analysis, AC sources are zeroed out and steady state is assumed, that is, capacitors are open circuits and inductors are short circuits. The results of DC analysis are usually intermediate values for further analysis. For example, the DC operating point obtained from DC analysis determines approximate linearized, small-signal models for any nonlinear components such as diodes and transistors for the AC frequency analysis.

**Assumptions** Digital components are treated as large resistances to ground. Results include node DC voltages and branch currents.

**Note** You can specify whether or not specific node trace widths are to be used for this type of analysis. For details, see “3.10.2 Modifying Node Numbers” on page 3-28.

## 8.4.2 Setting DC Operating Point Analysis Parameters

There are no analysis parameters to be set for this analysis.

## 8.4.3 Troubleshooting DC Operating Point Analysis Failures

DC operating point analysis may fail to converge for various reasons. The initial guesses for the node voltages may be too far off, the circuit may be unstable or bi-stable (there may be more than one solution to the equations), there may be discontinuities in the models or the circuit may contain unrealistic impedances.

**Note** All error messages generated during an analysis appear in the error log/audit trail. For more information about this function, see “8.21 Viewing the Analysis Results: Error Log/Audit Trail” on page 8-63.

Use the following techniques to solve many convergence problems and analysis failures. Before you proceed, identify which analysis is causing the problem (keep in mind that DC operating point analysis is often performed as the first step of other analyses). In each of the following solutions, begin with step 1, then continue performing the subsequent steps, in order, until the problem is solved.

1. Check the circuit topology and connectivity. Make sure that:
  - The circuit is correctly wired, and includes no dangling nodes or stray parts.
  - You haven't confused zeros with the letter O.
  - Your circuit has a ground node and every node in the circuit has a DC path to ground. Make sure no sections of your circuit are completely isolated from ground by transformers, capacitors, etc.
  - Capacitors and voltage sources are not in parallel.
  - Inductors and current sources are not in series.
  - All devices and sources are set to their proper values.
  - All dependent source gains are correct.
  - Your models/subcircuits have been correctly entered.
2. Increase the operating point analysis iteration limit to 200-300. This allows the analysis to go through more iterations before giving up.
3. Reduce the RSHUNT value by a factor of 100.
4. Increase the Gmin minimum conductance by a factor of 10.
5. Enable the option **Set to zero**.

## 8.5 AC Analysis

### 8.5.1 About the AC Analysis

In AC analysis, the DC operating point is first calculated to obtain linear, small-signal models for all nonlinear components. Then a complex matrix (containing both real and imaginary components) is created. To construct a matrix, DC sources are given zero values. AC sources, capacitors, and inductors are represented by their AC models. Nonlinear components are represented by linear AC small-signal models, derived from the DC operating point solution. All input sources are considered to be sinusoidal. The frequency of the sources is ignored. If the function generator is set to a square or triangular waveform, it will automatically switch internally to a sinusoidal waveform for analysis. AC analysis then calculates the AC circuit response as a function of frequency.

**Assumptions** Applied to an analog circuit, small-signal. Digital components are treated as large resistances to ground.

### 8.5.2 Setting AC Analysis Frequency Parameters

Before you perform the analysis, review your circuit and decide on the nodes for analysis. You can specify magnitude and phase of a source for AC frequency analysis through the placed component's parameters, as described in "3.8.3 Controlling How a Placed Component is Used in Analyses" on page 3-23.

AC Analysis Frequency parameters are set in the following screen:

Enter the start frequency for the sweep.

Enter the stop frequency for the sweep.

Select a sweep type: decade, linear, or octave. Defines how points to be calculated are distributed across the frequency range.

Enter the number of points to be calculated during the analysis. For a linear sweep type, use the number of points between start and end.

Select a vertical scale: linear, logarithmic, decimal, or octave. Vertical scale controls the y-axis scaling on the output graph.

**Note** To reset all parameters to their default values, click **Reset to default**.

The result of the AC frequency analysis is displayed in two parts: gain versus frequency and phase versus frequency.

If you have the Bode plotter connected to your circuit and activate the circuit, a similar analysis is performed.

## Setting AC Analysis Frequency Parameters for Normal Use

In most cases, you only need to:

- set a **Start Frequency (FSTART)**
- set a **Stop Frequency (FSTOP)**

## Setting AC Analysis Frequency Parameters for Advanced Use

In addition to the frequency range, you can also:

- choose a desired sweep type (decade, linear, or octave) from the **Sweep type** drop-down list
- enter the number of points to be calculated in the **Number of points per decade** field
- choose the vertical scale (linear, logarithmic, decimal or octave) from the **Vertical scale** drop-down list.

**Note** The greater the number of points calculated, the more accurate the results will be; however, the simulation speed will be adversely affected.

## 8.6 Transient Analysis

### 8.6.1 About the Transient Analysis

In transient analysis, also called time-domain transient analysis, Multisim computes the circuit's response as a function of time. Each input cycle is divided into intervals, and a DC analysis is performed for each time point in the cycle. The solution for the voltage waveform at a node is determined by the value of that voltage at each time point over one complete cycle.

DC sources have constant values; AC sources have time-dependent values. Capacitors and inductors are represented by energy storage models. Numerical integration is used to calculate the quantity of energy transfer over an interval of time.

if initial conditions are set to be...	then...
Automatically determined initial conditions	Multisim tries to start the simulation using the DC operating point as the initial condition. If the simulation fails, it uses user-defined initial conditions.
Calculate DC operating point	Multisim first calculates the DC operating point of the circuit, then uses that result as the initial conditions of the transient analysis.
Set to zero	the transient analysis starts from zero initial conditions.
User-defined	the analysis starts from initial conditions as set in the transient analysis screen.

**Assumptions** None.

## 8.6.2 Setting Transient Analysis Parameters

Before you perform the analysis, review your circuit and decide on the nodes for analysis. Transient Analysis Parameters are set in the following screen:

**Note** If your screen does not look like the example below, click **More** to display the full analysis screen.

Set initial conditions: Set to Zero, User-Defined, Calculate DC Operating Point, or Automatically Determine Initial Conditions.

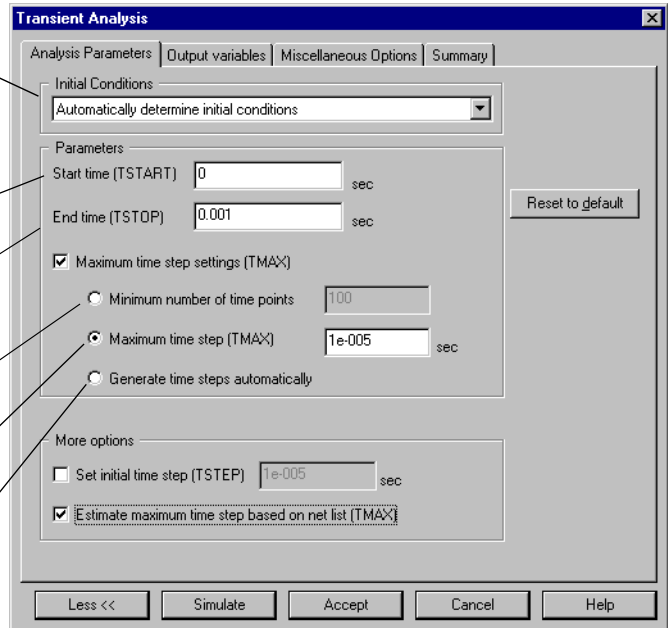
Start time of transient analysis must be greater than or equal to 0 and less than End time.

End time of transient analysis must be greater than Start time.

Click to enter minimum number of time points (number of points between start and stop times).

Click to enter the maximum time step the simulation can handle.

Click to generate time steps automatically.



The result of the transient analysis is a calculation of voltage versus time.

If you have the oscilloscope connected to your circuit and activate the circuit, a similar analysis is performed.

**Note** To re-set all parameters to their default values, click **Reset to default**.

### Setting Transient Analysis Parameters for Normal Use

The default settings are appropriate for normal use, providing the transient response of the selected output variables starting at time 0 seconds and stopping after 1 ms (0.001). You can, if you wish:

- change the start time by entering a value greater than or equal to 0 and less than the End time in the **Start time (TSTART)** field
- change the end time by entering a value greater than the Start time in the **End time (TSTOP)** field.

## Setting Transient Analysis Parameters for Advanced Use

For advanced use, you can:

- define the initial conditions at time 0 seconds by selecting an initial condition (Set to Zero, User-Defined, Calculate DC Operating Point, or Automatically Determine Initial Conditions) from the **Initial conditions** drop-down list

You can have the initial conditions set to zero, or you can use the steady state values of the circuit under analysis. During and/or after circuit construction, you can specify node voltages. These forced values can also be used as initial conditions for the analysis.

**Note** If you select “Automatically determine initial conditions”, Multisim will attempt to use steady state conditions to run the analysis. If this is unsuccessful, Multisim will set initial conditions to zero. If simulation is still not possible, Multisim will use the specified user-defined conditions.

- define the maximum time step to be taken by the simulation engine by enabling **Maximum time step (TMAX)** and entering the desired time step
- define the minimum time step to be taken by enabling **Minimum number of time points** and entering the desired number of points to be calculated

**Note** The value of TMAX is determined by dividing the interval between the specified analysis start and end times by the minimum number of time points specified.

- enable **Set Initial Time step (TSTEP)**, and enter a value less than the specified maximum time step value in the **Time step (TSTEP)** field. If possible, the size of the time steps taken during the simulation will begin with the initial timestep and will continue to increase to the value specified by the maximum time step.

### 8.6.3 Troubleshooting Transient Analysis Failures

If transient analysis is being performed (time is being stepped) and the simulator cannot converge on a solution using the initial time step, the time step is automatically reduced, and the cycle is repeated. If the time step is reduced too far, an error message (“Timestep too small”) is generated and the simulation is aborted. If this occurs, try one or more of the following:

- Check the circuit topology and connectivity. See step 1 of “8.3.3 Troubleshooting DC Operating Point Analysis Failures” on page 8-13.
- Set relative error tolerance to 0.01. By increasing the tolerance from 0.001 (0.1% accuracy), fewer iterations are required to converge on a solution and the simulation finishes much more quickly.
- Increase transient time point iterations to 100. This allows the transient analysis to go through more iterations for each time step before giving up.



- Reduce the absolute current tolerance, if current levels allow. Your particular circuit may not require resolutions down to 1  $\mu\text{V}$  or 1 pA. You should allow at least an order of magnitude below the lowest expected voltage or current levels of your circuit.
- Realistically model your circuit. Add realistic parasitics, especially junction capacitances. Use RC snubbers around diodes. Replace device models with subcircuits, especially for RF and power devices.
- If you have a controlled one-shot source in your circuit, increase its rise and fall times.
- Change the integration method to Gear. Gear integration requires longer simulation time, but is generally more stable than the trapezoid method.

## 8.7 Fourier Analysis

### 8.7.1 About the Fourier Analysis

Fourier analysis is a method of analyzing complex periodic waveforms. It permits any nonsinusoidal period function to be resolved into sine or cosine waves (possibly an infinite number) and a DC component. This permits further analysis and allows you to determine the effect of combining the waveform with other signals.

Given the mathematical theorem of a Fourier series, the period function  $f(t)$  can be written as follows:

$$f(t) = A_0 + A_1 \cos \omega t + A_2 \cos 2\omega t + \dots + B_1 \sin \omega t + B_2 \sin 2\omega t + \dots$$

where:

$A_0$	=	the DC component of the original wave
$A_1 \cos \omega t + B_1 \sin \omega t$	=	the fundamental component (has the same frequency and period as the original wave)
$A_n \cos n\omega t + B_n \sin n\omega t$	=	the $n^{\text{th}}$ harmonic of the function
A, B	=	the coefficients
$\frac{2\pi}{T}$	=	the fundamental angular frequency, or $2\pi$ times the frequency of the original periodic wave

Each frequency component (or *term*) of the response is produced by the corresponding harmonic of the periodic waveform. Each term is considered a separate source. According to the principle of superposition, the total response is the sum of the responses produced by each

term. Note that the amplitude of the harmonics decreases progressively as the order of the harmonics increases. This indicates that comparatively few terms yield a good approximation.

When Multisim performs Discrete Fourier Transform (DFT) calculations, only the second cycle of the fundamental component of a time-domain or transient response (extracted at the output node) is used. The first cycle is discarded for the settling time. The coefficient of each harmonic is calculated from the data gathered in the time domain, from the beginning of the cycle to time point “t”. That is set automatically and is a function of the fundamental frequency. This analysis requires a fundamental frequency matching the frequency of the AC source or the lowest common factor of multiple AC sources.

**Assumptions** None.

## 8.7.2 Setting Fourier Analysis Parameters

Before you perform the analysis, review your circuit and select an output node in the screen. The output variable is the node from which the analysis extracts the voltage waveform. Fourier Analysis Parameters are set in the following screen :

**Note** If your screen does not look like the example below, click **More**.

Set to the frequency of an AC source in your circuit. If you have several AC sources, use the lowest common factor of frequencies. Click to have Multisim estimate fundamental frequency.

Set the number of harmonics of the fundamental frequency that are calculated.

Click to set parameters for associated transient analysis.

Enable to display results as phase.

Set the amount of time during which sampling should occur (or use Edit Transient Analysis).

Enable to display results as bar graph. If not enabled, results display as linegraph.

Enable to normalize graphs. Normalized graphs are normalized against the 1st harmonic.

Enable to enter degree to be used when interpolating between points on simulation

Specify a sampling frequency.

Choose a vertical scale: linear, logarithmic, decibel, or octave.

Choose a display option: chart, graph, or chart and graph.

Fourier analysis produces a graph of Fourier voltage component magnitudes and, optionally, phase components versus frequency. By default, the magnitude plot is a bargraph but may be displayed as a line graph.

The analysis also calculates Total Harmonics Distortion (THD) as a percentage. The THD is generated by notching out the fundamental frequency, taking the square root of the sum of the squares of each of the  $n$  harmonics, and then dividing this number by the magnitude of the notched out fundamental frequency.

$$THD = [(\sum_{i=2}^n V_i^2) / V_1] \times 100 \%, \text{ where } V_1 \text{ is the magnitude of the } i^{\text{th}} \text{ harmonics.}$$

## Setting Fourier Analysis Parameters for Normal Use

For normal use, you just need to specify parameters for the following:

- frequency under test, either by clicking **Estimate** to have a value selected based on the AC sources in the circuit, or by entering a value in the **Frequency resolution (Fundamental frequency)** field. This value should be the lowest common factor for the frequencies present in the circuit.
- number of harmonics, by entering a value in the **Number of harmonics** field. You can specify the stopping time for sampling to avoid unwanted transient results prior to the circuit reaching steady-state operation.
- stopping time for sampling by enabling **Stopping time for sampling (TSTOP)** and entering a new stopping time for sampling. Although the Nyquist rate specifies only two times the highest frequency component being considered in the analysis as a suitable sampling rate, it is recommended that you specify a sampling frequency sufficient to obtain a minimum of 10 sampling points per period.
- enter a value in the **Sampling Frequency** field.

**Note** The sampling frequency should be equal to the frequency resolution (the number of harmonics plus one) times at least 10.

## Setting Fourier Analysis Parameters for Advanced Use

In addition to the basic procedures, you can also specify parameters for the following:

- degree of polynomial for interpolation, by enabling **Degree of polynomial for interpolation** and entering a value in the appropriate field. The higher the degree of polynomial the greater the accuracy of the results.
- results display format by doing one or all of the following:
  - choosing a vertical scale (linear, logarithmic, decibel or octave) from the **Vertical Scale** list
  - choosing a display option (chart, graph, or chart and graph) from the **Display** list
  - enabling **Display phase** to display results as phase
  - enabling **Display as bar graph** to display the results as a bar graph instead of a line-graph.
  - enabling **Normalize graphs** to normalize the results with respect to the frequency under test.
- transient analysis options by clicking **Edit transient analysis** to edit the transient analysis setup. For details, see “ Setting Transient Analysis Parameters for Normal Use” on page 8-17.

## 8.8 Noise Analysis

### 8.8.1 About the Noise Analysis

Noise is any undesired voltage or current appearing in the output. One common result of noise is “snowy” television reception caused by fluctuations across all frequencies of the television signal.

Multisim can model three different kinds of noise:

1. **Thermal noise** (also known as *Johnson*, or *white* noise) is temperature dependent and caused by the thermal interaction between free electrons and vibrating ions in a conductor. Its frequency content is spread equally throughout the spectrum.

The power of this generated noise is given by Johnson’s formula:

$$P = k \times T \times BW$$

where

$k$  = Boltzmann’s constant ( $1.38 \times 10^{-23} J/K$ )

$T$  = resistor temperature in Kelvin ( $T = 273 + \text{temperature in Celsius}$ )

$BW$  = frequency bandwidth of the system being considered

The thermal voltage could be represented by a mean-square voltage source in series with the resistor

$$e^2 = 4kTR \times BW$$

or the resistor mean-square current generator

$$i^2 = 4kTBW/R.$$

2. **Shot noise** is caused by the discrete-particle nature of the current carriers in all forms of semiconductors. It is the major cause of transistor noise. The equation for shot noise in a diode is:

$$i = (2q \times Idc \times BW)^{1/2}$$

where

$i$  = shot noise (RMS amperes)

$q$  = electron charge ( $1.6 \times 10^{-19}$  Coulomb)

$I_{dc}$  = DC current (A)

BW = bandwidth (Hz)

For all other devices, such as transistors, no valid formula is available. See the device manufacturer's data sheet. Shot noise and thermal noise are additive.

3. **Flicker noise** (also known as *excess* noise, *pink* noise, or *1/f* noise) is present in BJTs and FETs and occurs at frequencies below 1kHz. It is inversely proportional to frequency and directly proportional to temperature and DC current levels.

**Assumptions** Applied to an analog small-signal circuit. Non-conforming parts are ignored. Noise models for SPICE components are used.

## 8.8.2 Noise Analysis Example

Noise analysis is used frequently when diagnosing problems in communications systems. It calculates the noise contribution from each resistor and semiconductor at the specified output node. Each noise source is assumed not to be statistically correlated with the other noise sources in the circuit and their values are calculated independently. The *total output noise* at the output node is the Root Mean Square (RMS) sum of the individual noise contributions. The result is then divided by the gain between the input source (V0 in the sample circuit below) and the output source (node 13) to give the *equivalent input noise*. Equivalent input noise is the amount of noise that you would need to inject at the input source of a noiseless circuit to give an output noise level matching the noisy circuit. The total output noise voltage can be referenced to ground or to another node on the circuit. In this case, the total output noise is taken across these two nodes.

For example, if you choose V1 as the input noise reference source, and N1 as the output node, all the noise generators in the circuit have their noise contributions summed at N1 to provide the output noise. This value is then divided by the gain from V1 to N1 to get the equivalent input noise which, if injected from V1 into a noiseless circuit, would cause the previously calculated amount of output noise at N1.

### 8.8.3 Setting Noise Analysis Parameters

Before you perform the analysis, review your circuit and decide on an input noise reference source, output node and reference node.

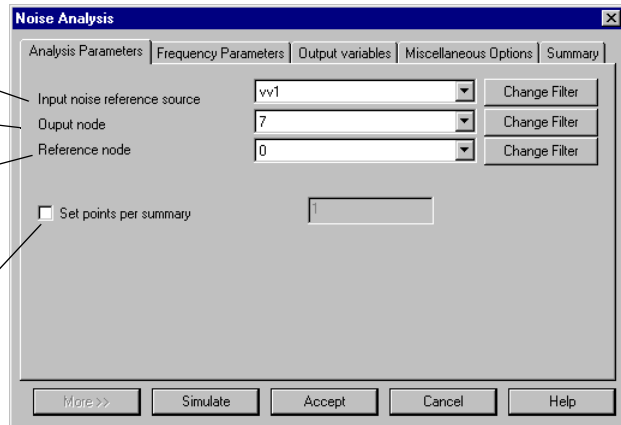
Noise Analysis Parameters are set in the following screen:

Choose an AC voltage source as input.

This is the node at which all noise contributions will be summed.

Reference voltage

When enabled, generates a trace of the selected component's noise contribution. Number of frequency steps is divided by the points per summary, which reduces the resolution of the output graph.



#### Setting Noise Analysis Parameters for Normal Use

Noise analysis performs an AC analysis to determine the noise. Noise analysis produces an output noise spectrum, an input noise spectrum and, optionally, a component contribution spectrum. When the analysis is finished, its results are displayed as a graph of voltage squared,  $V^2$ , versus frequency.

The thick trace identifies the total output noise at node 13, while the thin trace identifies the equivalent input noise at the AC source (V0). For this example, the output noise power is constant for all frequencies within the frequency sweep specified in the window.

On the analysis parameters tab, specify:

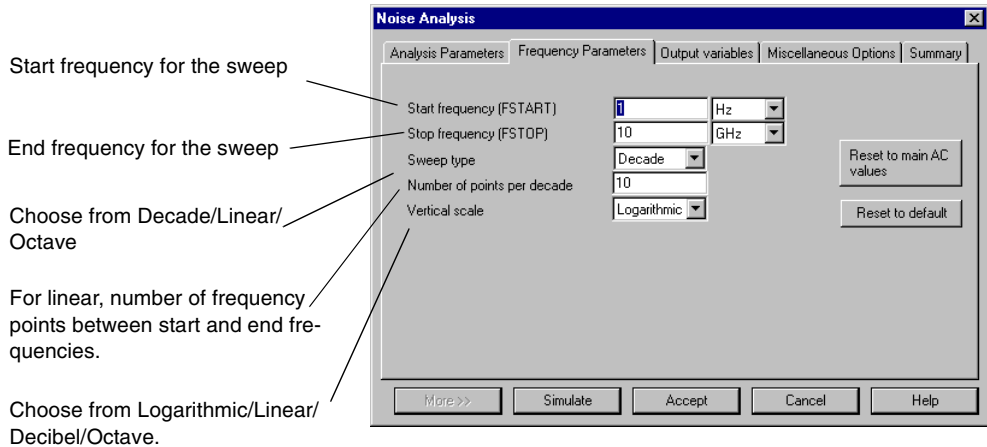
- input noise reference source
- output node
- reference node.

#### Setting Noise Analysis Parameters for Advanced Use

On the analysis parameters tab you can specify how often the noise contributions of each noise generating device are produced by enabling **Set points per summary** and entering a value.

## Setting Noise Analysis Frequency Parameters

Noise Analysis Frequency Parameters are set in the following screen:



## Setting Noise Analysis Frequency Parameters for Normal Use

The default settings on the Frequency Parameters tab are appropriate for most cases. You just need to define a frequency range by typing a value in the **Start Frequency (FSTART)** field and in the **Stop Frequency (FSTOP)** field.

- To copy the settings from the current AC analysis to this analysis, click **Reset to main AC values**.

Once the required variables are selected and the frequency range has been defined, you can then run the analysis.

## Setting Noise Analysis Frequency Parameters for Advanced Use

On the Frequency Parameters tab you can also set:

- sweep type, by choosing the desired sweep type (decade, linear, or octave) from the **Sweep type** drop-down list. The sweep type defines how the points to be calculated are distributed across the frequency range.
- the number of points to be calculated during the analysis, by entering a value in the **Number of points per decade** field.

**Note** The greater the number of points calculated, the more accurate the results will be, however, the simulation speed will be adversely affected.



- the format of the analysis results by choosing the desired scale (linear, logarithmic, decimal, or octave) from the **Vertical scale** drop-down list.
- Note** Click **Reset to Default** to reset all parameters on the Frequency Parameters tab to their default values.

## 8.9 Distortion Analysis

### 8.9.1 About the Distortion Analysis

Signal distortions are usually the result of gain nonlinearity or phase nonuniformity in a circuit. Nonlinear gain causes *harmonic distortion*, while nonuniform phase causes *intermodulation distortion*.

Distortion analysis is useful for investigating small amounts of distortion that are normally unresolvable in transient analysis. Multisim simulates harmonic distortion and intermodulation distortion products for analog small-signal circuits. If the circuit has one AC frequency, the analysis determines the complex values of the second and third harmonics at every point in the circuit. If the circuit has two AC frequencies, the analysis finds the complex values of the circuit variables at three different frequencies: at the sum of the frequencies, at the difference of the frequencies, and at the difference between the lowest and highest frequencies of the second harmonic.

The analysis carries out a small-signal distortion analysis of the circuit. A multi-dimensional Volterra analysis is carried out using a multi-dimensional Taylor series to represent the nonlinearities at the operating point. The series expansion uses terms of up to the third order.

**Assumptions** Analog circuit, small-signal. Non-conforming parts are ignored. Distortion models for SPICE components are used.

### 8.9.2 Setting Distortion Analysis Parameters

Before you perform the analysis, review your circuit and decide on one or two sources and one or more nodes for analysis. You can also change the magnitude and phase of sources for distortion analysis through the placed component's parameters, as described in "3.8.3 Controlling How a Placed Component is Used in Analyses" on page 3-23.

Distortion Analysis Parameters are set in the following screen:

Enter the start frequency for the sweep.

Enter the stop frequency for the sweep.

Select a sweep type: decade, linear, or octave.

Enter number of points. For linear, number of frequency points between start and end.

Select a vertical scale: logarithmic, linear, decibel, or octave.

When enabled, if there are signals of two frequencies (F1 and F2), then F2 is set to this ratio multiplied by the start frequency while F1 sweeps. Must be greater than 0.0 and less than 1.0.

The screenshot shows the 'Distortion Analysis' dialog box with the 'Analysis Parameters' tab selected. The fields are as follows:

- Start frequency (FSTART): 1 Hz
- Stop frequency (FSTOP): 10 GHz
- Sweep type: Decade
- Number of points per decade: 10
- Vertical scale: Logarithmic
- F2/F1 ratio: 0.1 (with an unchecked checkbox)

Buttons include 'Reset to main AC values', 'Reset to default', 'More >>', 'Simulate', 'Accept', 'Cancel', and 'Help'.

If the F2/F1 ratio is disabled, the analysis calculates harmonic distortion of one frequency which is swept according to the screen entries. If F2/F1 ratio is enabled, a spectral analysis is performed. Each independent source in the circuit may potentially have two (superimposed) sinusoidal inputs for distortion at frequencies F1 and F2.

If the F2/F1 ratio is disabled, the analysis produces a graph of the second and third harmonics, displaying them on a Distortion tab in the Grapher. If the F2/F1 ratio is enabled, the analysis produces a graph of the selected voltage or branch current at the intermodulation frequencies,  $F1 + F2$ ,  $F1 - F2$ ,  $2 * F1 - F2$ , versus the swept frequency, F1. These graphs appear in the IM Distortion tab of the Grapher. For more information on the Grapher, see “8.22 Viewing the Analysis Results: Grapher” on page 8-64.

## Setting Distortion Analysis Parameters for Normal Use

The default settings on the Analysis Parameters tab are appropriate for most cases. You need only define the frequency range by typing a value in the **Start Frequency (FSTART)** field and in the **Stop Frequency (FSTOP)** field.

**Note** Click **Reset to main AC values** to set the Analysis Parameters to the values defined for the AC frequency analysis.

## Setting Distortion Analysis Parameters for Advanced Use

Using the Analysis Parameters tab, you can set:

- the sweep type, by choosing the desired sweep type (decade, linear, or octave) from the **Sweep type** drop-down list. The sweep type defines how the points to be calculated are distributed across the frequency range.
- the number of points to be calculated during the analysis, by entering a value in the **Number of points per decade** field.

**Note** The greater the number of points calculated, the more accurate the results will be; however, the simulation speed will be adversely affected.

- the type of vertical scale, by choosing the desired scale (linear, logarithmic, decimal, or octave) from the **Vertical scale** drop-down list.

When you enable **F2/F1 ratio**, circuit variables are calculated at  $(F_1+F_2)$ ,  $F_1-F_2$ , and  $(2F_1)-F_2$ .

➤ To set the F2/F1 ratio for multiple AC source circuits:

1. Enable **F2/F1 ratio**.
2. Enter a value in the appropriate box. The value must be greater than zero and less than one.

**Note** F1 is swept according to the values specified as the start and stop frequencies for the analysis. F2 is kept at a single frequency as F1 sweeps. The value of F2 is determined by multiplying the F2/F1 ratio by the start frequency (FSTART) specified.

**Note** Click **Reset to Default** to reset all parameters on the Analysis Parameters tab to their default values.

## 8.10 DC Sweep Analysis

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

The DC sweep analysis computes the DC operating point of a node in the circuit for various values of one or two DC sources in the circuit.

Using a DC sweep analysis, you can quickly verify the DC operating point of your circuit by simulating it across a range of values for one or two DC voltage or current sources. The effect is the same as simulating the circuit several times, once for each different value or pair of values. You control the source values by choosing start, stop and increment values in the Analysis Parameters tab of the DC Sweep screen.

**Assumptions** Digital components are treated as large resistances to ground.

## 8.10.1 Setting DC Sweep Analysis Parameters

Before setting analysis parameters, review your circuit and decide on one or two DC sources to sweep, and a node for analysis.

DC sweep analysis plots the appropriate curves sequentially. If only one source is being swept, a curve of the output node value versus source value is traced. If two sources are swept, the number of curves equals the number of points for the second source. Each curve represents the output node value versus the first source value while the second source value is held at each of its sweep values.

DC Sweep Analysis parameters are set on the following screen:

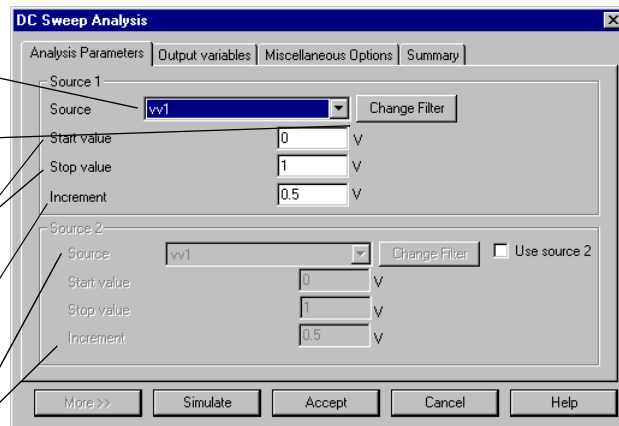
Choose the source for the sweep.

Click to filter the items in the Source list.

The values at which the analysis will stop and start

The values by which each sweep will increase

As above, but for second source



### Setting DC Sweep Analysis Parameters for Normal Use

For normal use, you only need to set:

- the source for the sweep, by choosing from the **Source** drop-down list under the **Source 1** options
- a starting value for the sweep, by entering it in the **Start Value** field
- a stop value for the sweep, by entering it in the **Stop Value** field
- an increment value for the sweep, by entering it in the **Increment** field.

The analysis will calculate the circuit's bias point values beginning with the specified start value. The **Increment** value will then be added to the start value and the circuit variables will be recalculated. The **Increment** value is added again and the process continues until the stop value is reached.

## Setting DC Sweep Analysis Parameters for Advanced Use

You can filter the variables displayed to include internal nodes (such as nodes inside a BJT model or inside a SPICE subcircuit), open pins, as well as output variables from any sub-modules contained in the circuit.

- To filter the variables displayed:
  1. Click **Change Filter**. The Filter Nodes screen appears.
  2. Enable one or more settings.
  3. Click **OK**.

## 8.11 DC and AC Sensitivity Analyses

### 8.11.1 About the Sensitivity Analyses

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

Sensitivity analyses help to identify the components which affect a circuit's DC bias point the most. This will focus efforts on reducing the sensitivity of the circuit to component variations and/or drift, or it may provide evidence that a design is too conservative and that less expensive components, with more variation and/or drift, may be used.

Sensitivity analyses calculate the sensitivity of an output node voltage or current with respect to the parameters of all components (DC sensitivity) or one component (AC sensitivity) in your circuit. Sensitivity analyses produce the relevant parameters with their original values and their sensitivities. Sensitivity is expressed as the change in output per unit change of input both in values and percentages.

Both analyses calculate the change produced in an output voltage or current by perturbing each parameter independently. The results of the DC sensitivity analysis are sent to a table, whereas the AC sensitivity analysis plots the AC graphs for each parameter of the component.

For the DC sensitivity analysis, a DC analysis is first performed to determine the DC operating point of the circuit. Then the sensitivity of each output for all of the device values (as well as model parameters) is calculated.

**Assumptions** Applied to an analog circuit, small-signal. Models are linearized.

## 8.11.2 Sensitivity Analyses Example

Consider the following example.

**Note** If the flat line overlapping X-axis gets displayed, this means that the output voltage/ current is not affected by the chosen component value.

The DC sensitivity analysis generated a report (shown below) of the output voltage at node 12 sensitivity with respect to all components and their parameters. (Alternatively, you can choose to run a DC sensitivity analysis of the current source.)

What does the DC report mean? In the first line of the report, the change (increase) of one unit of flicker noise (AF) of the Zener diode (D10) will cause the decrease of the output voltage by 1.583e-012 V. Note that the sensitivity number is negative: -1.582e-012. This indicates that the increase of one unit of a device parameter will cause the decrease of voltage. Each line is to be similarly interpreted.

Component	Sensitivity (V/unit)
D10:af	-1.582e-012
D10:bv	0.00046414
D10:eg	-1.4252e-012
D10:fc	-3.1639e-012
D10:ibv	-0.0002668
D10:is	-38.487
D10:m	-4.7507e-012
D10:n	-1.582e-012
D10:rs	2.3913e-006
D10:tnom	-6.5352e-014
D10:vj	-2.1093e-012
D10:xti	-5.2732e-013
D10_area	-2.2575e-005
D10_temp	-8.6695e-008
D11:af	-3.9034e-013
D11:bv	1.4958

Component	Sensitivity (V/unit)
D11:eg	-3.5165e-013
D11:fc	-7.8067e-013
D11:ibv	-1.5478
D11:is	954.04
D11:m	-1.1722e-012.....

The AC small signal sensitivity is calculated if AC analysis is selected. For AC sensitivity only, the analysis measures the sensitivity of the voltage or current with respect to the parameters of the chosen component.

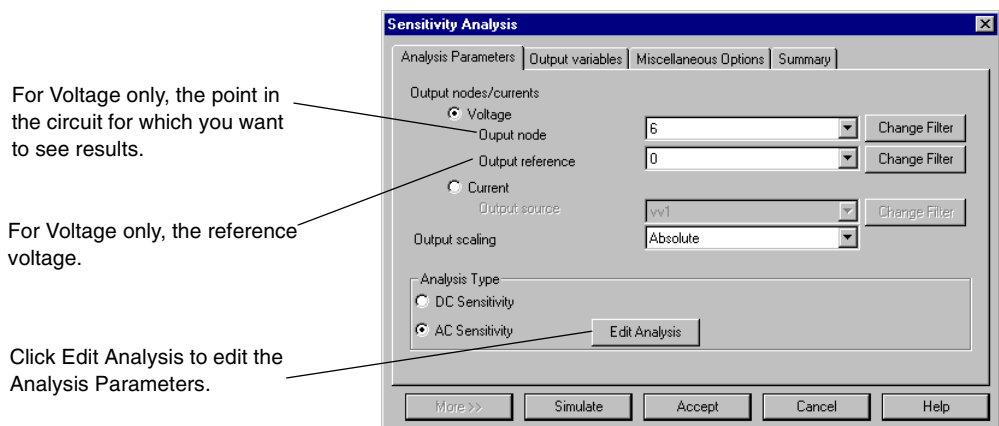
In the example, the chosen component is BJT transistor Q18. The analysis calculates the AC sensitivity of the output voltage at node 12 with respect to all parameters of Q18 transistor.

**Note** The AC sensitivity analysis does not generate any data (fails) if resistors are chosen as components. The transistors seem to be very unpredictable as well.

### 8.11.3 Setting Sensitivity Analysis Parameters

Before you perform the analysis, review your circuit and decide on an output voltage or current. For an output voltage, choose nodes on either side of the circuit output. For an output current, choose a source.

Sensitivity Analysis Parameters are set in the following screen:



## Setting Sensitivity Analysis Parameters for Normal Use

For normal use, you only need to specify:

- the output node or source current to use during the analysis, by enabling **Voltage** and choosing an output node from the **Output node** drop-down list or by enabling **Current** and choosing a source current from the **Output reference** drop-down list.
- the type of sensitivity analysis to run by enabling either **DC Sensitivity** or **AC Sensitivity**.

## Setting Sensitivity Analysis Parameters for Advanced Use

From the **Output scaling** drop-down list, you can select the type of output scaling: absolute or relative. You can also click **Change filter** to access the Filter Node screen. Using this screen, you can include internal nodes, open pins, as well as output variables from any submodules in the circuit.

When performing an AC sensitivity analysis, you can also edit the AC frequency Analysis Parameters by clicking **Edit Analysis**. The Frequency Parameters tab appears. You can then set the sweep type, the number of points, and the vertical scale.

# 8.12 Parameter Sweep Analysis

## 8.12.1 About the Parameter Sweep Analysis

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



Using parameter sweep analysis, you can quickly verify the operation of your circuit by simulating it across a range of values for a component parameter. The effect is the same as simulating the circuit several times, once for each different value. You control the parameter values by choosing a start value, an end, the sweep type, and an increment value.

You may perform three types of sweeps: DC Operating Point, Transient Analysis, and AC Frequency Analysis.

**Assumptions** See the assumptions for the selected analysis: DC operating point analysis described in “8.3 DC Operating Point Analysis” on page 8-12, transient analysis described in “8.5 Transient Analysis” on page 8-16, or AC frequency analysis described in “8.3 DC Operating Point Analysis” on page 8-12.

You will find that some components have more parameters that can be varied than other components. The number of varied parameters depends on the model of the component. That is, active components such as op-amps, transistors, diodes and others have more parameters available to perform a sweep than passive components such as resistors, inductors and capacitors.



itors. For example, an inductor has only inductance available as a parameter for analysis, whereas a diode model contains a set of approximately 25 parameters such as Saturation current, Ohmic resistance, Junction potential, Break Down voltage and others available for analysis.

## 8.12.2 Setting Parameter Sweep Analysis Parameters

The behavior of a circuit is affected when certain parameters in specific components change. Before you perform the analysis, review your circuit and decide on a component and parameter to sweep, and a node for analysis.

Parameter Sweep Analysis Parameters are set in the following screen:

**Note** If your screen does not look like the example below, click **More** to display the full analysis screen.

The screenshot shows the 'Parameter Sweep' dialog box with the following annotations:

- Choose sweep parameter: Device Parameter or Model Parameter.** Points to the 'Sweep Parameter' dropdown menu, which is set to 'Device Parameter'.
- Dictates how Multisim calculates the interval between the stop and start values. Choose from Decade, Octave, Linear, or List.** Points to the 'Sweep Variation Type' dropdown menu, which is set to 'List'.
- For List sweep only. A list of values to sweep over. Items in the list must be separated by spaces, commas or semicolons.** Points to the 'Values' text box, which contains '0,1'.
- Choose the type of device to sweep: i.e. BJT, Capacitor or Inductor.** Points to the 'Device' dropdown menu, which is set to 'BJT'.
- Enter the reference ID of the component to sweep.** Points to the 'Name' dropdown menu, which is set to 'q5:u1.1y'.
- Choose the device parameter of the component to be swept.** Points to the 'Parameter' dropdown menu, which is set to 'off'.
- Click to edit parameters of the chosen analysis.** Points to the 'Edit Analysis' button.
- Choose DC Operating Point, AC Analysis, Transient Analysis, Nested Sweep.** Points to the 'Analysis to sweep' dropdown menu, which is set to 'Transient analysis'.
- If not enabled, each trace appears on a separate plot.** Points to the 'Group all traces on one plot' checkbox, which is checked.

The dialog box also includes tabs for 'Analysis Parameters', 'Output variables', 'Miscellaneous Options', and 'Summary'. At the bottom are buttons for 'Less <<', 'Simulate', 'Accept', 'Cancel', and 'Help'.

Parameter sweep analysis plots the appropriate curves sequentially. The number of curves is dependent on the type of sweep as shown below:

Type of Sweep	Curves
Linear	The number of curves is equal to the difference between the start and end values divided by the increment step size.
Decade	The number of curves is equal to the number of times the start value can be multiplied by ten before reaching the end value.
Octave	The number of curves is equal to the number of times the start value can be doubled before reaching the end value.

### Setting Parameter Sweep Analysis Parameters for Normal Use

For normal use, you only need to:

- select a sweep parameter by choosing a parameter type (Device or Model) from the **Sweep Parameter** drop-down list, then entering information in the **Device**, **Name**, and **Parameter** fields

**Note** A brief description of the parameter appears in the **Description** field and the present value of the parameter is displayed in the **Present Value** field.

- set the sweep variation type by choosing a type of distribution (linear, decade, or octave) from the **Sweep Variation Type** drop-down list.
- select the analysis to sweep by choosing from the **Analysis to sweep** drop-down list.

Optionally, you can set the analysis parameters by clicking **Edit Analysis**. Under the Analysis Parameters screen that appears:

- enter a start and stop value in the **Start time (TSTART)** and **End time (TSTOP)** fields
- enter the number of points in the **Minimum number of time points** field. The increment value will be calculated and automatically set.

**Note** If the analysis is unedited, the last values set for the analysis will be used. If the analysis has not been run previously, the default values will apply.

- if you want to sweep other than the list, type the desired parameter values, separated by a space, in the **Values** field

## Setting Parameter Sweep Analysis Parameters for Advanced Use

You can use the Analysis Parameters screen to select different sweep variation types while setting Analysis Parameters. You can also perform nested sweeps, combining various levels of device/model parameter sweeps.

- To set the sweep variation type and specify the range and number of points to sweep:
  1. Choose the type of distribution (linear, decade, or octave) from the **Sweep Variation Type** drop-down list.
  2. Click **Edit Analysis**. A new Analysis Parameters screen appears.
  3. Enter a start value for the sweep in the **Start time (TSTART)** field.
  4. Enter a stop value for the sweep in the **End time (TSTOP)** field.
  5. Enter the number of points in the **Minimum number of time points** field. The increment value will be calculated and automatically set.

## 8.13 Temperature Sweep Analysis

### 8.13.1 About the Temperature Sweep Analysis

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



Using temperature sweep analysis, you can quickly verify the operation of your circuit by simulating it at different temperatures. The effect is the same as simulating the circuit several times, once for each different temperature. You control the temperature values by choosing start, stop and increment values.

You may perform three types of sweeps: DC Operating Point, Transient Analysis, and AC Frequency Analysis.

**Assumptions** See the assumptions for the selected analysis: DC operating point analysis described in “8.3 DC Operating Point Analysis” on page 8-12, transient analysis described in “8.5 Transient Analysis” on page 8-16, or AC frequency analysis described in “8.3 DC Operating Point Analysis” on page 8-12.

Temperature sweep analysis affects only components whose model includes temperature dependency, such as:

- Virtual Resistor
- 3 - Terminal Depletion N-MOSFET
- 3 - Terminal Depletion P- MOSFET
- 3 - Terminal Enhancement N- MOSFET

- 3 - Terminal Enhancement P- MOSFET
- 4 - Terminal Depletion N- MOSFET
- 4 - Terminal Depletion P- MOSFET
- 4 - Terminal Enhancement N- MOSFET
- 4 - Terminal Enhancement P- MOSFET
- Diode
- LED
- N-Channel JFET
- NPN Transistor
- P-Channel JFET
- PNP Transistor

## 8.13.2 Setting Temperature Sweep Analysis Parameters

Before you perform the analysis, review your circuit and decide on a node for analysis.

Temperature Sweep Analysis Parameters are set in the following screen:

**Note** If your screen does not look like the example below, click **More** to display the full analysis screen.

Displays the parameter type: Temperature

Dictates how Multisim calculates the interval between the stop and start values. Choose from Decade, Octave, Linear, or List.

For List sweep only. A list of values to sweep over. Items in the list must be separated by spaces, commas or semicolons.

If not enabled, each trace appears on a separate plot.

Choose DC Operating Point, AC Analysis, Transient Analysis.

Click to edit parameters of the chosen analysis.

Temperature sweep analysis plots the appropriate curves sequentially. The number of curves is dependent on the type of sweep, as shown below.

Type of Sweep	Curves
Linear	The number of curves is equal to the difference between the start and end values divided by the increment step size.
Decade	The number of curves is equal to the number of times the start value can be multiplied by ten before reaching the end value.

Type of Sweep	Curves
Octave	The number of curves is equal to the number of times the start value can be doubled before reaching the end value.

### Setting Temperature Sweep Analysis Parameters for Normal Use

You can use the Analysis Parameters tab to define the temperature values to be swept, and the type of analysis to be run at the various swept temperatures. You can also edit the analysis.

The **Sweep Parameter** field is set to **Temperature** by default and the default setting for the **Sweep Variation Type** is **List**. You need only enter the desired list of temperatures to sweep, and the type of analysis to be performed.

- To specify the list of temperatures and the analysis:
  1. Enter the list of temperatures (separated by a space) in the **Values** field.
  2. Choose the analysis to be performed by choosing from the **Analysis to sweep** drop-down list.
  3. Click **Edit Analysis** to specify the Analysis Parameters.

**Note** If the analysis is unedited, the last values set for the analysis will be used. If the analysis has not been run previously, the default values will apply.

### Setting Temperature Sweep Analysis Parameters for Advanced Use

You can select different sweep variation types while setting Analysis Parameters.

- To set the sweep variation type and specify the range and number of points to sweep:
  1. Choose the type of distribution (linear, decade, or octave) from the **Sweep Variation Type** drop-down list.
  2. Click **Edit Analysis**. A new Analysis Parameters screen appears.
  3. Enter a start value for the sweep in the **Start time (TSTART)** field.
  4. Enter a stop value for the sweep in the **End time (TSTART)** field.
  5. Enter the number of points in the **Minimum number of points** field. The increment value will be calculated and automatically set.
  6. Choose the analysis to sweep from the **Analysis to sweep** drop-down list.

## 8.14 Transfer Function Analysis

### 8.14.1 About the Transfer Function Analysis



**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

Transfer function analysis calculates the DC small-signal transfer function between an input source and two output nodes (for voltage) or an output variable (for current) in a circuit. It also calculates input and output resistances. Any nonlinear models are first linearized based on the DC operating point and then small-signal analysis is performed. The output variable can be any node voltage, while the input must be an independent source defined somewhere in the circuit.

**Assumptions** Analog circuit, linear models. Models are linearized.

The DC small signal gain is the derivative of the output with respect to the input at the DC bias-point (and zero frequency). For example:

$$\frac{dV_{OUT}}{dV_{IN}}$$

The input and output resistance of a circuit refers to the “dynamic” or small-signal resistance at the input or output. Mathematically, small-signal DC resistance is the derivative of the input voltage with respect to the input current at the DC bias-point (and zero frequency). The following is an expression for input resistance:

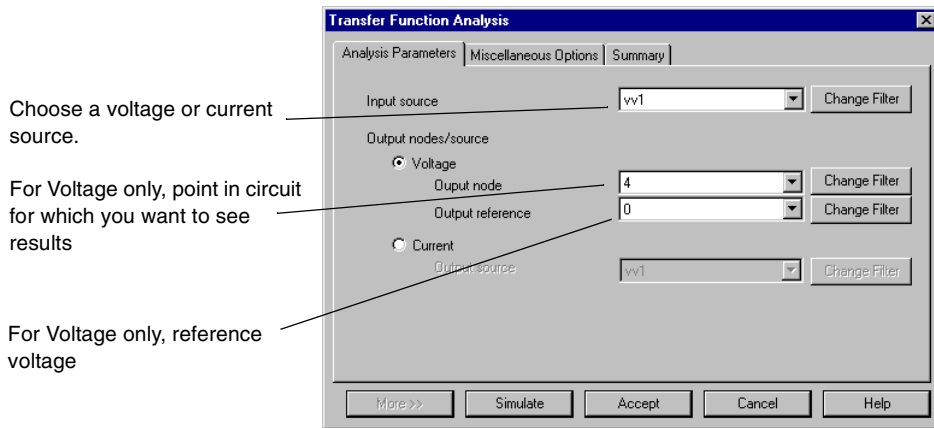
$$\frac{dV_{IN}}{dI_{IN}}$$

In Multisim, the results of the Transfer function analysis produce a chart showing the ratio of the output to the input signal, the input resistance at the input source node and the output resistance across the output voltage nodes.

## 8.14.2 Setting Transfer Function Analysis Parameters

Before you perform the analysis, review your circuit and decide on an output node, a reference node and an input source.

Transfer Function Analysis Parameters are set in the following screen:



Transfer function analysis produces a chart showing the transfer function (output/input), input resistance at the input source and output resistance across the output voltage nodes or at the output variable.

### Setting Transfer Function Analysis Parameters for Normal Use

For normal use, you only need to:

- choose an input source from the **Input source** drop-down list
- enable **Voltage** and select an output node from the **Output node** drop-down list and an output reference node (usually ground or node 0) from the **Output reference** drop-down list
- enable **Current** and select a source current from the **Output source** drop-down list

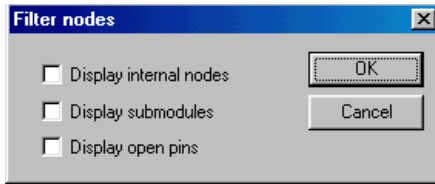
For more advanced use, a source current as well as output node voltage can be used for this analysis.

### Setting Transfer Function Analysis Parameters for Advanced Use

You can filter the variables displayed to include internal nodes (such as nodes inside a BJT model or inside a SPICE subcircuits), open pins, as well as output variables from any sub-modules contained in the circuit. Filtering the variables shortens the list of results.



- To filter the variables displayed:
  1. Click **Change Filter**. The Filter Nodes screen appears:



2. Enable one or more settings.
3. Click **OK**.

## 8.15 Worst Case Analysis

### 8.15.1 About the Worst Case Analysis

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



Worst case analysis is a statistical analysis that lets you explore the worst possible effects of variations in component parameters on the performance of a circuit.

The first simulation is performed with nominal values. Then, a sensitivity run (AC or DC) is performed. This allows the simulator to calculate the sensitivity of the output waveform (voltage or current) with respect to each parameter. Expressing a specific component's sensitivity as a *negative* number yields the component's minimum value for the worst case analysis. For example, if the sensitivity of resistor R1 is -1.23V/Ohm, then the minimum sensitivity value of this component is calculated with the following formula:

$$R1_{min} = (1 - Tol) \times R1_{nom}$$

where

$R1_{min}$  = minimum value of the R1 resistor

Tol = tolerance specified in the dialog box divided by 100%

$R1_{nom}$  = nominal value of the resistor R1

Expressing a specific component’s sensitivity as a *positive* number yields the component’s maximum value for the worst case analysis. The maximum value is calculated with the following formula:

$$R2_{max} = (1 + Tol) \times R2_{nom} .$$

Once all the sensitivities have been obtained, a final run provides the worst case analysis result.

Data from the worst case simulation is gathered by collating functions. A collating function acts as a highly selective filter by allowing only one datum to be collected per run.

The six collating functions are:

This collating function...	Captures...
Maximum voltage	the values of the Y-axis maxima.
Minimum voltage	the values of the Y-axis minima.
Frequency at maximum	the X value where the Y-axis maxima occurred.
Frequency at minimum	the X value where the Y-axis minima occurred.
Rising edge frequency	the X value the first time the Y value rises above the user-specified threshold.
Falling edge frequency	the X value the first time the Y value falls below the user-specified threshold.

**Assumptions** Applied to an analog circuit, DC and small-signal. Models are linearized.

### Setting Worst Case Analysis Tolerance Parameters

Before you perform the analysis, review your circuit and decide on an output node.

In the Model tolerance list tab, choose which tolerance parameters are to be used. You can do this using any of the following methods:

- To edit a tolerance in the list, select it and click **Edit selected tolerance**. The tolerance’s current variable settings appear. Modify the variables as desired and click **OK** to save.
- To delete a tolerance from the list, select it and click **Delete tolerance entry**.
- To manually add a tolerance, click **Add a new tolerance**.

The Tolerance screen appears:

Choose type of sweep to be performed: Model Parameter or Device Parameter.

When selected, the current value and a description of the parameter appear.

Choose the type of distribution: Gaussian or Uniform.

Choose Unique (each random number generation is distinct), or a numbered lot (the same random number generation for various parameters).

Choose Absolute to enter a value, or Percent to vary the parameter by the specified percentage of its stated value.

Enter a percentage value, or a set value depending on the tolerance type selected.

The Tolerance dialog box is shown with the following settings:

- Parameter Type:** Model Parameter
- Parameter:**
  - Device Type: BJT
  - Name: qideal.xu1.1y
  - Parameter: npr
  - Description: NPN type device
- Tolerance:**
  - Distribution: Gaussian
  - Lot number: Unique
  - Tolerance Type: Percent
  - Tolerance value: 0

Buttons at the bottom: Accept, Cancel, Help.

Enter the desired variables in the appropriate fields.

## 8.15.2 Setting Worst Case Analysis Parameters

Worst Case Analysis Parameters are set in the following screen:

Choose DC Operating Point or AC Analysis. For details on either parameter, see “8.3 DC Operating Point Analysis” on page 8-12.

Click to edit selected parameter.

Choose an output variable.

Choose a function: MAX, MIN, RISE\_EDGE, FALL\_EDGE.

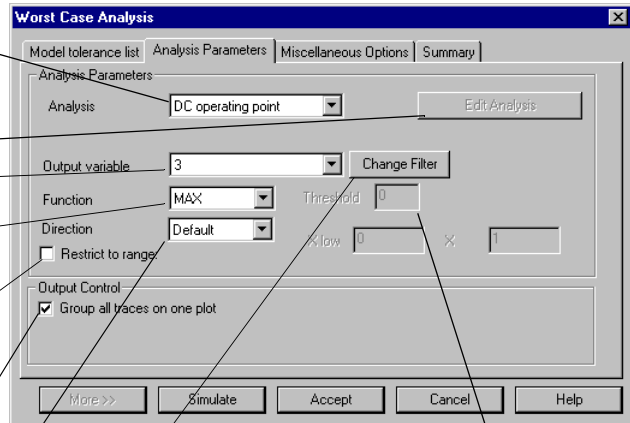
Enable to restrict the x-axis range of the analysis being run. Set X low (default is 0) and X high (default is 1).

Click to group all traces on a single plot.

Choose a direction: Default, Low or High.

Click to change the filter that affects the list of possible output variables.

From RISE\_EDGE and FALL\_EDGE functions only: enter a value for the threshold voltage.



For DC circuits, the worst case analysis generates a plot of the circuit’s possible output voltages ranging from the nominal specification value (1 on the x-axis) to the worst case value (2 on the x-axis). A list of the components and their worst case values appears in tabular form.

For AC circuits, the worst case analysis generates separate plots for the nominal and worst case runs. A list of the components and their worst case values appears in tabular form.

## 8.16 Pole Zero Analysis

## 8.16.1 About the Pole Zero Analysis



If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

Pole zero analysis will only work properly if there are passive components in the circuit. Pole zero analysis finds the poles and zeros in the small-signal AC transfer function of a circuit. The analysis begins by calculating the DC operating point and determining the linearized small-signal models for all nonlinear devices. From the resulting circuit, the analysis finds the poles and zeros of the transfer function.

Pole zero analysis is useful in determining the stability of electronic circuits. When designing circuits, it is important to know whether the output signal remains bounded or increases indefinitely following the application of an input signal. An unbounded output could damage or destroy the circuit; therefore, it is important to know if the circuit can accommodate the expected output before applying the input signal. A circuit is said to have *bounded input-bounded output* (BIBO) stability if any bounded input results in bounded output. BIBO stability can be determined by examining the poles of the transfer function of the circuit. Your circuit should have poles with negative real parts; otherwise, it could have an unintentionally large and potentially damaging response to certain frequencies.

Transfer functions are a convenient way of expressing the behavior of analog circuits in the frequency domain. A transfer function is ratio of the LaPlace Transform of the output signal to the LaPlace Transform of the input signal in a circuit. The LaPlace Transform of the output signal is commonly referred to as  $V_o(s)$  and the LaPlace Transform of the input signal is referred to as  $V_I(s)$  where the parameter  $s = j\omega$ , or more commonly,  $s = j2\pi f$

A transfer function is in general a complex quantity the magnitude of which gives the magnitude response (or transmission) and the angle of which gives the phase response. One way of expressing the transfer function is the following:

$$T(s) = \frac{V_o(s)}{V_I(s)} = \frac{K(s + z_1)(s + z_2)(s + z_3)(s + z_4) \dots}{(s + p_1)(s + p_2)(s + p_3)(s + p_4) \dots}$$

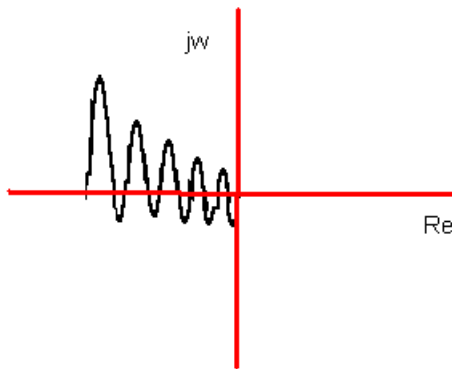
The numerator of the function contains the zeroes of the function ( $-z_1, -z_2, -z_3, -z_4, \dots$ ) and the denominator contains the poles of the function ( $-p_1, -p_2, -p_3, -p_4, \dots$ ).

The zeroes of the function are those frequencies at which the transmission will be zero. The poles of the function are the natural modes of the network, and define natural frequencies. Both poles and zeros can contain either real, complex, or purely imaginary numbers.

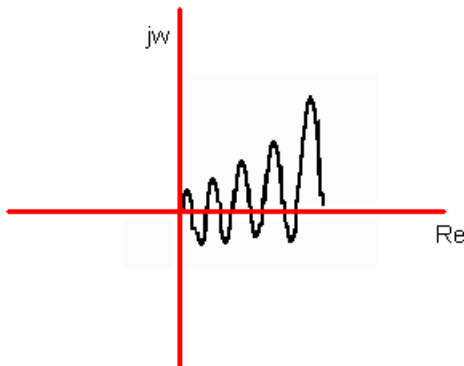
### 8.16.1.1 About Circuit Stability

As stated earlier, the stability of the circuit can be determined by examining the transfer function of the circuit. Since the transfer function is a representation of the circuit in the frequency domain, the location of the poles and zeros will be referred to the Complex plane. The complex plane is the set of axes in which the horizontal is defined as the Real Axis(Re) and the vertical is the Imaginary Axis(jw).

When all the poles of the circuit have negative real parts, the poles are located on the left hand side of the complex plane. In this situation the circuit is stable, that is, it does not generate signals on its own. The following diagram illustrates the behavior of a stable circuit:



If there are poles present on the right hand side of the complex plane, then the circuit will generate a signal of its own and, therefore, be considered unstable. The following diagram illustrates the behavior of an unstable circuit:



As stated earlier, for absolute stability there can be no poles with positive real parts, since these may cause the output signal to become unbounded. Using the poles and zeros of the transfer functions of the circuit, you can get a graphical representation of the behavior of the circuit in the frequency domain. You can obtain the approximate plots of magnitude and phase of the transfer function using Bode plots.

### 8.16.1.2 About the Bode Phase Plot

To obtain the Bode plot for the magnitude of transfer function, the asymptotic plot for each pole and zero is first drawn. The slope of the high-frequency asymptote of the curve corresponding to a zero is +20dB/decade, and that for a pole is -20dB/decade. Then the plots are added together, and the overall curve is shifted vertically by an amount determined by the multiplicative constant of the transfer function (in this case K).

The Bode phase plot is done using the same concept; however, there are some differences. The asymptotic plots consists of three lines. The first line is a horizontal one at a level of zero up to  $s=0.1\text{p11}$ , the second line has a slope of  $-45^\circ/\text{decade}$  and extends from  $s=0.1\text{p11}$  to  $s=10\text{p11}$ , and the third line has a slope of zero at a level of  $-90^\circ$ . The complete phase response can be found by adding the plots of the poles and zeros.

The frequency response of the low pass filter is an example of the above discussion.

The circuit is defined by the following transfer function:

$$T(s) = \frac{a_0}{s + \omega_0} = \frac{-40}{s + 1592.4}$$

where the natural frequency (expressed in radians) is

$$\omega_0 = \frac{1}{2\pi(5\mu F)(20\Omega)} = 1.59\text{KHz}$$

and the dc gain

$$a_0 = \frac{-20\Omega}{0.5\Omega} = -40$$

As can be seen from the cursors, the break frequency in Hertz is 1.59KHz. This is the point at which the slope of the magnitude plot is -20dB/decade due to the pole present at this frequency. The phase plot displays a slope of  $-45^\circ/\text{decade}$  between 159 Hz and 159KHz.

In the case of higher order circuits, for example, the transfer functions contain multiple poles or zeros at certain frequencies. The order of the pole or zero, n, is determined by the number of times this pole is present at a certain frequency. In this case, when plotting the Bode magni-

tude plot each higher order pole has an asymptote of -20 ndB/decade and each higher order zero has an asymptote of +20 ndB/decade.

**Assumptions** Applied to an analog circuit, small-signal. Digital pins are treated as large resistances to ground.

## 8.16.2 Setting Pole Zero Analysis Parameters

Before you perform the analysis, review your circuit and decide on input and output nodes (positive and negative). The input nodes are the positive and negative points in the circuit which are the transfer function inputs. Likewise, the output nodes are the positive and negative points in the circuit which are the transfer function outputs. You can use 0 (ground) for both positive nodes or both negative nodes.

Pole Zero Analysis Parameters are set in this screen:

Choose an analysis type: Gain Analysis (output voltage/input voltage), Impedance Analysis (output voltage/input current), Input Impedance (voltage/current as seen from the input terminals), Output Impedance (voltage/current as seen from the output terminals).

Choose input nodes on opposite sides of the input.

Choose output nodes on opposite sides of the output.

Choose analyses to be performed: Pole Analysis (finds poles of transfer function), Zero Analysis (finds zeros of transfer function), Pole and Zero Analysis (finds both).

Pole zero analysis produces the real and imaginary coordinates of the poles and/or zeros, depending on which analyses are enabled.

The Pole Zero analysis provides precise results on circuits containing passive devices (resistors, capacitors and inductors). Circuits containing active devices (transistor or opamps) will not always display the expected results.

**Note** The SPICE algorithm used in the pole zero analysis may occasionally result in an error message such as “Pole zero iteration limit reached, giving up after 200 iterations.”



Note that the analysis may still have found all the poles and zeros even if you receive this message.

## Setting Pole Zero Analysis Parameters for Normal Use

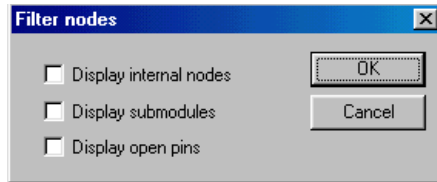
For normal use, you only need to:

- select the analysis type by enabling the desired type
- select an input node from the **Input (+)** drop-down list and the **Input (-)** drop-down list
- select an output node from the **Output (+)** drop-down list and the **Output (-)** drop-down list
- select the analysis to be performed, by choosing from the **Analysis performed** list

## Setting Pole Zero Analysis Parameters for Advanced Use

For more advanced use, you can filter the variables displayed to include internal nodes (such as nodes inside a BJT model or inside a SPICE subcircuits), open pins, as well as output variables from any submodules contained in the circuit.

- To filter the variables displayed:
1. Click **Change Filter**. The Filter Nodes screen appears.



2. Enable one or more settings.
3. Click **OK**.

## 8.17 Monte Carlo Analysis

### 8.17.1 About the Monte Carlo Analysis

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



Monte Carlo analysis is a statistical technique that lets you explore how changing component properties affects circuit performance. Multiple simulations are performed and, for each simulation, the component parameters are randomly varied according to the distribution type and parameter tolerances that you set in the screen.

The first simulation is always performed with nominal values. For the rest of the simulations, a delta value is randomly added to or subtracted from the nominal value. This delta value can be any number within the standard deviation. The probability of adding a particular delta value depends on the probability distribution. Two probability distributions are available:

**Uniform** is a linear distribution that generates delta values uniformly within the tolerance range. Any value in the tolerance range is equally likely to be chosen.

**Gaussian** distribution is generated with the following probability function:

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{u-x}{\sigma}\right)^2}$$

where

u = the nominal parameter value

s = the standard deviation value

x = the independent variable

The standard deviation,  $s$ , is calculated from the parameter tolerance according to:

$$\sigma = \frac{\text{tolerance percentage} \times \text{nominal value}}{100}$$

The percentage of the population included in the tolerance band is determined by a component's nominal parameter value plus or minus  $\sigma$  times the number of standard deviations, SD, in the tolerance band. SD is related to the percentage of population included as shown:

SD	Percentage of Population Included
1.0	68.0
1.96	95.0
2.0	95.5
2.58	99.0
3.0	99.7
3.29	99.9

For example, if you set the tolerance percentage to 5%, then, for a 1 k $\Omega$  resistor in your circuit,  $\sigma$  is 50  $\Omega$ . One standard deviation leads to a tolerance band of 0.95 k $\Omega$  to 1.05 k $\Omega$  (1 k $\Omega$  +/- 50  $\Omega$ ), and 68.0% of the population is included. At 1.96 standard deviations, the tolerance band is 0.902 k $\Omega$  to 1.098 k $\Omega$  (1 k $\Omega$  +/- 98  $\Omega$ ), and 95.0% of the population is included.

Note that the tolerance percentage is applied globally to all components.

**Assumptions** See the assumptions for the selected analysis: DC operating point analysis described in “8.3 DC Operating Point Analysis” on page 8-12, transient analysis described in “8.5 Transient Analysis” on page 8-16, or AC frequency analysis described in “8.3 DC Operating Point Analysis” on page 8-12.

## Setting Monte Carlo Analysis Tolerance Parameters

Before you perform the analysis, review your circuit and decide on an output node.

In the Model tolerance list tab, choose which tolerance parameters are to be used. You can do this using any of the following methods:

- To edit a tolerance in the list, select it and click **Edit select tolerance**. The tolerance's current variable settings appear. Modify the variables as desired and click **OK** to save.
- To delete a tolerance from the list, select it and click **Delete tolerance entry**.
- To manually add a tolerance, click **Add a new tolerance**.

The Tolerance screen appears:

Choose type of sweep to be performed: Model Parameter or Device Parameter.

When selected, the current value and a description of the parameter appear.

Choose the type of distribution: Gaussian or Uniform.

Choose Unique (each random number generation is distinct), or a numbered lot (the same random number generation for various parameters).

Choose Absolute to enter a value, or Percent to vary the parameter by the specified percentage of its stated value.

Enter a percentage value, or a set value depending on the tolerance type selected.

The screenshot shows the 'Tolerance' dialog box with the following fields and annotations:

- Parameter Type:** A dropdown menu set to 'Model Parameter'. An arrow points to it from the text 'Choose type of sweep to be performed: Model Parameter or Device Parameter.'
- Parameter Section:**
  - Device Type:** A dropdown menu set to 'BJT'. An arrow points to it from the text 'When selected, the current value and a description of the parameter appear.'
  - Name:** A text field containing '2n2222a\_bjt\_npn\_471'.
  - Parameter:** A dropdown menu set to 'npn'.
  - Description:** A text field containing 'NPN type device'.
  - Present Value:** An empty text field.
- Tolerance Section:**
  - Distribution:** A dropdown menu set to 'Gaussian'. An arrow points to it from the text 'Choose the type of distribution: Gaussian or Uniform.'
  - Lot number:** A dropdown menu set to 'Unique'. An arrow points to it from the text 'Choose Unique (each random number generation is distinct), or a numbered lot (the same random number generation for various parameters).'
  - Tolerance Type:** A dropdown menu set to 'Percent'. An arrow points to it from the text 'Choose Absolute to enter a value, or Percent to vary the parameter by the specified percentage of its stated value.'
  - Tolerance value:** A text field containing '0'. An arrow points to it from the text 'Enter a percentage value, or a set value depending on the tolerance type selected.'
- Buttons:** 'Accept', 'Cancel', and 'Help' buttons are at the bottom.

Enter the desired variables in the appropriate fields.

## 8.17.2 Setting Monte Carlo Analysis Parameters

Monte Carlo Analysis parameters are set in the following screen:

Analysis to be swept for: DC Operating Point, Transient Analysis, AC Analysis. To edit the parameters of AC or Transient Analysis, click Edit Analysis.

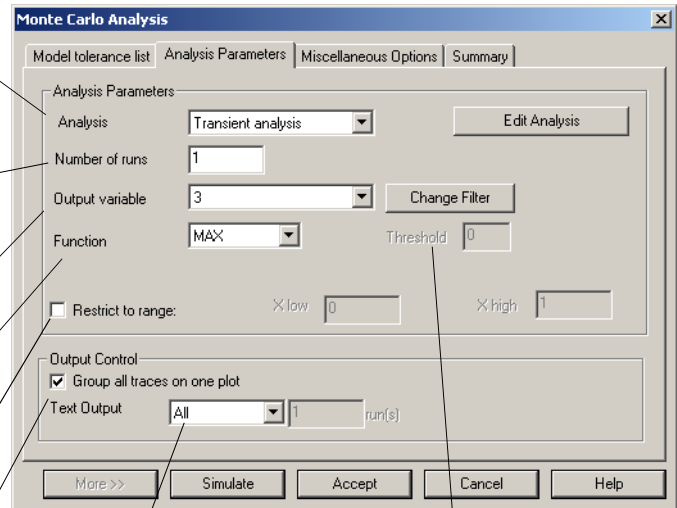
Number of runs must be greater than or equal to 2.

Choose output variable. To change the filter that affects the list of possible output variables, click Change Filter and choose the desired filter node.

Choose a collating function: MAX, MIN, RISE\_EDGE, FALL\_EDGE.

Option to restrict the x-axis range of the analysis being run. Set X low (default is 0) and X high (default is 1).

Enable to have all traces grouped on a single plot.



Choose from: All, Every, List, First, and None. If you choose Every, List or First, set the number of runs for which this applies.

For RISE\_EDGE and FALL\_EDGE functions only, enter a value for threshold voltage.

Monte Carlo analysis produces the appropriate voltage curves sequentially. The number of curves is equal to the number of runs you specified in the screen.

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.

## 8.18 RF Analyses

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



RF analyses (Characterizer, Noise Figure and Matching Networks analyses) are performed through the Network Analyzer instrument and are described in “14.3.2 Network Analyzer” on page 14-18.

## 8.19 Batched Analyses

**Note** If the simplified version option is selected, this analysis is hidden. For details on the simplified version option, see “13.3.1 Setting Global Restrictions” on page 13-4.



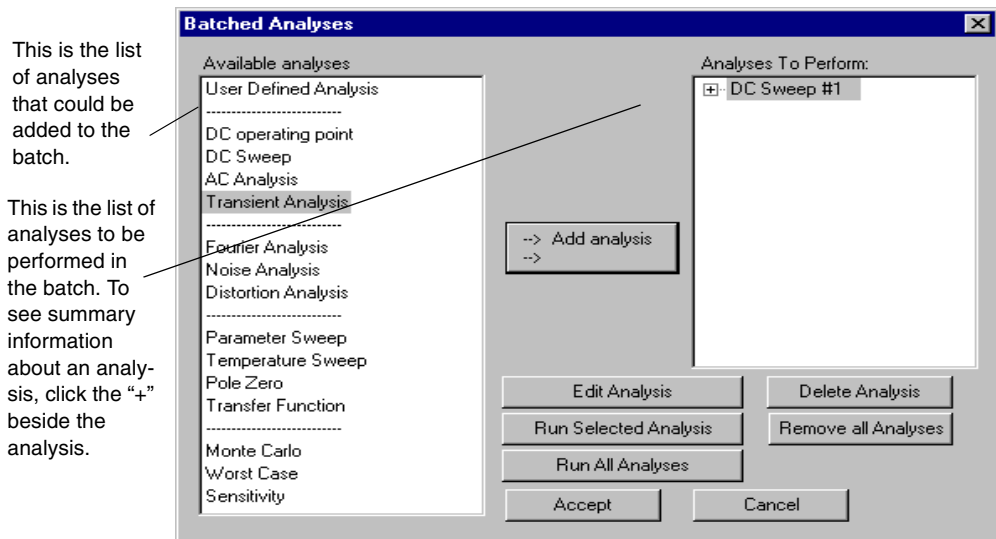
You can batch together different analyses, or different instances of the same analysis, to be performed in sequence. This provides a convenient way for advanced users to perform multiple analyses from a single, interpreted command.

For example, you might use batched analyses to:

- repeatedly perform the same set of analyses, such as when trying to fine-tune a circuit
- prepare demonstrations of circuit principles, for educational purposes
- build a record of the analyses that you performed on the circuit
- set up a sequence of long analyses to run automatically.

➤ To set up batched analyses:

1. Choose **Analysis/Batched Analyses**. The Batched Analyses screen appears:



2. To add an analysis to the batch, select it and click the **Add Analysis** button. The parameters screen for the selected analysis appears, so you can set the parameters for the analysis.
3. When you have finished the settings for the analysis, click **Add to List**.

**Note** To save the settings without adding the analysis to the batch, click **Accept**. To discard your changes click **Cancel**. Your display returns to the Batched Analysis screen.

The analysis is added to the **Analyses to Perform** list on the right. Summary information can be revealed by clicking the “+” beside the analysis.

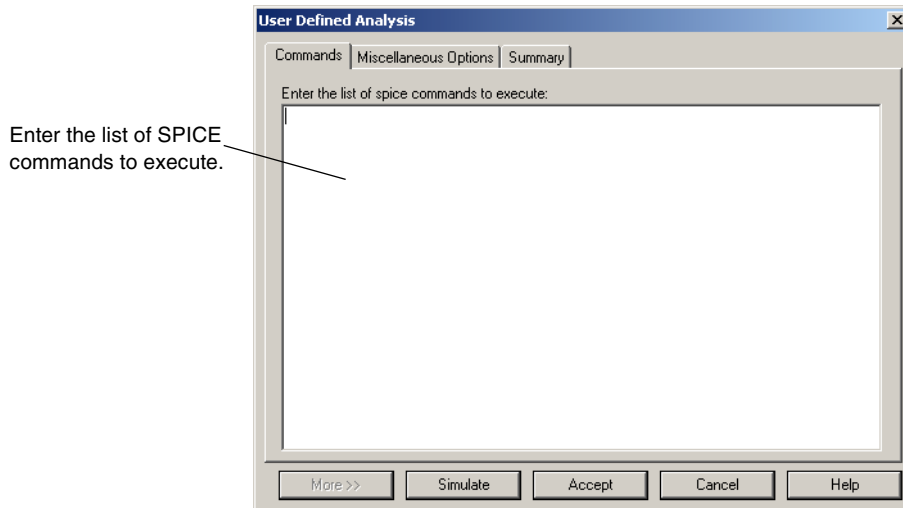
4. Continue to add analyses as desired. Note that the settings for one instance of an analysis become the default settings for that analysis during this operation. For example, if you set your first DC Sweep to an increment of 0.6, the 0.6 increment is the default value when you add your next DC Sweep to the batch.
5. To run just one of the analyses in the batch, select it and click **Run Selected Analysis**. To run all of them, click **Run All Analyses**.

The Summary tab shows the results of the analyses performed in the session.

- To edit an analysis' parameters in the batch, select it and click **Edit Analysis**. The selected analysis' parameters screen appears, allowing you to make any modifications you wish to the analysis.
- To remove an analysis from the batch, select it and click **Delete Analysis**. To remove all analyses, click **Remove All Analyses**.

## 8.20 User-Defined Analyses

The user-defined analysis presents you with the following screen into which you can type SPICE commands to be executed to perform the analysis.



You require a working knowledge of SPICE to use this interface. It provides an advanced, fully customizable way for you to set up your own analyses.

## 8.21 Noise Figure Analysis



This analysis is part of Multisim's RF Design module (standard in the Power Professional edition, optional in the Professional edition) and is described in Chapter 14, "RF".

## 8.22 Viewing the Analysis Results: Error Log/Audit Trail

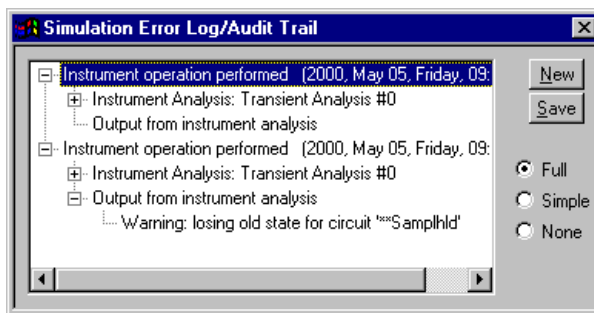
When you click **Simulate** to begin simulating the circuit using the chosen analysis or analyses, you may see one or both of the following views of the results:

- the error log/audit trail, which displays the results in text format (explained here)
- the Grapher, which displays the results in graphical format (explained in "8.22 Viewing the Analysis Results: Grapher" on page 8-64.)

If you set the analysis option ACCT on, the error log/audit trail also includes errors or warning messages generated during simulation, and a chart of simulation statistics. For more on analysis options, see "8.28 Analysis Options" on page 8-78.

- To have the error log/audit trail appear, from the **View** menu choose **Show Simulation Error Log/Audit Trail**.

The error log/audit trail display is useful for diagnosing the analysis and its results. Here is an example display:



Each analysis you perform, either individually or in batch, during this Multisim session, is stored in the audit trail. The file is cleared when you exit Multisim.



As with the summary information, you can expand or contract the contents of the display to show or hide details. You can also choose to display all the errors (**Full**), only the simple errors (**Simple**), or none of the errors (**None**) by choosing one of the options on the right of the screen. Regardless of the option you choose, the full set of errors is saved. You can also save the results of the audit trail in a separate file (click **Save** and choose a file name and location) or clear them (click **New**).

## 8.23 Viewing the Analysis Results: Grapher



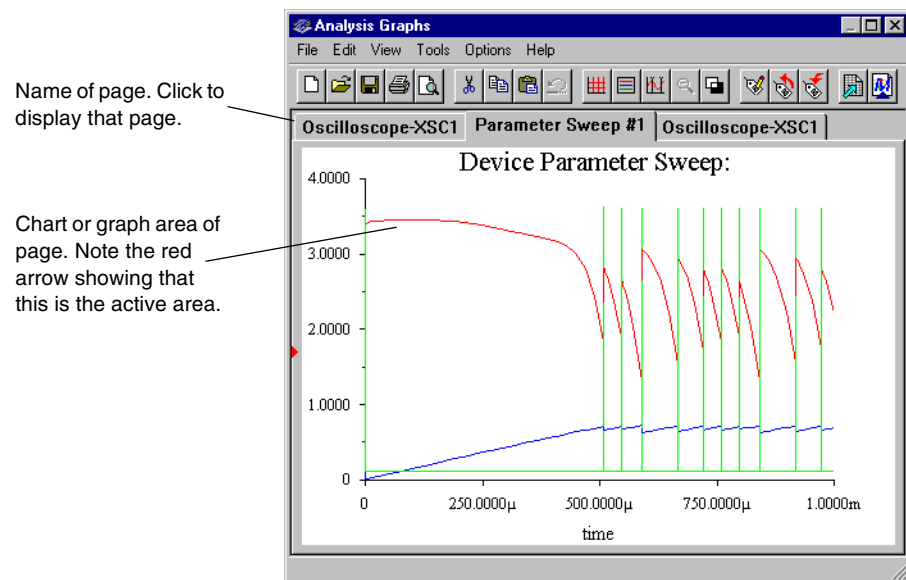
To have the Grapher appear, from the **View** menu choose **Show Grapher**.

The Grapher is a multi-purpose display tool that lets you view, adjust, save, and export graphs and charts. It is used to display:

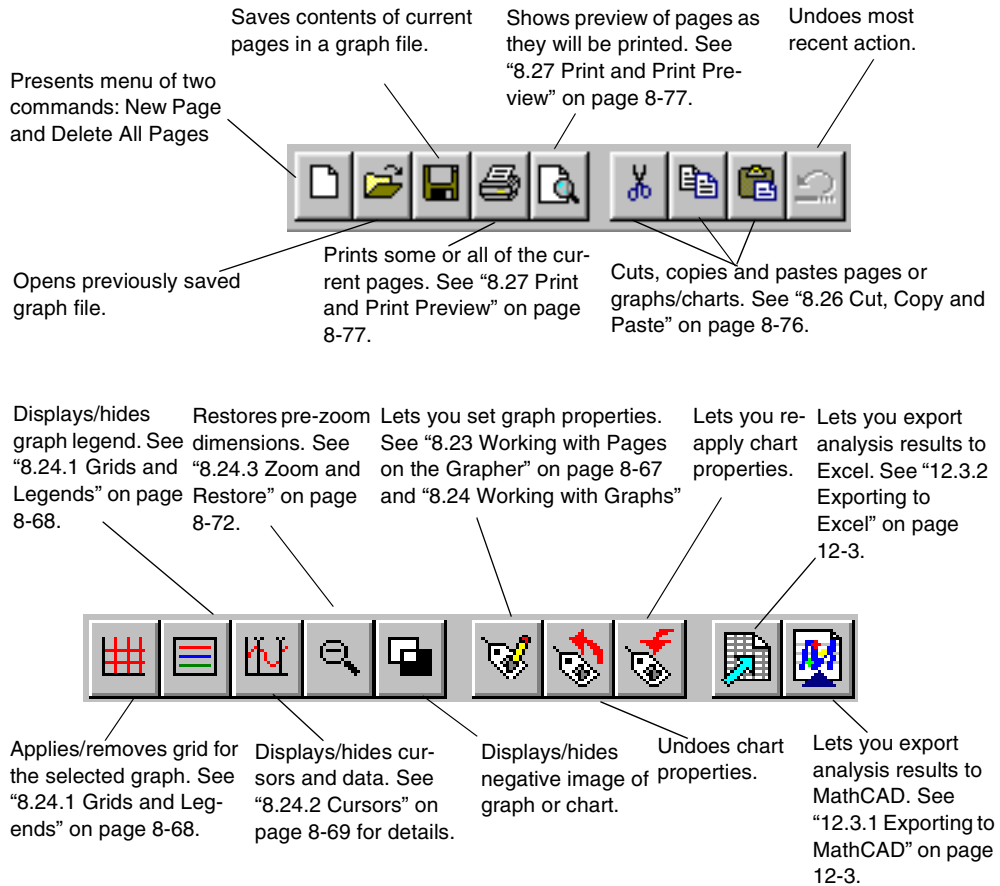
- the results of all Multisim analyses in graphs and charts
- a graph of traces for some instruments (for example, oscilloscope and Bode Plot).

The display shows both graphs and charts. In a graph, data are displayed as one or more traces along vertical and horizontal axes. In a chart, text data are displayed in rows and columns. The window is made up of several tabbed pages.

Each page has two possible active areas, indicated by the red arrow: the whole page, or the chart/graph displayed on that page. Some functions, such as cut/copy/paste, affect only the active area, so be sure you have selected the desired area before performing a function.



The window offers a number of buttons on a toolbar:



When the Grapher is on the screen, it remains visible until you do one of the following:

- Close the window.
- or
- Toggle off **View/Show Grapher**.

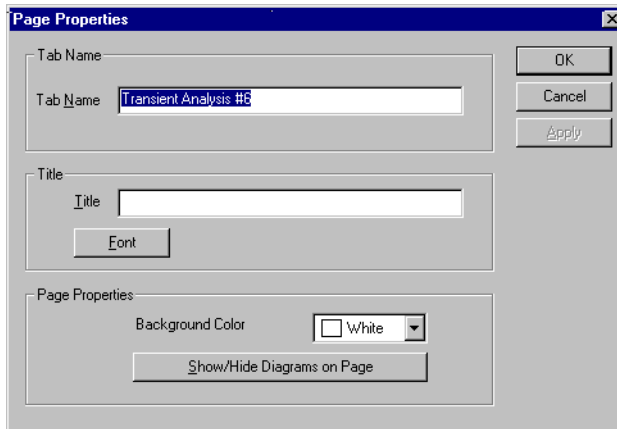
## 8.24 Working with Pages on the Grapher

Every analysis you perform on a circuit displays its results on a separate page of the grapher. Every trace may also appear on a separate page, if that is how you have set up your analysis.

- To view a page on the grapher, click its tab.
- To scroll through pages (when there are too many tabs to fit in the available space), click the forward or reverse arrow buttons that appear at the right edge of the tabs.
- To change page properties:



1. Select a page by clicking its tab.
2. Click the **Properties** button. The Page Properties screen appears.



To change:	Do this:
Name of the tab	modify <b>Tab Name</b> field
Title of chart or graph	modify <b>Title</b> field
Title's font	click <b>Font</b> button and choose from fonts displayed
Background color of page	select from <b>Background Color</b> drop-down list box
Which diagrams appear on the page	click <b>Show/Hide Diagrams on Page</b> and select from the list that appears.

3. To apply the change and close the screen, click **OK**. To apply the change and leave the screen open for additional selections, click **Apply**.

## 8.25 Working with Graphs

To help you examine graphical data, you can use a grid, a legend and vertical cursors. You can also zoom in on any part of a graph. You can apply these tools separately or together. In addition, you can change several graph display characteristics from the tabs of the Graph Properties screen.

**Note** To display the Graph Properties screen or to use the buttons described in this section, you must have a graph selected. If the Page Properties screen appears, you have a *page* selected rather than a *graph*. Click on a graph to select it. A red arrow appears to the left of the graph to indicate it is selected.

**Note** The Graph Properties screen allows you to click either **OK** or **Apply**. Clicking **OK** applies the change and closes the screen. Clicking **Apply** applies the change and leaves the screen open for additional selections.

### 8.25.1 Grids and Legends

➤ To apply a grid to a graph:

1. Select a graph by clicking anywhere on it.



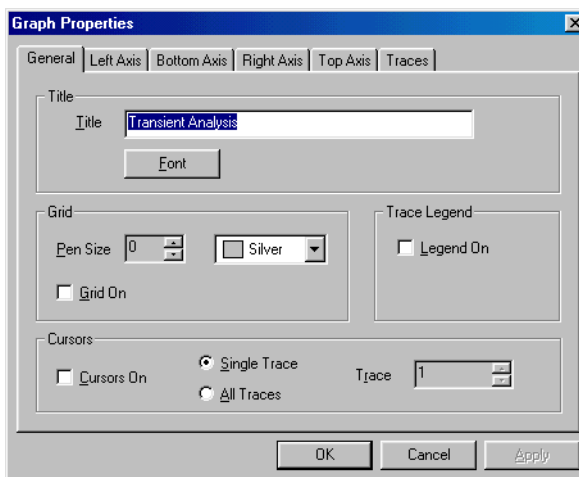
2. Click the **Show/Hide Grid** button. To remove the grid, click the button again.

or

Select a graph by clicking anywhere on it.



1. Click the **Properties** button. The Graph Properties screen appears:

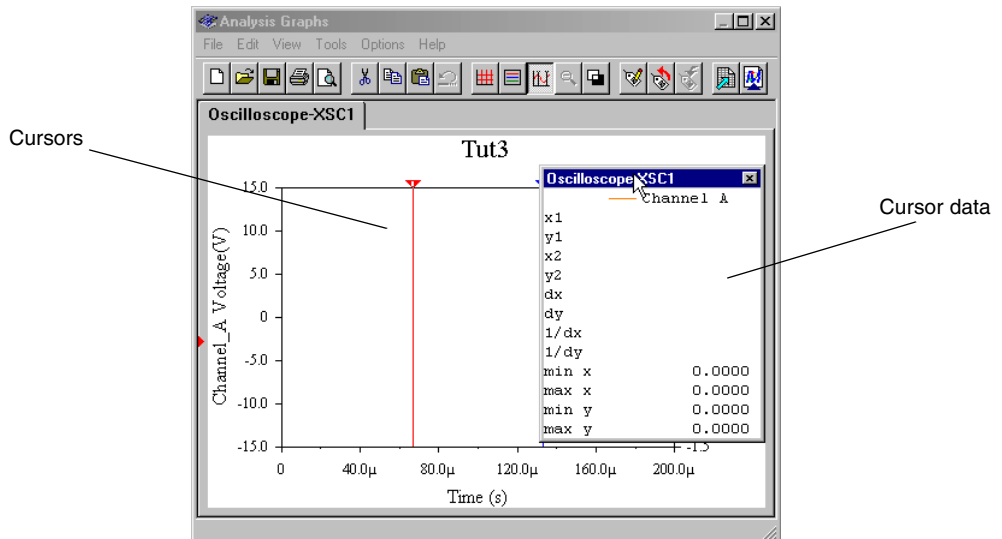


2. Click the General tab.
3. Enable the **Grid On** option. If desired, change the grid pen size and color.
- To apply a legend to a graph:
  1. Select a graph by clicking anywhere on it.
  2. Click the **Show/Hide Legend** button. To remove the legend, click the button again.  
or  
Select a graph by clicking anywhere on it.
  3. Click the **Properties** button. The Graph Properties screen appears.
  4. Select the General tab.
  5. Enable the **Legend On** option. If desired, change the labels of the traces using the Traces tab. See “8.24.6 Traces” on page 8-75 for details.



## 8.25.2 Cursors

When you activate the cursors, two vertical cursors appear on the selected graph. At the same time, a window pops up, displaying a list of data for one or all traces.



The cursor data includes:

x1,y1	(x,y) co-ordinates for the left cursor
x2,y2	(x,y) co-ordinates for the right cursor
dx	x-axis delta between the two cursors
dy	y-axis delta between the two cursors
1/dx	reciprocal of the x-axis delta
1/dy	reciprocal of the y-axis delta
min x, min y	x and y minima within the graph ranges
max x, max y	x and y maxima within the graph ranges

➤ To activate the cursors:



1. Select a graph by clicking anywhere on it.
2. Click the **Show/Hide Cursors** button. To remove the cursors, click the button again.

or

Select a graph by clicking anywhere on it.

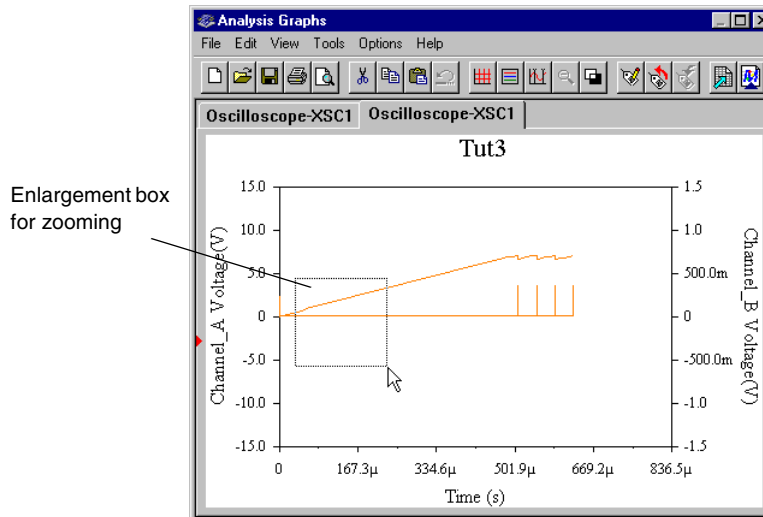


3. Click the **Properties** button. The Graph Properties screen appears.
4. Select the General tab.
5. Enable the **Cursors On** option.
6. Select **Single Trace** to view cursor data for one trace or **All Traces** to view cursor data for all traces. If you select **Single Trace** and there is more than one trace in your graph, use the **Trace** field to select the one you want.

➤ To move a cursor, click and drag it horizontally.

## 8.25.3 Zoom and Restore

- To zoom in on any part of a graph:
  1. Select a graph by clicking anywhere on it.
  2. Click and drag the pointer until the dotted enlargement box covers the region of the graph that you want to zoom in on.



3. Release the mouse button. The axes are scaled and the graph redrawn based on the enlargement box.

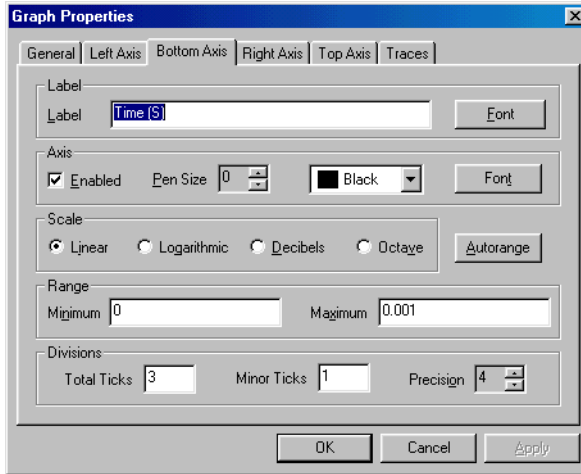
or

Select a graph by clicking anywhere on it.



4. Click the **Properties** button. The Graph Properties screen appears.

- Click an axis tab to zoom along that axis. For example, choose the Bottom Axis tab to zoom along the horizontal dimension. (Check the Traces tab to see which axis is used for the range you want to zoom.)



- Type a new minimum and maximum.



- To restore a graph to its original scale, click the **Restore Graph** button.

## 8.25.4 Title

- To apply a title to a graph:

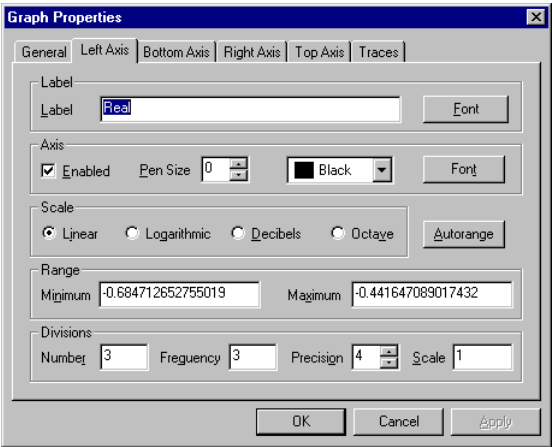


- Select a graph by clicking anywhere on it.
- Click the **Properties** button. The Graph Properties screen appears.
- Choose the General tab.
- Type a new title. To change the title's font, click the **Font** button.



## 8.25.5 Axes

You can change several characteristics of a graph's axes from the four axes tabs in the Graph Properties screen. The options are identical in each of the tabs.



- To change the characteristics of an axis:
1. Select a graph by clicking anywhere on it.
  2. Click the **Properties** button. The Graph Properties screen appears.
  3. Click the axis tab for the axis you want to change.
  4. Change any of the axis' characteristics, using the following fields:

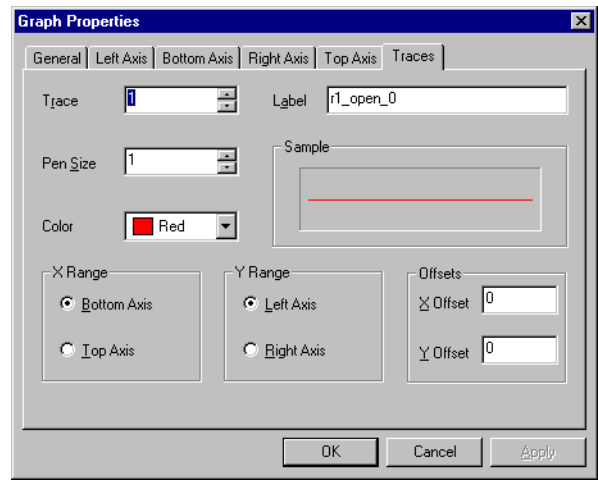


Field	Use
Label	Label for the axis. To change the axis font, click <b>Font</b> .
Pen Size	Controls the thickness and color of the axis and the font of the axis' values. To change color or font, click <b>Color</b> or <b>Font</b> .
Minimum/ Maximum	Controls the minimum and maximum values displayed. Values change when you zoom.
Number	Sets the number of tick marks on axis.
Frequency	Sets the occurrence of values on tick marks. For example, "2" means that a value appears every two tick marks.
Precision	Sets the number of significant digits for axis values.
Scale	Sets a multiplication factor for axis values. Changes the scale of the axis.

Field	Use
Enabled	Determines whether or not the axis appears.

## 8.25.6 Traces

You can change several characteristics of each trace in a graph from the Traces tab in the Graph Properties screen.



- To change the characteristics of a trace:
1. Select a graph by clicking anywhere on it.
  2. Click the **Properties** button. The Graph Properties screen appears.
  3. Click the Traces tab.
  4. Select a trace.
  5. Change any of the trace’s characteristics, using the following fields:



Field	Use
Trace	Specifies the trace whose properties are being affected.
Label	Specifies a label for the trace. Appears in legend.
Pen Size	Controls the thickness of the trace.

Field	Use
Color	Controls the color of the trace. The <b>Sample</b> box shows a preview.
Bottom Axis/Top Axis	Controls the X range of the trace.
Left Axis/Right Axis	Controls the Y range of the trace.
X Offset/Y Offset	Value to offset the trace from its original coordinates.

## 8.26 Viewing Charts

To help you examine and organize a chart, you can sort rows, adjust column widths, change precision and add a title.

- To sort a row of data, click the column name button of the column you want to sort by. Sorting order is from low to high for numbers; otherwise, it is alphabetical.
- To adjust the width of a column, click and drag the left edge of the column name button.
- To change the chart's column precision (number of significant digits) or title:



1. Select a chart by clicking anywhere on it.
2. Click the **Properties** button. The Chart Properties screen appears.
3. To change the chart title, type a new title. To change the font, click the **Font** button.
4. To change a column's precision, select a column number and a precision (number of significant digits). Precision only affects columns that contain numerical values.
5. Click **OK**.

## 8.27 Cut, Copy and Paste

The Grapher window lets you cut, copy and paste pages, graphs and charts.

**Note** You must use the cut, copy and paste buttons from this window. You cannot use the Multisim menus, buttons or keyboard shortcuts for these functions.

- To cut, copy and paste pages:



1. Select a page by clicking on its tab.
2. Click the **Cut** or **Copy** button.
3. Click the **Paste** button. The cut or copied page appears.

**Note** When a page is selected (the red arrow points to the tab), cut, copy and paste affect page properties only. They do *not* affect the graphs or charts on the page.

- To cut, copy and paste graphs and charts:
  1. Select a graph or chart.
  2. Click the **Cut** or **Copy** button.
  3. Click the **Paste** button to paste the graph or chart onto the same page.  
or, to paste onto a new page:
    4. Click the **New** button.
    5. Choose **New Page**.
    6. Type a tab name and click **OK**.
    7. Click the **Paste** button.



**Note** When a graph or chart is selected (the red arrow points to the graph or chart), cut, copy and paste affect the selected graph or chart only. They do *not* affect overall page properties.

- To open a new page, click the **New** button and choose **New Page**.
- To delete all pages, click the **New** button and choose **Delete All Pages**.
- To open an existing graph file:



1. Click the **Open** button. A file browser appears.
2. Select the file you want to open. Graph files have the file extension `.gra`.
3. Click **Open**.

- To save a graph file:



1. Click the **Save** button. A file browser appears.
2. Select a file you want to overwrite or type a new filename. Graph files have the file extension `.gra`. The file extension is automatically added.
3. Click **Save**.

## 8.28 Print and Print Preview

- To view the printed pages before you print:



1. Click the **Print Preview** button. One or two pages appear in the window.
  - Use **Next Page** and **Prev Page** to scroll through the pages.
  - Use **One Page/Two Page** to toggle between viewing one or two pages at a time.
  - Use **Zoom In**, **Zoom Out** to control the zoom on the pages.



2. Click the **Print** button to open the print screen and print the pages.

or

Click **Close** to close print preview.

➤ To print pages:

1. Click the **Print** button on the toolbar or from the print preview. The print screen appears.
2. If desired, enable **Print to file**.
3. Choose a print range.
4. Choose the number of copies.
5. Enable **Collate** if required.
6. Click **OK**.

Printed graphs indicate a key to the line colors or styles (for black and white printers) and label the names of all traces.

**Note** Colored lines are distinguished through different line styles for black and white printers.

## 8.29 Analysis Options

Multisim lets you control many aspects of the simulation used within the analyses, such as resetting error tolerances, selecting simulation techniques and viewing the results. Simulation efficiency is also dependent on the options you choose.

This section briefly describes the simulation options you have for controlling simulation used within the analyses and lists their default values. You will find these options through the Miscellaneous Options tabs of the various analyses screens, as explained in “8.2.4 The Miscellaneous Options Tab” on page 8-8.

Code	Option Name	Description	Default	Unit	Recommendation
ACCT	Print simulation statistics	Turns on/off display of statistical data on simulation-related information. Data may be useful for debugging simulation-related problems. Data appears in the Grapher screen.	Off	-	-

Code	Option Name	Description	Default	Unit	Recommendation
GMIN	Minimum conductance	Resets the minimum conductance used in any circuit branch. Cannot be zero. Increasing this may positively improve the convergence of the solution; however, it will also negatively affect simulation accuracy.	1.0e-12	mho	Do not change default.
RELTOL	Relative error tolerance	Resets the relative error tolerance of the simulation, which is the universal accuracy control. The value can significantly affect the convergence of the solution and the simulation speed. Value must be between 1 and 0.	0.001	-	Use typical values between 1.0e-06 and 0.01.
ABSTOL	Absolute error tolerance	Resets the absolute current error tolerance. Default is suitable for most bipolar transistor VLSI circuits.	1.0e-12	A	Generally, set to 6 to 8
VNTOL	Voltage error tolerance	Resets the absolute voltage error tolerance of the program.	1.0e-06	V	Generally, set to 6 to 8 orders of magnitude smaller than the largest voltage signal in the circuit.
TRTOL	Truncation error overestimation factor	Resets transient error tolerance. Only used in the local truncation error criterion.	7	-	Use default value
CHG-TOL	Charge error tolerance	Resets the charge tolerance in coulombs.	1.0e-14	C	Do not change default.
PIVTOL	Minimum acceptable pivot	Resets the absolute minimum value for a matrix entry to be accepted as a pivot.	1.0e-13	-	Do not change default.
PIVREL	Minimum acceptable ratio of pivot	Resets the relative value between the largest column entry in the matrix and an acceptable pivot value. Value must be between 1 and 0.	0.001	-	Do not change default.

Code	Option Name	Description	Default	Unit	Recommendation
TNOM	Nominal temperature	Resets the normal temperature at which model parameters are measured and calculated.	27	°C	Do not change unless you want your circuit to match data book specifications that were extracted at a temperature other than 27°C.
ITL1	DC iteration limit	Resets the upper bound limit to the number of Newton-Raphson iterations during a DC operating point analysis.	100	-	If you receive the error message “No convergence in DC analysis”, increase the ITL1 value to 500 or 1000 and rerun the analysis.
ITL2	DC transfer curve iteration limit	Resets the DC transfer curve iteration limit.	50		
ITL4	Upper transient iteration limit	Resets the upper bound limit to the number of Newton-Raphson iterations at each transient time point. Increasing the value may slow down transient simulation time. Decreasing the value increases the chance of in-convergence.	10	-	If you receive the error message “Time step too small” or “No convergence in transient analysis”, increase the ITL4 value to 15 and rerun the analysis.
DEFL	Default MOS-FET length	Resets the value for MOS channel length.	0.0001	μm	Use default value unless you know how to specify a value from a MOS device datasheet.
DEFW	Default MOS-FET width	Resets the value for MOS channel width	0.0001	μm	Use default value unless you know how to specify a value from a MOS device datasheet.
DEFAD	Default MOS-FET area of drain	Resets the value for MOS drain diffusion area.	0	m <sup>2</sup>	Use default value unless you know how to specify a value from a MOS device datasheet.

Code	Option Name	Description	Default	Unit	Recommendation
DEFAS	Default MOS-FET area of source	Resets the value for MOS source diffusion area.	0	m <sup>2</sup>	Use default value unless you know how to specify a value from a MOS device datasheet.
BYPASS	Allow bypass of unchanging elements	Turns off/on the device bypass scheme for nonlinear model evaluation. Turning off may increase simulation time.	On	-	Do not change default.
MAX-ORD	Maximum integration order	Sets the maximum order for integration when GEAR chosen as transient analysis integration method. Must be between 2 and 6. Using a higher order theoretically leads to more accurate results, but slows down simulation.	2	-	Use the default value for most circuit simulation.
TEMP	Operating temperature	Resets the temperature at which the entire circuit will be simulated. Setting in the Analysis Parameters screen will override.	27	°C	-
OLD-LIMIT	Use SPICE2 MOSfet limiting'		-	-	
ITL6	Steps in source stepping algorithm	Sets the number of steps in the Gmin stepping algorithm. Helps find a solution during a DC operating point analysis. See "7.3.8 Convergence Assistance Algorithms" on page 7-11 for more information.	10	-	-
GMIN-STEPS	Number of Gmin steps	Sets the number of steps in the Gmin stepping algorithm. Helps find a solution during the DC operating point analysis. See "7.3.8 Convergence Assistance Algorithms" on page 7-11 for more information. If a zero value is specified, the Gmin stepping algorithm is disabled.	10	-	-



Code	Option Name	Description	Default	Unit	Recommendation
MIN-BREAK	Minimum time between breakpoints		0		
NOOP-ITER	Go directly to Gmin stepping		-	-	
METHOD	Integration method	Selects for transient analysis. Default provides faster simulations with same numerical accuracy, but can produce unintended results.	TRAPEZOIDAL	-	Use GEAR (gear integration method) if unwanted numerical oscillations occur during simulation or if circuit contains ideal switches. Use default if circuit operates in oscillation mode, for example, oscillator circuits. Be aware that Gear integration may overdamp results.
TRYTO-COMPACT	Try compaction for LTRA lines	Applicable only to lossy transmission line component. When option turned on, Multisim tries to reduce data storage and memory usage needed for transient simulation of circuits containing lossy transmission lines.	Off	-	-
BADMO S3	Use old mos3 model (discontinuous with respect to kappa)		-	-	
KEEP-POP-INFO	Record operating point for each small-signal analysis	Retains the operating point information whether an AC, Distortion, or Pole-Zero analysis is run.	-	-	Particularly useful if the circuit is large and you do not want to run a redundant ".OP" analysis.

Code	Option Name	Description	Default	Unit	Recommendation
NOOPALTER	Do not do analog/event alternation in DCOP		-	-	
RAMP-TIME	Transient analysis supply ramping time	Ramps independent sources, capacitor and inductor initial conditions from zero to their final values during the time period specified.	0	s	-
MAXEV-TITER	Maximum event iterations at analysis point		0		
MAXOPALTER	Maximum analog/event alternations in DCOP		0		
CONV-LIMIT	Enable convergence assistance on code models	Enables/disables a convergence algorithm used in some built-in component models.	ON	-	-
CONV-ABSSTEP	Absolute step allowed by code model inputs between iterations	Controls automatic convergence assistance by establishing an absolute step size limit in solving for the DC operating point.	0.1	-	-
CONV-STEP	Fractional step allowed by code model inputs between iterations	Controls automatic convergence assistance by establishing a relative step size limit in solving for the DC operating point.	0.25	-	-
AUTO-PARTIAL	Use auto-partial computation for all models		-	-	

Code	Option Name	Description	Default	Unit	Recommendation
RSHUNT	Shunt resistance from analog nodes to ground	Inserts resistance to ground at all analog nodes in the circuit. Reducing value reduces simulation accuracy.	Disabled (1.0e12 when disabled)	$\Omega$	Should be set to some very high resistance, say $1\text{e}+12\Omega$ . If you get a "No DC path to ground" or a "Matrix is nearly singular" error message, try decreasing RSHUNT to $1\text{e}+9\Omega$ or $1\text{e}+6\Omega$ .
	Temporary file size for simulation.	Allows you to adjust the file size for storage of simulation results. When the file reaches its maximum size, you are prompted to stop simulation, use remaining disk space and continue, or discard existing data and continue.	10	Mb	If your circuit has many nodes and you want to scroll the oscilloscope back in time to the start of the simulation, you may need to increase the temporary file size.



# Chapter 9

## Postprocessor

### 9.1 About this Chapter

This chapter explains how to use the Postprocessor to mathematically manipulate the results of simulation obtained through analyses in different ways. Several examples are provided at the end of the chapter.

To use the Postprocessor, you must have performed at least one analysis on your circuit. This chapter assumes that you are familiar with the analyses offered by Multisim 2001, and the Grapher function that displays analysis results. For details, see Chapter 8, “Analyses”.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 9.2 Introduction to the Postprocessor

The Postprocessor allows you to manipulate the output from analyses performed on a circuit and plot the results on a graph or chart. The plotted results are referred to as “traces”. Types of mathematical operations that can be performed on analysis results include: arithmetic, trigonometric, exponential, logarithmic, complex, vector, logic, etc.

The following examples illustrate possible uses of the Postprocessor:

- Divide the output curve by the input curve obtained from a transient analysis, and observe the results.
- Multiply a voltage by a current to observe circuit power.
- Assess the differences caused by minor changes to your circuit. For example, run an analysis on a circuit, then change one condition of the circuit (such as changing the input voltage of the component’s value) and run the analysis again. Subtract one set of results from the other to show the effect of the circuit modification.

## 9.3 Using the Postprocessor

The Postprocessor calculates the results of equations and plots these results as “traces” on graphs and charts. To use the Postprocessor, you build the equations yourself by combining the variables from previous circuit analysis results with mathematical functions.

To build equations for the Postprocessor, you must have performed at least one analysis. When you perform an analysis on a circuit, the results appear in the Grapher screen and are stored for use by the Postprocessor. Some analysis results may have been saved only for the Postprocessor. For information on performing analyses, see Chapter 8, “Analyses”.

### 9.3.1 Basic Steps

- To construct an equation from which a trace will be plotted, you select variables (from previous analyses) and mathematical operators, successively moving from the left side of the equation to the right. Follow the steps below:

1. Click the Postprocessor button on the Design Bar. The Postprocessor screen appears.

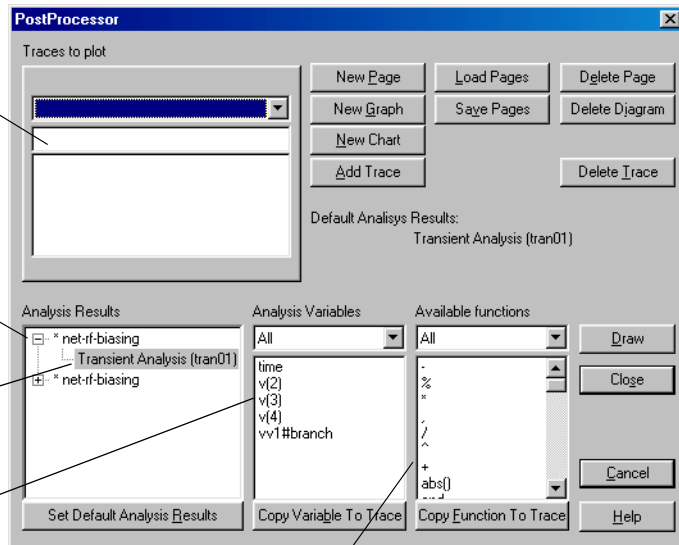


This is where you will build your Postprocessor equations.

This is the name of the circuit on which analyses have been performed in this session.

This is an analysis performed on this circuit.

These are the variables that resulted from the selected analysis.



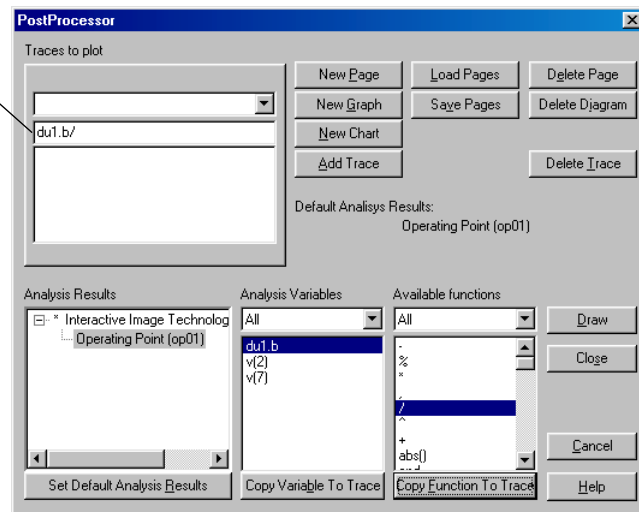
These are the mathematical functions available for use in your equations.

2. In the **Analysis Results** list, click the “+” sign beside the item naming the first analysis whose results you want to work with. Each analysis is identified with a name followed by a code in brackets. That code will be used to identify the variables from that analysis when the trace is plotted. The variables that resulted from that analysis appear in the **Analysis Variables** list. For more about the syntax of these variables, see “9.3 Postprocessor Variables” on page 9-8.

To filter the **Analysis Variables** list to show only certain variables, choose from the drop-down list of options to show:

- all variables
  - top level variables only (not those in subcircuits)
  - subcircuit variables only
  - open pins variables only
  - device parameters variables only.
3. From the **Analysis Variables** list, select the variables you want included in the equations being used to define the trace, and click **Copy Variable to Trace**. The variable appears in the “Traces to plot” window, prefixed with the code of the analysis from which it is drawn (unless the selected analysis is the default analysis — for details on using the default analysis, see “9.2.1.1 Using the Default Analysis” on page 9-5).

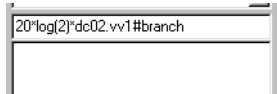
Here is the equation you are building, which will be used to plot the trace. Note how the variables are prefixed with the analysis code. In this example, the first variable comes from the analysis “disto03” and the second from analysis “dc05”.



From the **Available Functions** list, select the mathematical function you want included in the equation for this trace, and click **Copy Function to Trace**. To filter the list to show only certain mathematical functions, choose from the drop-down list of options. For details about the available functions, see “9.4 Available Functions” on page 9-8.

**Note** Although it is possible to manually type or modify a trace's equation, manual intervention can introduce syntax errors. Wherever possible, use the “copy” buttons to build your traces.

4. Continue to choose analyses, variables and functions until the equation is complete. For example, your equation might look like this:



5. To plot the traces defined by this equation, click **Draw**.
6. You are prompted for a page name. This is the name of the tabbed page that will display the results in the Grapher. It is also the name given to the Postprocessor page on which the trace will be stored. Depending on the trace, you are prompted for a Grapher graph name, plot name, or both. If prompted for both, and you do not want to create one of them, click **Cancel** when prompted for that name.

The results of the Postprocessor plotting the trace appear in the Grapher screen, on pages with the names you specified (that is, one for a plot, one for a graph), along with the results of the analyses previously performed. Results, including errors, are also recorded in the audit trail. If the audit trail is not already open, you can open it from the **View** menu.

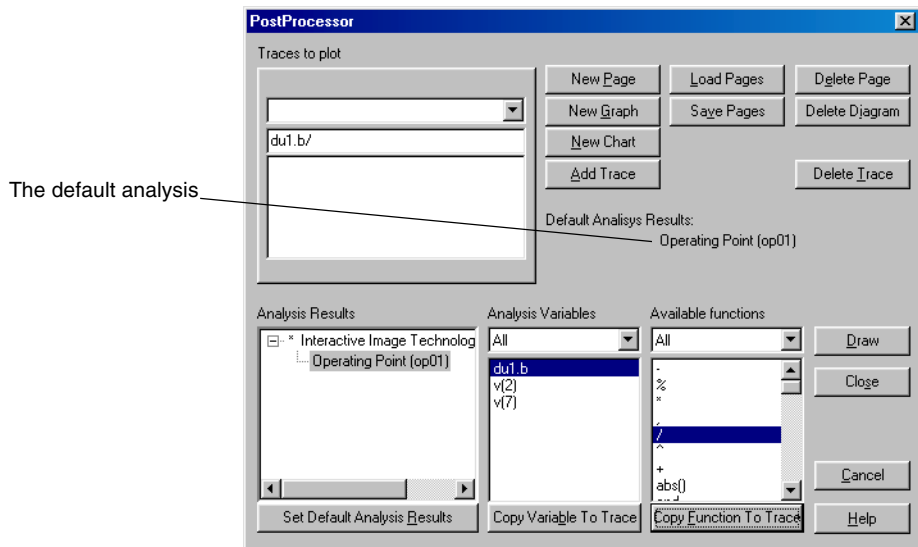
The equation as shown in the “Traces to plot” window moves down a line, leaving the top line free for a new trace. For more on working with multiple traces, see “9.2.1.2 Creating Multiple Traces” on page 9-5.



### 9.3.1.1 Using the Default Analysis

The equation you build using the Postprocessor contains variables that are prefixed with their analysis' code. To simplify the equation and the trace displayed on the graph, you can set one of the analyses to be the default analysis.

The **Analysis Results** list always contains one analysis defined as the default. The default is the analysis that, in the absence of any other indication, the Postprocessor uses for calculations. Variables from the default analysis do not have identifying prefixes in the equation or when the trace is plotted.

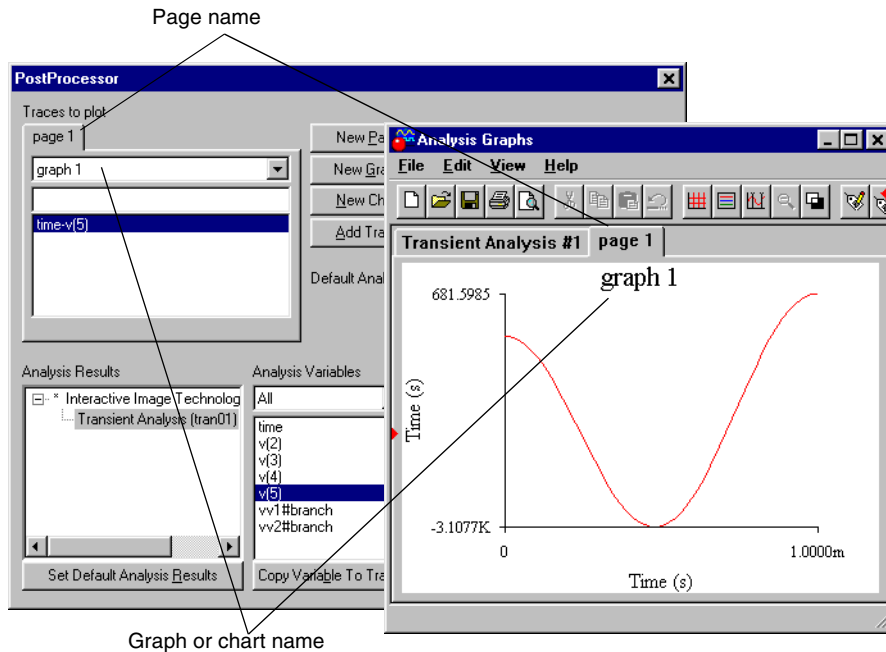


The default analysis is identified on the Postprocessor screen at the bottom of the **Analysis Results** list.

- To change the default analysis, select the desired analysis and click **Set Default Analysis Results**. The equation changes to reflect your choice.

### 9.3.1.2 Creating Multiple Traces

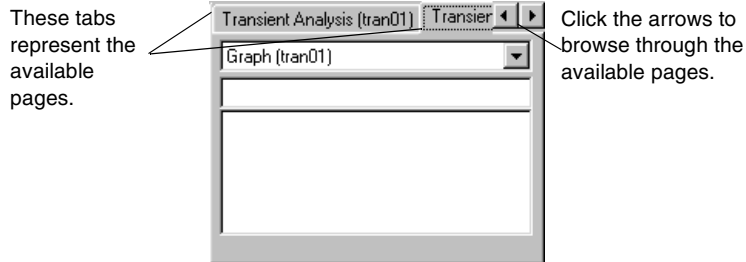
The Postprocessor screen uses the same conventions as the Grapher screen, as shown below:



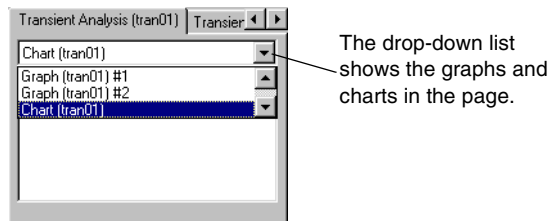
- To add another trace to a page:
  1. Click on the tab of the page to which you want to add the trace.
  2. Create the trace as usual.
  3. When you click **Draw**, the trace is added to the current page. To add it without drawing, click **Add Trace**.

## 9.3.2 Working with Pages, Graphs and Charts

- To add another page for holding traces, click **New Page**. You are prompted for a name for the page. When you click **OK**, a tab with that name is added to the Postprocessor.



- To add a graph or chart to an existing page:
  1. Click on the tab of the page to which you want to add the graph or chart.
  2. Click **New Graph** or **New Chart**. You are prompted for a name.
  3. The name is added to the drop-down list for that page.



Each chart or graph on a page appears on the same tab in the Grapher screen.

- To remove a trace, select it and click **Delete Trace**.
- To delete a page, select it and click **Delete Page**.
- To save the current set of pages, click **Save Pages**. Navigate to the location where you want to save the file and provide a file name.
- To load a saved set of pages, click **Load Pages**, navigate to the location of the saved file, select it and click **Open**.

## 9.4 Postprocessor Variables

The variables that appear in the **Analysis Variables** list of the Postprocessor are based on the selected analysis. They can include any or all of the following:

v(#)	voltage in node, where # represents the node number
vv# #branch	branch current through voltage source "vv#" (vv# represents the voltage source name)
expr.x#	expression within subcircuit x#

## 9.5 Available Functions

The functions you can apply to the Postprocessor variables are:

Symbol	Type	Description
+	Algebraic	plus
-	Algebraic	minus
*	Algebraic	times
/	Algebraic	divided by
^	Algebraic	to the power of
%	Algebraic	percentage
,	Algebraic	complex $3,4 = 3 + j(4)$
abs()	Algebraic	absolute value
sqrt()	Algebraic	square root
sin()	Trigonometric	sine
cos()	Trigonometric	cosine
tan()	Trigonometric	tangent
atan()	Trigonometric	inverse tangent
gt	Relationship	greater than
lt	Relationship	less than

Symbol	Type	Description
ge	Relationship	greater than or equal to
le	Relationship	less than or equal to
ne	Relationship	not equal to
eq	Relationship	equal to
and	Logic	and
or	Logic	or
not	Logic	not
db()	Exponential	$20 \log_{10}(\text{mag}(\text{vector}))$
log()	Exponential	logarithm (base 10)
ln()	Exponential	natural logarithm (base 3)
exp()	Exponential	e to the vector power
j()	Complex	$i (\text{sqrt}(-1))$ times vector
real()	Complex	real component of vector
image()	Complex	imaginary part of vector
vi()	Complex	$\text{vi}(x) = \text{image}(\text{v}(x))$
vr()	Complex	$\text{vr}(x) = \text{real}(\text{v}(x))$
mag()	Vector	magnitude
ph()	Vector	phase
norm()	Vector	vector normalized to 1 (that is, the largest magnitude of any component is 1)
rnd()	Vector	random
mean()	Vector	results in a scalar (a length 1 vector) that is the mean of the elements of the vector
Vector( <i>number</i> )	Vector	results in a vector of length <i>number</i> , with elements 0, 1, ... <i>number</i> -1. If <i>number</i> is a vector than just the first element is taken, and if it isn't an image then the floor of the magnitude is used.
length()	Vector	length of vector

Symbol	Type	Description
deriv()	Vector	derivative of vector — uses numeric differentiation by interpolating a polynomial and may not produce satisfactory results, particularly with iterated differentiation. Only calculates the derivative with respect to the real component of the vector's scale.
max()	Vector	maximum value from vector
min()	Vector	minimum value from vector
vm()	Vector	$vm(x) = \text{mag}(v(x))$
vp()	Vector	$vp(x) = \text{ph}(v(x))$
yes	Constat	yes
true	Constat	true
no	Constat	no
false	Constat	false
pi	Constat	pi
e	Constat	natural logarithm base
c	Constat	speed of light in vacuum
i	Constat	square root of -1
kelvin	Constat	degrees kelvin
echarge	Constat	fundamental charge
boltz	Constat	Boltzman's constant
planck	Constat	Planck's constant

# Chapter 10

## HDLs and Programmable Logic

### 10.1 About this Chapter

This chapter deals with Hardware Description Languages (HDLs) generally, and more specifically with the two most common HDLs: VHDL and Verilog HDL, and their usage in Multisim 2001. It also addresses one of the most common applications of using HDLs — designing with programmable logic devices and the process of synthesis.

This chapter is divided into three main parts: the first section provides a brief overview of HDLs within Multisim 2001, the second section deals with VHDL, and the third section deals with Verilog HDL. Important information, particularly for newcomers to HDLs, also exists in the VHDL appendix.

This chapter is primarily of use to those with the VHDL or Verilog HDL Design, Simulate and Debug module of Multisim 2001, included in the Power Professional edition and available as an add-on product to Professional Edition users. Also available is a separate add-on product that includes the ability to simulate a circuit containing a device for which the model already exists in VHDL or Verilog HDL, but not the ability to write or design VHDL/Verilog HDL source code.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description. See “1.5 Multisim 2001 Features Summary” on page 1-3 for a description of the features available in your edition.

## 10.2 Overview of HDLs within Multisim

### 10.2.1 About HDLs

HDLs are designed specially to describe the behavior of complex digital devices. For this reason they are referred to as “behavioral level” languages. They can use behavioral level models (instead of transistor/gate level, like SPICE) to describe the behavior of these devices. Using HDLs avoids the unwieldy task of describing such devices at the gate level, greatly simplifying the design process.

Designers typically choose from two different HDLs: VHDL and Verilog HDL. Multisim supports both of these languages.

HDLs are commonly used for modeling complex digital ICs that could not easily be modeled in SPICE, or for designing circuits in programmable logic. Multisim supports both of these applications of HDLs.

### 10.2.2 Using Multisim with Programmable Logic

Designing circuits using programmable logic is becoming increasingly common as engineers deal with the need for shorter design cycles, smaller products, and lower cost results. Programmable logic devices (PLDs) generally fall into three broad categories (listed in order of increasing complexity):

- PLAs (the original Programmable Logic Devices, introduced just after the earlier, simpler Programmable Array Logic)
- CPLDs (Complex PLDs)
- FPGAs (Field Programmable Gate Arrays).

All such devices share a common characteristic: they have standard blocks of digital logic and a means of interconnecting these blocks on the semiconductor. This allows you to program the device to perform a particular function. In spite of this common characteristic, however, each of these three broad classes of devices uses a different architecture for its logic blocks and the interconnections between them. Describing these varying architectures that the device vendors use for implementing blocks/interconnects within the semiconductor wafers is beyond the scope of this chapter, but is covered sufficiently in many text books on the subject.



This chapter deals with CPLDs and FPGAs because simple PLDs are not often designed using VHDL or Verilog HDL. Such simple PLDs are now less common and typically programmed with the older ABEL or CUPL languages, not addressed in Multisim. The following is a list of the key steps in designing with CPLDs and FPGAs:


- creating/writing source code in VHDL or Verilog HDL
- simulating/analyzing the operation/performance of that code
- debugging the code to generate final source code
- synthesizing the source code (specific to a particular device vendor)
- fitting (for CPLDs) or placing a routing (for FPGAs)
- physically programming the device.

The last two steps in the process must be done with tools provided by the programmable logic device vendor and are therefore not a part of Multisim.

## 10.2.3 Using Multisim for Modeling Complex Digital ICs

In addition to using Multisim for Programmable Logic design, you may also use it to write VHDL or Verilog HDL code, modeling the behavior of complex digital ICs. Alternatively, you may accomplish the same thing by obtaining models for certain devices through the public domain, from device vendors, or from others in your company who have written or obtained them. If you already have a model, you do not need to program in VHDL or Verilog HDL. Multisim will simulate such components as part of a board level circuit, as long as the model exists.

## 10.2.4 How to Use HDLs in Multisim

- To use Multisim to simulate a circuit containing a device for which the model exists in VHDL or Verilog HDL (instead of SPICE), just begin simulation, as described in Chapter 7, “Simulation”. As long as the model exists in Multisim, or you have added it (as described in Chapter 5, “Component Editing”), Multisim will automatically select the appropriate type of simulation engine when you begin simulation. Multisim also deals with the communication between the various simulation engines, without manual intervention. This function is unique to Multisim.
-  To use Multisim for writing, simulating and debugging HDL source code and/or for synthesizing HDLs, click the **VHDL/Verilog HDL** button on the design bar. From the pop-up menu that appears, select from the following options:
  - VHDL simulation
  - VHDL synthesis
  - Verilog HDL simulation

## 10.2.5 Introduction to VHDL

VHDL is a programming language that has been designed and optimized for describing the behavior of digital hardware circuits and systems. As such, VHDL combines features of a simulation modeling language, a design entry language, a test language, and a netlist language.

VHDL is an extremely comprehensive and extensive language, and cannot be entirely covered in this manual. However, an introduction to programming in VHDL, including a detailed primer and a set of examples, can be found in the appendices.

As a simulation modeling language, VHDL includes many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates, and functional operation) to be precisely described. The resulting VHDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams or system-level VHDL descriptions) for the purpose of simulation.

Just as high-level programming languages allow complex design concepts to be expressed as computer programs, VHDL allows the behavior of complex electronic circuits to be captured into a design system for automatic circuit synthesis or for system simulation. This process is called “design entry”, and is the first step taken when a circuit concept is to be realized using computer-aided design tools.

Design entry using VHDL is very much like software design using a software programming language. Like Pascal, C and C++, VHDL includes features useful for structured design techniques, and offers a rich set of control and data representation features. Unlike these other programming languages, VHDL provides features allowing concurrent events to be described. This is important because the hardware being described using VHDL is inherently concurrent in its operation. Users of PLD programming languages such as PALASM, ABEL, CUPL and others will find the concurrent features of VHDL quite familiar. Those who have only programmed using software programming languages will have some new concepts to grasp.

One area where hardware design differs from software design is in the area of testing. One of the most important (and under-utilized) aspects of VHDL is its use as a way to capture the performance specification for a circuit, in the form of a test bench. Test benches are VHDL descriptions of circuit stimulus and corresponding expected outputs that verify the behavior of a circuit over time. Test benches should be an integral part of any VHDL project and should be created in parallel with other descriptions of the circuit.

VHDL is also useful as a low-level form of communication between different tools in a computer-based design environment. VHDL's structural language features allow it to be effectively used as a netlist language, replacing (or augmenting) other netlist languages such as EDIF.

### 10.2.5.1 VHDL: A Standard Language

One of the advantages of using VHDL is that it is a standard in the electronic design community. As a result, you will be able to use your design concepts because the design entry method you have chosen is supported in a newer generation of design tools.

You will also be able to take advantage of the most up-to-date design tools and have access to a knowledge base of thousands of other engineers, many of whom are solving problems similar to your own.

### 10.2.5.2 A Brief History of VHDL

VHDL (VHSIC [Very High-Speed Integrated Circuit] Hardware Description Language) was developed in the early 1980s as a spin-off of a high-speed integrated circuit research project funded by the U.S. Department of Defense. During the VHSIC program, researchers had to describe circuits of enormous scale (for their time) and manage very large circuit design problems that involved multiple teams of engineers. With only gate-level design tools available, it soon became clear that better, more structured design methods and tools would be needed.

There are a number of progressively more advanced standards that define VHDL in detail, as described in Appendix A, “VHDL Primer”. All are supported by Multisim. The major milestones in the VHDL standards evolution are summarized below:

- first introduction of publicly available edition of VHDL (1985)
- IEEE Standard 1076 — basis of almost all of today’s products, released in 1987 and updated in 1993/94
- IEEE Standard 1164 — solves the problem of non-standard types
- IEEE Standard 1076.3 — the standard for synthesis
- IEEE Standard 1076.4 — adds timing information, known as VITAL.



# Chapter 11 Reports

## 11.1 About this Chapter



Multisim 2001 allows you to generate a number of reports. This chapter explains the major types of reports: Database Family List, Component Detail Report and Instrument Maps.

Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

## 11.2 Database Family List

You can produce a Database Family List showing all the components in a family.

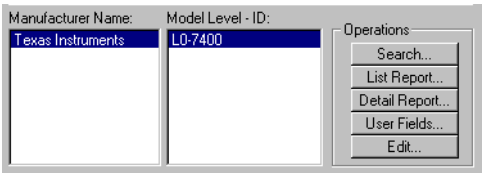
**Note** Although this report appears in the pop-up menu under the **Reports** Design Bar button, when you choose **Database Family List** you are reminded that this report is accessed through the Browser screen only. Click **OK** from this prompt to open the Browser screen. The Database Family List report is included in the **Reports** pop-up menu only as a way of listing all available Multisim reports in one location.

➤ To produce a Database Family List showing all the components in a specific family:

1. Access the database, as described in “3.3 Setting up Your Circuit Window” on page 3-1, to select a component Parts Bin and a family within that Parts Bin (for example, the 74STD of the TTL group).

or

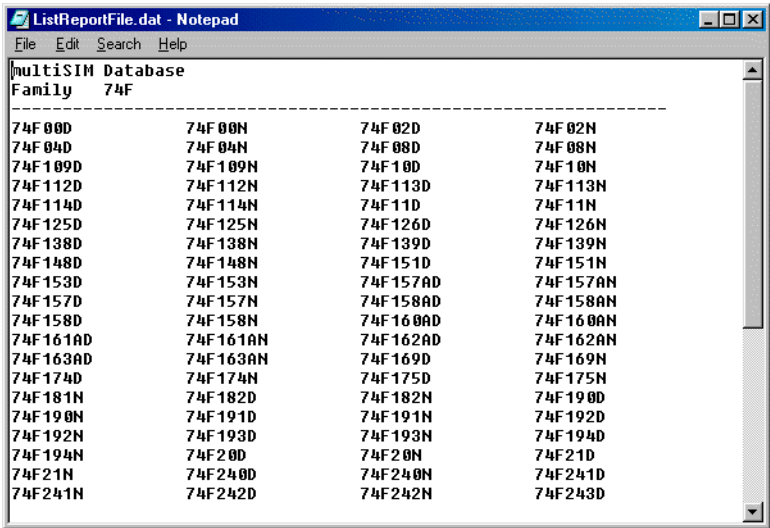
Access the Component Browser by clicking the **Reports** Design button, **Database Family List**, then **OK**.



2. From the Browser screen, click **List Report**.

**Note** Normally when you use the Browser you first select a specific component. This is not necessary when creating a Component Family list, since the list shows all the parts in this family.

3. A Notepad screen appears, listing all the components found within the currently selected family. For example:



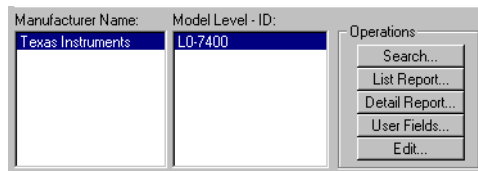
4. Use any of the standard Notepad functions to search, scroll, file, edit or print this information.
5. When done, choose **File/Exit**.

## 11.3 Component Detail Report

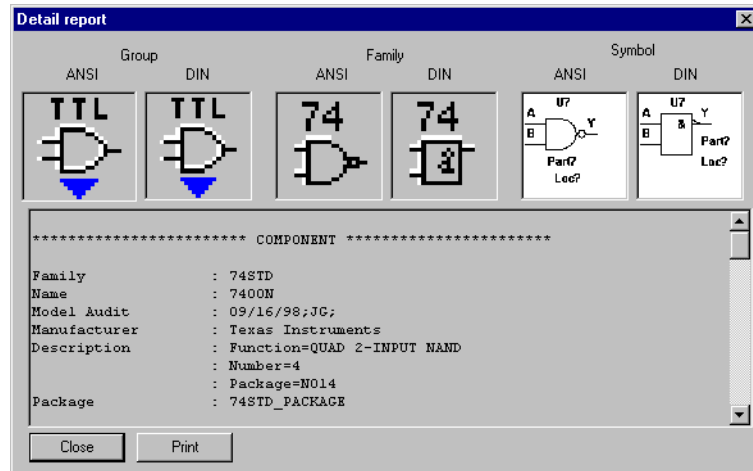
You can produce a Database Detail Report showing all the information stored in the Multisim database about a particular component.

**Note** Although this report appears in the pop-up menu under the **Reports** Design Bar button, when you choose **Database Family List** you are reminded that this report is accessed through the Browser screen only. Click **OK** from this prompt to open the Browser screen. The Database Family List report is included in the **Reports** pop-up menu only as a way of listing all available Multisim reports in one location.

- To produce a database family list showing detailed information about a specific component:
1. Access the database, as described in “3.3 Setting up Your Circuit Window” on page 3-1, to select a component Parts Bin and a family within that Parts Bin (for example, the 74STD of the TTL group).
- or
- Access the Component Browser by clicking the **Reports** Design button, **Database Family List**, then **OK**.
2. In the Browser screen, select a specific component in the family and click **Detail Report**.



3. This produces a screen that contains all the details about the selected component, including its schematic symbols, manufacturer, electrical parameters, simulation model and footprint (package). For example:



4. Scroll through the information as necessary.
5. To print the information, click **Print**. You are prompted, with a standard Windows print screen, to choose a destination printer.
6. To close the screen, click **OK**.

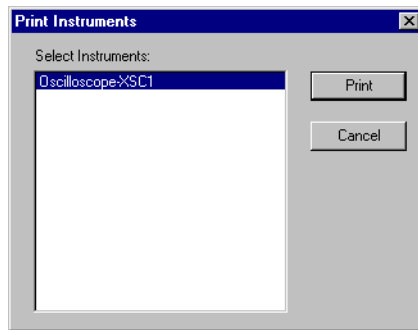
## 11.4 Instruments

The Instruments report show the faces of any or all of the instruments in your circuit. Any simulation data for the circuit appears on the instrument report.

- To produce a print-out of the instrument faces:
  1. Click the Reports button on the Design Bar and choose **Instruments** from the pop-up menu that appears.



2. You are prompted to choose which of the possible instruments in your circuit to include in printing:



3. Click on any instrument to select/deselect it, then click **Print** to print the selected instruments.
4. A standard print dialog appears. Choose the desired print options and click **OK**.



# Chapter 12

## Transfer/Communication

### 12.1 About this Chapter

This chapter explains how to use Multisim 2001 to transfer either circuit schematics themselves (in whole or in part), or the results of simulation.



Some of the features described in this chapter may not be available in your edition of Multisim 2001. Such features have an icon in the column next to their description.

### 12.2 Introduction to Transfer/Communication

Multisim makes it easy to transfer schematic and simulation data to other programs for further processing. In a unique capability, Multisim can even combine schematic information and simulation data for transfer together. For example, when transferring your schematic to perform a PCB layout, Multisim can include optimized trace width information (calculated using the Trace Width Analysis during simulation).

### 12.3 Transferring Data

#### 12.3.1 Transferring from Multisim to Ultiboard for PCB Layout

One of the most common applications to which you may want to transfer data is a PCB layout program. Ultiboard, also from Electronics Workbench, is one of the industry's leading PCB layout tools and offers many advantages over other layout programs, including trace width optimization synchronized with Multisim simulation.



- To transfer a circuit design from Multisim to Ultiboard, in order to perform a PCB layout:
1. Click the **Transfer** button on the Design Bar.
  2. From the menu that appears, choose **Transfer to Ultiboard**. A standard file browser appears.
  3. Specify the name and location of the file to be created. Multisim then creates files that can then be loaded into Ultiboard.
  4. Load the created files into Ultiboard, following the instructions in the *Ultiboard User Guide*.

**Note** If changes are made to your design while in Ultiboard, you may want to backannotate them in Multisim. This is done using the **Transfer** menu (not the Design Bar button) and is explained in “2.5.6 Transfer Menu” on page 2-34.

## 12.3.2 Transferring to Other PCB Layout

If you are using a PCB layout package produced by a vendor other than Electronics Workbench, you can create files in the necessary formats for transfer to the following third party layout packages:

- Eagle
- Lay
- OrCAD
- Protel
- Tango
- PCAD

- To transfer the circuit design to a third party layout package:



1. Click the Transfer button on the Design Bar.
2. From the menu that appears, choose **Transfer to Other PCB Layout**. A standard file browser appears.
3. Navigate to the desired folder, enter a file name and choose the desired manufacturer from the drop-down list. Multisim creates a file of the appropriate format that can then be loaded into the layout package of your choice.

## 12.4 Exporting Simulation Results

### 12.4.1 Exporting to MathCAD

You can export the results of your simulation to MathCAD, allowing you to perform sophisticated mathematical operations on your data.

**Note** This function is only available if you have MathCAD installed on your computer.

- To export the simulation results to a MathCAD session:



1. Click the **Transfer** button on the Design Bar.
2. From the list that appears, choose **Export Simulation Results to MathCAD**. A prompt screen appears, asking you to confirm that you want to open the Grapher and continue with the export process.
3. Click **OK**. The Grapher appears, showing the results of your simulation and/or analysis.
4. Use the Grapher to define which data will be transferred to MathCAD. By default, MathCAD will assign the x and y coordinates of the current trace to the variables in1 and in2. If necessary, move the trace to the correct location by clicking on it (to check which trace is current, enable the cursors).



5. Click the **Transfer to MathCAD** button.
6. A new MathCAD session is started.

**Note** MathCAD will shut down when Multisim shuts down.

### 12.4.2 Exporting to Excel

You can export your simulation results to Excel, allowing you to use the data for further processing in a spreadsheet.

**Note** This function is only available if you have Excel installed on your computer.

- To export the simulation results to an Excel spreadsheet:



1. Click the **Transfer** button on the Design Bar.
2. From the list that appears, choose **Export Simulation Results to Excel**. A prompt screen appears, asking you to confirm that you want to open the Grapher and continue with the export process.
3. Click **OK**. The Grapher appears, showing the results of your simulation and/or analysis.

4. Use the Grapher to define which data will be transferred to MathCAD. The Excel spreadsheet will contain the x and y coordinates of the current trace. If necessary, move the trace to the correct location by clicking on it (to check which trace is current, enable the cursors).



5. Click the **Transfer to Excel** button. A new Excel spreadsheet is created, with data from the x coordinates in column one and data from the y coordinates in column two.
6. If desired, save the Excel spreadsheet.

# Chapter 13

## Educators Guide

### 13.1 About this Chapter

This chapter describes the features that Multisim offers to allow you to exercise greater control over the program's interface and functionality, as well as to set certain aspects of a circuit's behaviour for teaching and training. These features include assigning faults to components in a circuit, setting global and circuit restrictions, and sharing/controlling files over a network.

### 13.2 Circuit Creator's Name

Multisim provides a feature by which the name of the creator of each circuit is stored with that circuit. This name is taken from the name entered at the time Multisim is installed. Educators can take advantage of this feature to identify the student who, for example, created the circuit being submitted as the answer to an assignment (provided that the student uses his/her own copy of the program to create the circuit). The name appears on the Circuit Restrictions screen, which you can view as long as no passwords have been set — see “13.4.2 Setting Circuit Restrictions” on page 13-9 for more information.

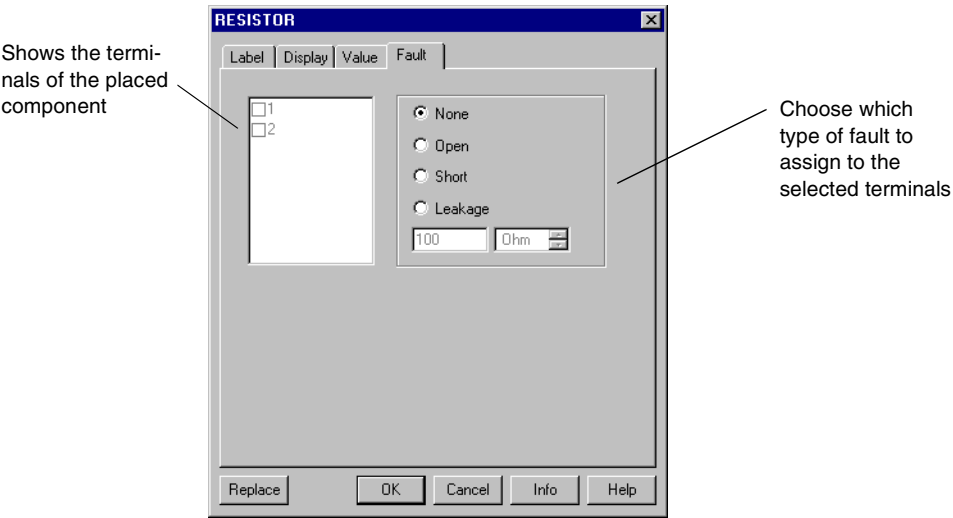
### 13.3 Assigning Faults to Components

You may want to assign faults to components for instructional purposes, such as troubleshooting exercises. You can manually assign faults to individual components in a circuit or let Multisim randomly assign faults to various components across a circuit.

### 13.3.1 Setting a Placed Component's Faults

You can assign a fault to any terminal of the placed component using the Faults tab of that component's "properties" screen.

- To assign a fault to a placed component:
  1. Double-click on the component. The component's "properties" screen appears.
  2. Click the Faults tab:



3. Select the terminals to which the fault should apply.
4. Enable the type of fault you want assigned to a terminal. The options are:

Option	Description
None	No fault
Open	Assigns a very high resistance to the terminals, as if the wire leading to the terminals was broken.
Short	Assigns a very low resistance to the terminal, so the component has no measurable affect on the circuit.
Leakage	Assigns the resistance value specified in the fields below the option, in parallel with the selected terminals. This causes the current to leak past the terminals instead of going through them.



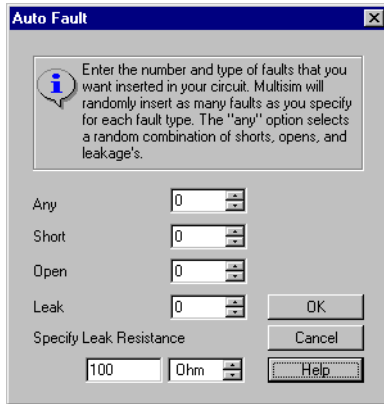
5. To cancel your changes, click **Cancel**. To save your changes, click **OK**.

## 13.3.2 Using the Auto Fault Option

When you use the Auto Fault option, you specify the number of faults or, optionally, the number of faults per different type of fault, that you want Multisim to assign to placed components in the circuit.

➤ To use the auto fault option:

1. Choose **Simulate/Auto Fault Option**. The Auto Fault screen appears:



2. Use the up/down arrow keys or enter numerical values directly in the **Short**, **Open**, and **Leak** fields, or enter a numerical value in the **Any** field to let Multisim randomly select the type of faults to assign.
3. If you specify a number of leaks, enter a number and unit of measurement in the **Specify Leak Resistance** fields.
4. Click **OK** to apply the faults, or **Cancel** to cancel, and return to the circuit window.

## 13.4 Using Restrictions

Restrictions are useful in a number of ways:

- when you are designing circuits for demonstration purposes and want to limit the functionality available to students;
- when you are sharing circuits with students and want
  - to prevent them from being able to edit the circuit in any way;
  - to limit the types of modifications they can make to a circuit;
  - to limit the types of analyses they can perform on it;
  - to limit the information they can see about certain parts of the circuit (for example, the value of a resistor you want them to calculate).

You can set global-level restrictions, which become default Multisim settings, or circuit-level restrictions, which affect only specific circuits.

To ensure that only you can set or modify restrictions, you use passwords which can protect both global and circuit restrictions. It is important that you set passwords immediately when using restrictions that you want to keep secure against any modification by students. The password for global restrictions is encrypted and stored in the Multisim program file. The password for circuit restrictions (for restricting only a particular circuit) is encrypted and stored in the circuit file.

## 13.4.1 Setting Global Restrictions

Use global restrictions to set the basic level of functionality of Multisim available to students in all circuits with which they will work. You can select a default path where circuits are to be saved, hide databases, Parts Bins and the “In Use” list, and determine whether students may edit components or place instruments.

You can also hide a number of instruments and analysis options from the menus by using the simplified version. See “13.4.1.2 Simplified Version” on page 13-7.

**Note** Global restrictions are overridden by circuit restrictions if the circuit restrictions are saved with the circuit. See “13.4.2 Setting Circuit Restrictions” on page 13-9 for information.

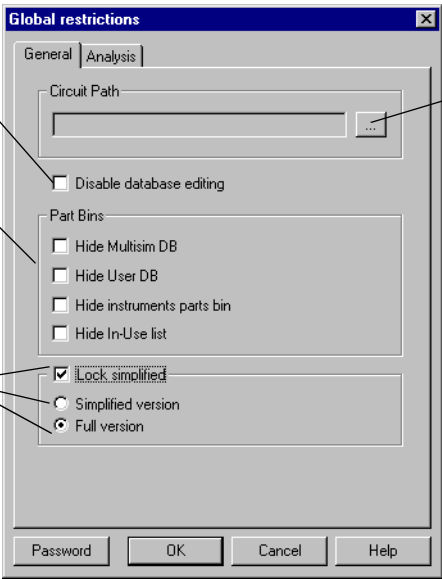
### 13.4.1.1 General Global Restrictions

- To set general global restrictions:
  1. Choose **Options/Global restrictions**. Enter the default password “testbench” and click **OK**. You can, and should, change this default password. (See “13.4.3 Setting Passwords for Restrictions” on page 13-11 for more information.) The Global restrictions screen appears:

2. If it is not displayed, click the General tab:

Enable these checkboxes to disallow component editing, to hide Parts Bins, databases, and "In-Use" list.

See "13.4.1.2 Simplified Version" on page 13-7.



Click here to select a default path/location to store circuits.

3. Set your options by enabling the appropriate checkboxes. Select from the following options:

Circuit path	Sets the default path and location where students find and save files.
Disable component editing	Ensures that students cannot edit component properties. Makes the Component editing unavailable and disables the <b>Tools/Edit Component</b> command.
Hide Multisim DB	Hides the Multisim database and Parts Bins from the interface.
Hide User DB	Hides the "user" database and Parts Bins from the interface.
Hide Instruments Parts Bin	Makes instruments unavailable to be placed in the circuit. Hides the Instruments Design Bar button from the interface, and disables the <b>Simulate/Instruments</b> command.
Hide In-Use List	Hides the In-Use List

4. Click **OK**.

Your options are immediately set for all circuits, unless you have set circuit restrictions. (See “13.4.2 Setting Circuit Restrictions” on page 13-9.)

### 13.4.1.2 Simplified Version

The simplified version presents an easier to use Multisim interface for students and instructors who do not need access to the full set of Multisim functions. The simplified version removes access to a variety of functions, including post processing, VHDL and/or Verilog, component editing and some instruments and analyses. The simplified version can also be locked, preventing students from turning it off with **Options/Simplified Version** and having access to all analyses and instruments.

- To set up the simplified version:
  1. Display the General tab of the Global restrictions screen:

Global restrictions

General Analysis

Circuit Path

☐ Disable database editing

Part Bins

☐ Hide Multisim DB

☐ Hide User DB

☐ Hide instruments parts bin

☐ Hide In-Use list

☒ Lock simplified

☐ Simplified version

☒ Full version

Password OK Cancel Help

Locking a view disables the Options/Simplified Version menu option.

Enable this box to turn on simplified version.

Enable this box to set the full version without restrictions.

2. Set your options by enabling the appropriate checkboxes. Select from the following options:

Simplified Version	Changes the interface display by hiding the design bar and restricting the available instruments and analyses. If the simplified version is restricted, it will be greyed out in the Options menu.
Full Version	Displays the Multisim interface without restrictions.

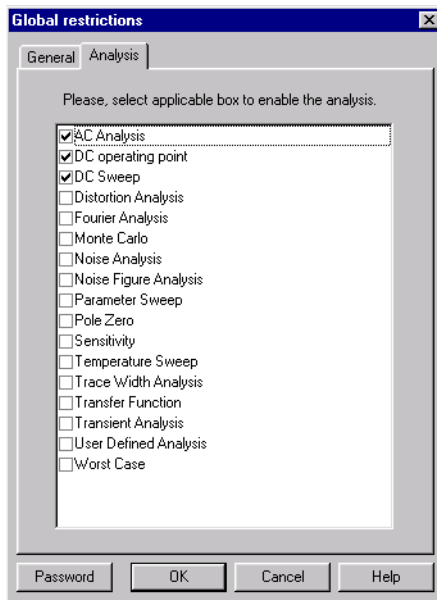
3. Click **OK**.

Your options are immediately set for all circuits, unless you have set circuit restrictions. (See “13.4.2 Setting Circuit Restrictions” on page 13-9.)

### 13.4.1.3 Global Analyses Restrictions

- To set global analyses restrictions:

1. From the Global restrictions screen, click the Analyses tab:



2. Enable the desired analyses by clicking the appropriate checkboxes (for more information on analyses, see Chapter 8, “Analyses”), and click **OK**. Only the analyses you check will be enabled on the **Simulate/Analyses** menu or when the student clicks the Analyses Design Bar button.

These options are immediately set for all circuits, unless you have set circuit restrictions. (See “13.4.2 Setting Circuit Restrictions” on page 13-9.)

## 13.4.2 Setting Circuit Restrictions

Use circuit restrictions to set restrictions on individual circuits. Circuit restrictions override global restrictions. They are saved with your circuit and invoked each time the circuit is loaded. In addition to hiding Parts Bins and databases and setting available analyses, you can set a schematic to be read-only, i.e. not editable by students, you can hide components' values, faults and uses in analyses, and you can lock subcircuits to make them unavailable for opening by students.

**Note** Remember that circuit restrictions only apply to the current circuit; i.e. when you create a new circuit, only the global restrictions will apply (see “13.4.1 Setting Global Restrictions” on page 13-5 for details). If you want circuit restrictions to apply to a new circuit, you will need to reset those restrictions each time you create a new circuit.

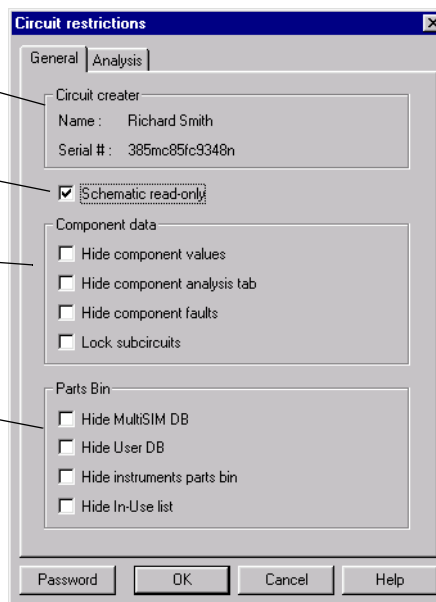
- To set general circuit restrictions:
  1. Choose **Options/Circuit restrictions**. If you have created a password, you will be prompted for it. (See “13.4.3 Setting Passwords for Restrictions” on page 13-11 for more information.) Enter your password in the Password screen, and click **OK**. The Circuit restrictions screen appears.
  2. If it is not displayed, click the General tab:

The creator of the circuit. (This information is supplied by the installation program.)

Enable to set the schematic as “read-only”.

Enable checkboxes to hide component properties and lock subcircuits.

Enable the appropriate checkboxes to hide desired Parts Bins, databases, and “In-Use” list.



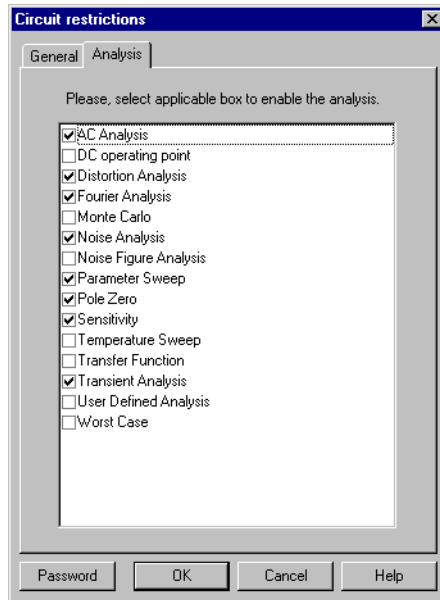
3. Set the desired options by enabling the appropriate checkboxes. Select from the following options:

Schematic read-only	Prevents students from saving the circuit, and hides Parts Bins. Students will only be able to draw wires between instruments and an open pin on an existing connector. Also, they can only remove wires that are between an instrument and a connector.
Hide component values	Marks the Values tab of components' "properties" screens with an "X" and hides values. You may wish to provide false values using labels.
Hide component analysis tab	Hides the Analysis setup tab of components' "properties" screens.
Hide component faults	Marks the Faults tab of components' "properties" screens with an "X", and hides faults.
Lock subcircuits	Prevents students from opening subcircuits and seeing their contents. Students must measure the input and output of a hidden subcircuit to determine its contents.
Hide Multisim DB	Hides the Multisim database and Parts Bins from the current circuit.
Hide User DB	Hides the "user" database and Parts Bins from the current circuit.
Hide Instruments Parts Bin	Makes instruments unavailable to be placed on the circuit. Hides the Instruments Design Bar button and disables the <b>Simulate/Instruments</b> command for the current circuit.
Hide "In-Use" list	Hides the "In-Use" list from the current circuit.

4. Click **OK**. The options you select are immediately invoked in the circuit.
5. To have the restrictions apply each time the circuit is opened, choose **File/Save** to save the restrictions in the circuit file.



- To set circuit analyses restrictions:
  1. From the Circuit restrictions screen, click the Analyses tab:



2. Enable the desired analyses by clicking the appropriate checkboxes (for more information on analyses, see Chapter 8, “Analyses”), and click **OK**. Only the analyses you check will be enabled on the **Simulate/Analyses** menu or when the student clicks the Simulate Design Bar button.
3. To have these analyses apply each time the circuit is opened, choose **File/Save** to save the restrictions.

### 13.4.3 Setting Passwords for Restrictions

When using restrictions, you should create a password immediately to ensure that your settings are secure.

- To create/change a password:
  1. For global restrictions, choose **Options/Global restrictions**. For circuit restrictions, choose **Options/Circuit restrictions**.

2. From the restrictions screen that appears, click **Password**. The Change Password screen appears:

3. If you are choosing a password for the first time, leave the **Old password** field blank. If you are changing a password, enter the old password in the **Old password** field.
4. Enter your (new) password in the **New password** field.
5. Confirm your password by entering it again in the **Confirm password** field.
6. Click **OK** to return to the Global restrictions screen, or **Cancel** to begin again.

**Note** If you want to change global/circuit restrictions, you will need to enter the password you have assigned in either case. Be sure to keep your passwords written down and in a safe place, as you will not be able to retrieve them from the program or circuit files, where they are stored in encrypted form.

**Note** A circuit password is not automatically transferred to a new circuit when you go to set circuit restrictions for it, so you will need to recreate the password every time you create circuit restrictions that you want to keep secure.

## 13.5 Remote Control/Circuit Sharing



Multisim's Remote Control/Circuit Sharing module allows you to share a circuit with students over a network, and to control Multisim running on each others' PCs.

This module offers a way for you and groups of students to work at a distance with the same circuit in a laboratory environment, and is useful for demonstrations and examples, allowing students to see you making changes to the circuit in real time. It can also be very useful for online tutors trying to assist a student having difficulty at home.

To use Multisim's Remote Control/Circuit Sharing capability, you must have access to a network or the Internet, and have the free Microsoft application Netmeeting installed. To install a copy of this product, go to <http://www.microsoft.com/netmeeting>.

Once you have established a connection with students, you can:

- send text messages back and forth (in a “chat” mode)
- see and/or talk to the students (if you have the necessary software and hardware to support audio/video links)
- use an electronic whiteboard to present ideas to students
- send circuit files to students
- control students’ machines, so you can show them the changes you are making to a circuit.



➤ To use the Remote Control/Design Sharing module:

1. Click the Transfer button on the Design Bar.
2. From the list that appears, choose **Remote Control/Design Sharing**. Netmeeting is launched.



# Chapter 14

## RF

### 14.1 About this Chapter



This chapter describes the key capabilities included in the RF Design Module. This module is part of the Multisim 2001 Power Professional product, and is available as optional for the Professional Edition.

This chapter contains descriptions of the elements (components, model makers, instruments, analyses) of Multisim 2001's RF Design module, as well as a tutorial demonstrating their use and some specific examples of RF functionality.

### 14.2 Introduction to the Multisim RF Module

The Multisim RF module is intended to provide fundamental RF circuit design features needed by engineers to design, analyze and simulate RF circuits.

The Multisim RF module is made up of the following:

- RF-specific components, including customized RF SPICE models
- model makers for creating your own RF models
- two RF-specific instruments (Spectrum Analyzer and Network Analyzer)
- several RF-specific analyses (circuit characterizer, matching network cells, noise figure).

Elements of the RF Design module are fully integrated into Multisim. That is, the instruments, analyses and components are installed in the same places and invoked in the same way as all other instruments, analyses and components. You will not see a separate access to the RF Design Module in Multisim's interface. Instead, for example, the RF components go into their own Parts Bin on the component toolbar and the RF instruments are accessed through the Design Bar's **Instrument** button.

## 14.3 Components

### 14.3.1 About RF Components

Components in the electronics field fall into two categories — “lumped” components and “distributed” components. Lumped components are smaller than the wavelength, where:

$$\lambda = \frac{c}{f}$$

In these cases, the wavelength of the voltages and currents at which they are operating is significantly larger than the components themselves. Ohm's Law, in this case, is valid. For example, a 1/4 Watt resistor is about 0.270 inch long and 0.090 inch in diameter.

RF components, on the other hand, are most often “distributed elements”, where the phase of a voltage or current changes significantly over the physical extent of the device, because the device dimensions are similar to, in some cases even larger than, the wavelength. Standard circuit theory is therefore not always applicable to circuits that are working at a few MHz to above GHz.

The usual models for lumped components are often not valid in the RF world. For example, a capacitor can behave as an inductor or an inductor can behave like a capacitor at high frequencies.

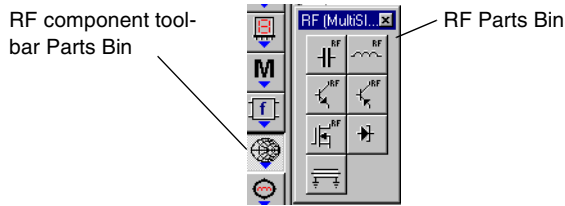
RF components exhibit parasitic effects, and have models different from those used in low frequencies. Connections between two nodes at high frequencies would exhibit different behavior from those at low frequencies, and are modeled using capacitances and inductances. These connections behave as transmission lines when implemented on a Printed Circuit Board (PCB). The board itself becomes part of the circuit, interfering with the normal circuit function. That is why low frequency circuit simulation EDA tools can become unreliable at higher frequencies.

Standard RF components include capacitors, inductors, toroids, ferrite beads, couplers, circulators, transmission lines or striplines, waveguides, and high frequency active devices such as transistors and diodes. More complex components, such as quadrature hybrids, mixers, filters, and attenuators, are built using these standard components. This chapter deals with the standard components and their models in high frequencies.

## 14.3.2 Multisim's RF Components

The RF Design module contains over 100 parts and models specifically built for accuracy at higher frequencies. This ability to handle higher frequencies helps overcome a typical problem with SPICE models, which tend to perform poorly at such frequencies.

These parts are found in the Component toolbar near the bottom in the Parts Bin that looks like this:



You access RF components as you would any other Multisim components. There are several Component Families in the RF Parts Bin, including: RF capacitors, RF inductors, RF NPN BJTs, RF PNP BJTs, RF MOSFETs, RF tunnel diodes and RF striplines/waveguides.

Families containing components with wide commercial availability (e.g. RF NPN BJT) have a large number of components within them. Families containing components that are not readily available “off-the-shelf” (e.g. RF inductors) contain only a few components. The latter were modelled using Multisim’s RF Model Makers, explained in “14.3 RF Instruments” on page 14-10. RF Model Makers are also used to customise your own parts, as explained in “14.5 RF Model Makers” on page 14-30.

## 14.3.3 Theoretical Explanation of the RF Models

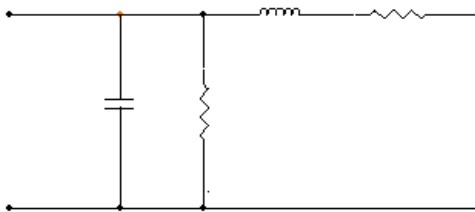
This section explains some of the scientific theory behind the operation of RF components and why they must be modelled differently at higher frequencies. You do not need to understand this information to use the RF components in Multisim, but it may be of interest to you. If it is not of interest, you may proceed to section 14.3 and begin using the RF components immediately.

### 14.3.3.1 Striplines/Microstrips/Waveguides

Wires are used to connect two internal nodes on a circuit and show no difference in phase and magnitude between two connecting nodes. Wires, however, behave differently at RF frequencies depending on their length and diameter. One effect at RF frequencies is called the “skin effect”, explained below.

A conductor, at low frequencies, utilizes its entire cross-sectional area as a transport medium for charge carriers. As the frequency is increased, an increased magnetic field at the center of the conductor presents an impedance to the charge carriers, thus decreasing the current density at the center of the conductor and increasing it around its perimeter. This effect is called the “skin effect”, and occurs in all conductors, including resistor leads, capacitor leads, and inductor leads. As the frequency increases, this effect is more pronounced.

A simple wire connecting two nodes in high frequencies behaves as a transmission line. The following figure shows the equivalent circuit of a transmission line. There are four components. The capacitor is the result of an actual capacitance existing between the center of the conductor and the ground. Between these two plates is the dielectric, which is not perfect. This leakage is modeled using conductance G and is given per unit length of line. Also, due to the resistance of the conductor itself, we have a series resistance R. Its value depends on the resistivity of the material used, the length, the cross-section of the conductor, and the skin effects.



Every transmission line has a resistance, called its “characteristic impedance”. Most microwave systems have a characteristic impedance of 50 Ohm. This value is a compromise between maximum power handling capability and minimum attenuation. At 50 Ohm, there is a reasonably low attenuation, and adequate power handling capability.

If the outer diameter of the conductor of a coaxial line is shown by “D”, and the inner diameter is shown in “d”, and  $\epsilon$  is the dielectric constant of the cable, the characteristic impedance is calculated by the following formula:

$$Z_0 = \frac{138}{\sqrt{\epsilon}} \log_{10} \left( \frac{D}{d} \right)$$



The components C and L shown in the figure above are calculated as follows:

$$C = \frac{7.354\epsilon}{\log_{10}(D/d)} \quad ((PF)/(ft))$$

$$L = 0 - 1404 \log_{10}(D/d) \quad ((\mu H)/(ft))$$

A stripline is a useful form of transmission line. The stripline consists of a conducting strip lying between, and parallel to, two wide conducting planes. The region between the strip and the planes is filled with a uniform dielectric.

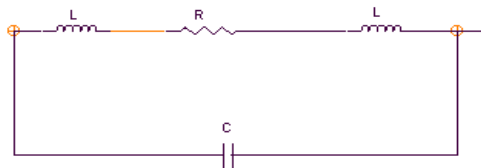
Microstrips are easy to fabricate using photolithographic processes. At the same time that a transistor is placed on top of the board, for example, a microstrip can also be placed. Microstrip is, therefore, easily integrated with other passive and active devices. A conductor of width W is printed on a thin, grounded dielectric substrate of thickness “d” and relative permittivity “ $\mu_r$ ”.

A waveguide is a structure, or part of a structure, that causes a wave to propagate in a chosen direction.

If the waveguide boundaries change direction, the wave is constrained to follow. Waveguides come in a variety of types: simple parallel plate structure, cylindrical structures with conducting boundaries, rectangular waveguides, and circular waveguides. A transmission line or a stripline is a special case of waveguide.

### 14.3.3.2 RF Resistors

Resistors find many applications as terminators or attenuators. The equivalent circuit of a resistor at radio frequency is shown in the following figure. The inductor is calculated using the physical geometry of the resistor.



$$L = 0.002 l \left[ 2.3 \log \left( \frac{4l}{d} - 0.75 \right) \right] (\mu H)$$

$l$  = length of wire in cm

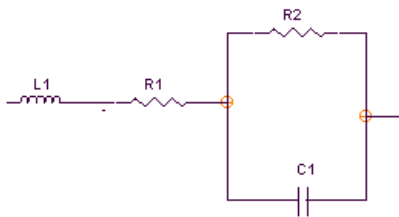
$d$  = diameter in cm

### 14.3.3.3 RF Capacitors

Capacitors are used for interstage coupling, for bypassing, in resonant circuits, and in filters. RF capacitors must be chosen carefully to ensure the best performance for specific applications. RF capacitors consist of two metal plates separated by a dielectric. The capacitance of an *ideal* capacitor has a direct relationship with the area (A), and is proportional inversely to the thickness of the dielectric (d). Its relationship is expressed in the following formula, where  $\epsilon$  is the dielectric constant of the dielectric material.

$$C = \epsilon \frac{A}{d}$$

The actual capacitor shows imperfection. One type of capacitor is modeled as shown in the following figure.



In order to find the numerical values of the ideal elements in the model above, we need to consider a number of factors.

Let  $\phi$  represent the phase of current compared to the voltage. This phase is ideally 90, but is smaller for real components. The power factor (PF) is defined as  $\cos(\phi)$ . This factor is a function of temperature, frequency, and the dielectric material. The power factor is usually used to describe the capacitor in low frequencies.

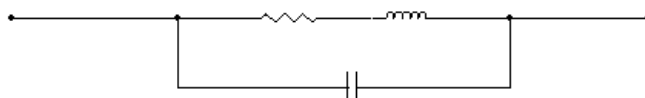
This factor in higher frequencies is sometimes referred to as the dissipation factor. This factor describes how much power is dissipated, lost, or transformed to heat energy in RF frequencies. Another factor that defines the quality of the capacitor is closely related to power factor and is called Q factor. This factor is the reciprocal of dissipation factor. The larger the Q, the better the capacitor.

No dielectric material is perfect. Therefore, there is always leakage current between two plates. This behavior is best described by  $R_p$  which is usually around 100,000 MOhm. The series resistor is the AC resistance of the capacitor in high frequencies, and is obtained using  $\cos(\phi)/C\omega * 1e6$ . Here,  $\omega = 2\pi * f$ .

There is a frequency point above which the capacitor starts to behave like an inductor.

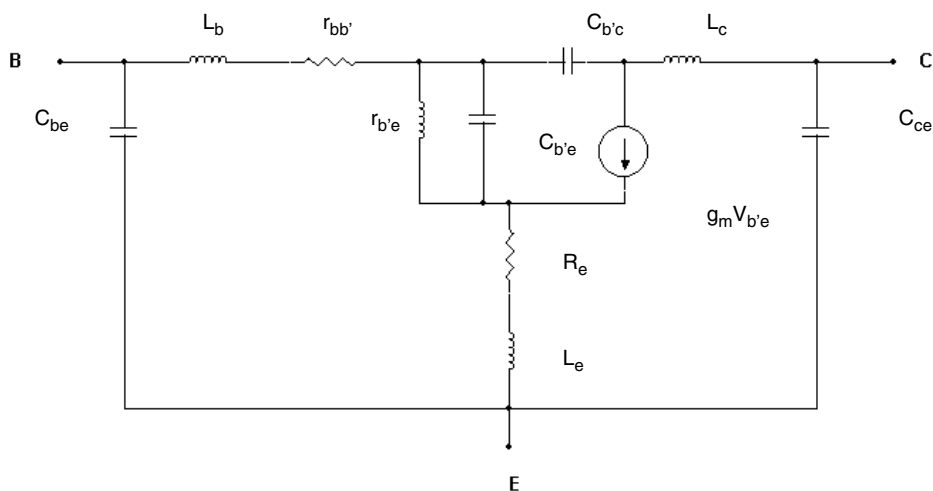
### 14.3.3.4 RF Inductors

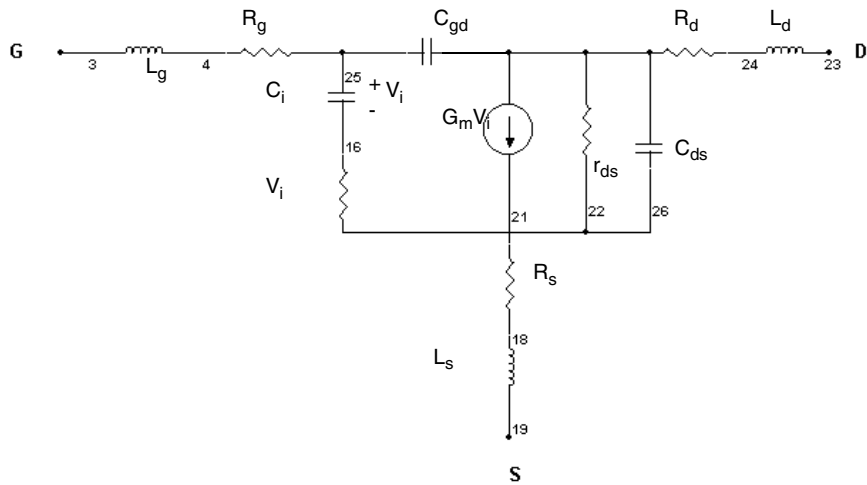
Inductors are extensively used in resonant circuits, filters, and matching networks. The following figure shows a typical inductor modeled for RF frequencies. An inductor is a wire wound or coiled. Each two windings are at close proximity, which creates a distributed capacitor,  $C_d$ . The inductor would behave like a capacitor at high frequencies. There is always a series resistance which prevents the coil from resonating. The ratio of an inductor's reactance to its series resistance is often used as a measure of the quality of the inductor. The larger the ratio, the better the inductor is.



### 14.3.3.5 Active Devices

In low frequencies, active devices are modeled using a number of ideal components such as resistors and capacitors. In high frequencies, each of these ideal components should be replaced by its equivalent, as discussed earlier. For example, a resistor should be replaced by a resistor in series with an inductor. Some simplifications would reduce redundant components. Two inductors in series, for example, can be replaced by one inductor. A typical equivalent circuit of RF transistor is shown in the following figures.





The cutoff frequency  $f_c$  can be derived from the equivalent circuit and is inversely proportional to the transit time  $\tau_c$ :

$$f_c = \frac{g_m}{2\pi C_i} = \frac{1}{2\pi \tau_c} = \frac{v_s}{2\pi L_g}$$

where  $L_g$  is the effective length of the gate, and  $v_s$  is the saturation velocity that electrons travel.

Active components included in Multisim are RF\_BJT\_NPN, RF\_BJT\_PNP, RF\_MOS\_3TDN, and tunnel diode. See Appendix M, “RF Components” for information on these components.

## 14.4 RF Instruments

Multisim's RF Design module provides two key instruments for successful RF circuit design and analysis: the Spectrum Analyzer and the Network Analyzer.

### 14.4.1 Spectrum Analyzer

#### 14.4.1.1 About the Spectrum Analyzer

The spectrum analyzer is used to measure amplitude versus frequency. This instrument is capable of measuring a signal's power and frequency components, and helps determine the existence of harmonics in the signal.

One area that has an interest in spectrum measurement is communications. For example, cellular radio systems must be checked for harmonics of the carrier signal that might interfere with other RF systems. Other interesting applications of spectrum analysis are distortions of the message modulated onto a carrier.

The spectrum analyzer displays its measurements in the frequency domain rather than the time domain. Usually the reference frame in signal analysis is time. In that case, an oscilloscope is used to show the instantaneous value as a function of time. Sometimes a sine waveform is expected but the signal, rather than being a pure sinusoidal, has a harmonic on it. As a result, it is not possible to measure the waveform's level. If the same signal were displayed on a spectrum analyzer, its amplitude would be displayed, but so would its frequency components, that is, its fundamental frequency and any harmonics it may contain.

Time domain measurements such as rise and fall times, pulse width, repetition rates, delays, etc., cannot be easily obtained in frequency domain measurements. Therefore, both techniques are important.

#### 14.4.1.2 Using Multisim's Spectrum Analyzer

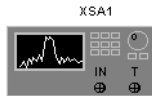
The spectrum analyzer in Multisim does not generate the noise one normally expects in a real spectrum analyzer. In reality, the noise generated by the spectrum analyzer itself (due to the random electron motion through the various circuit elements of an analyzer) is amplified by the various gain stages in the analyzer, and ultimately appears on the CRT as a noise signal below which measurement cannot be made. With Multisim's spectrum analyzer, no additional noise is introduced by the instrument itself.

A number of parameters characterize a spectrum analyzer:

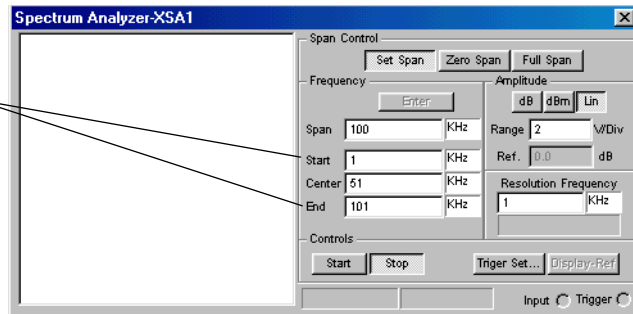
- frequency range in which the instrument operates
- frequency spans

- reference level
- measurement range.

These are all represented on the Multisim spectrum analyzer, and must be set manually.



f-start and f-end  
can be set manually.

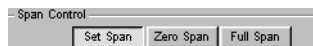


### 14.4.1.3 Frequency Range

Frequency range is the range of frequencies over which the spectrum analyzer will analyze signals. There are two frequencies that you manually set: f-start (minimum value 1kHz) is entered in the **Start** field and f-end (maximum value of 4 GHz) is entered in the **End** field. Zero frequency is not allowed for any of the frequency settings.

### 14.4.1.4 Frequency Spans

This parameter indicates the following frequency range to be displayed:



- If **Full Span** is selected, the entire instrument range, which is 1kHz to 4GHz, is displayed.
- If **Zero Span** is selected, a single frequency defined by the **Center** field is displayed.
- If **Set span** is selected, the frequency span is determined using either span control or frequency control, as explained in “14.3.1.5 Frequency Analysis” on page 14-12.

### 14.4.1.5 Frequency Analysis

There are two methods to select the frequency range:

- span control
- frequency control.

## Span Control

This technique sets the span and the center frequencies, i.e., f-center and f-span.

- To set the span and the center frequencies, click **Enter** (in the **Frequency** area of the instrument's display) to automatically calculate the values of f-start and f-end using the following expressions:

$$f\text{-start} = (f\text{-center} - f\text{-span} / 2)$$

$$f\text{-end} = (f\text{-center} + f\text{-span}/2)$$

## Frequency Control

You can define the starting and ending frequencies manually. In order to do this, you need to enter the numerical values of frequencies in the **Frequency** area of the screen. Their values should be non-zero values. When **Enter** is clicked, the center frequency (f-center) and the range of frequency displayed on the spectrum analyzer (f-span) are calculated automatically. The relationship among these parameters is expressed as follows:

$$f\text{-center} = (f\text{-start} + f\text{-end})/2$$

$$f\text{-span} = (f\text{-end} - f\text{-start})$$

Frequency	
<input type="button" value="Enter"/>	
Span	100 kHz
Start	1 kHz
Center	51 kHz
End	101 kHz

These two techniques are interrelated, that is, it is not possible to set all four parameters independently. Both techniques are useful. For example, if you want to see frequency components around one specific frequency such as 100 Mhz +/- 100kHz, then the frequency control technique is easier to apply. The center frequency, in this example, is 100 Mhz, and the span is  $2 \times (100\text{kHz}) = 200 \text{ kHz}$ .

### 14.4.1.6 Amplitude Range



You can set the amplitude range of the signal visible on the screen by choosing one of the following three options:

- **dB** - This option stands for  $20 \cdot \log_{10}(V)$ , where  $\log_{10}$  is the logarithm in base 10, and  $V$  is the amplitude of the signal. When this option is used, the signal is displayed by “dB per division”, shown in the right-hand side of the spectrum analyzer. The dB reading is of interest when measuring the power of the signal.
- **dBm** - This option stands for  $10 \cdot \log_{10}(V/0.775)$ . Zero dBm is the power dissipated in a 600 Ohm resistor when the voltage across it is 0.775 V. This power is equal to 1 mW. If the level of a signal is +10 dBm, it means that its power is 10 mW. When this option is used, the signal power is displayed based on the reference of 0 dBm. For applications in which the terminating resistor is 600 Ohm, such as in telephone lines, it is more convenient to read dBm as it is directly proportional to the power dissipation. However, in dB, you need to include the value of the resistor to find the dissipated power in the resistor. In dBm, the value of the resistor has been accounted for already.
- **LIN** - This option selects a linear display of the signal. To change the maximum amplitude displayed on the screen, enter a voltage value in the **Range** field.

### 14.4.1.7 Reference Level

The reference level is used to set the range of the input signal that can be displayed on the screen.

The axes of the spectrum analyzer are not marked by units and values. You can read the frequency and the amplitude of each point displayed on the screen by using the cursor. When the cursor is moved and placed on the point of interest, the frequency and the amplitude in V, dB, or dBm are displayed at the right-lower part of the analyzer.

You can observe more than one frequency and evaluate the results for the entire frequency range shown. You may want to know when the amplitude (in dB or dBm) of some components is above a certain limit in dB or dBm. For example, say you were interested in the (-3dB) amplitude. By locating (-3dB) points you can estimate the bandwidth of the amplifier. By clicking **Display-Ref**, you can set the reference level to (-3dB) and, using the cursor at the same time, you can find the lower edge and upper edge of the pass band.



You can also find out whether the amplitude of the signal is less than a certain value for a certain band of frequency. To do this, observe signals on the spectrum analyzer, and use the reference button. The maximum reference value in dB is set to (+30 dB). **Display-Ref** is available only if either **dB** or **dBm** are activated.

### 14.4.1.8 Frequency Resolution

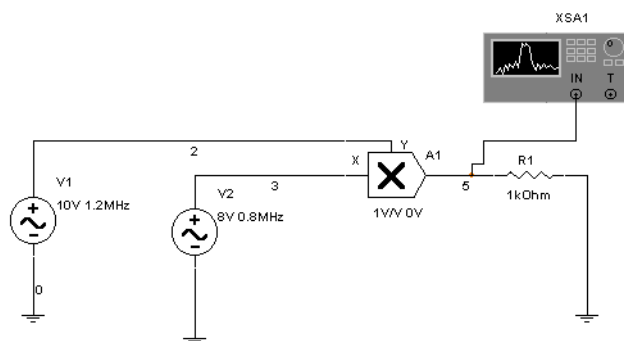
The frequency resolution is initially set to a minimum value of  $\Delta f = f_{\text{end}}/1024$ . However, you can change it to a greater value and observe the spectrum. You need to select the frequency resolution so that the frequencies are integer multiples of frequency resolutions.

**Note** For an accurate reading, the frequency components should not be below  $\Delta f$ .

### 14.4.1.9 Examples

#### Example 1

The following figure shows a mixer, which is often used in communications applications.



There are two input sinusoidal waveforms. Their frequencies are 0.8 MHz and 1.2 MHz. The amplitudes are set at 8 V and 10 V, respectively. Note that the amplitude is the peak value of the sinusoidal waveform—it is not the RMS value of the waveform. The mixer is set to multiply the signals with unity gain without introducing an offset in either of the input signals. You can expect to find two components at the output placed at  $(1.2+0.8) = 2$  MHz, and  $(1.2 - 0.8) = 0.4$  MHz.

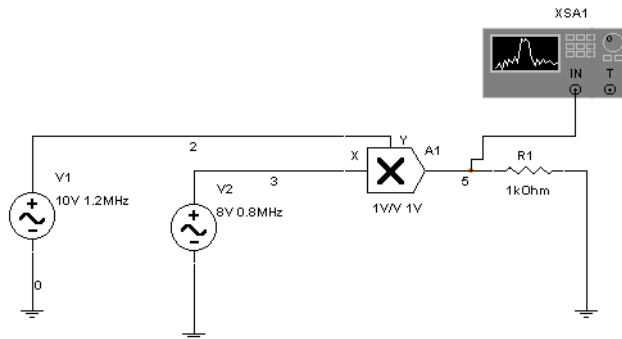
- If you want to try this example for yourself, do the following:
  1. Construct the network as shown above.
  2. Double click on the multiplier and set the gains to 1 and the offsets to zeros.

3. Double-click on the spectrum analyzer and initialize it using one start and end frequencies (example 3 in this chapter shows another way to initialize the spectrum analyzer).
  - Set **Span** to 3MHz and **Center** to 1.8MHz.
  - Click **Enter**. The frequency value of f-start is automatically set to  $(1.8\text{MHz} - 3/2 \text{ MHz}) = 300 \text{ kHz}$ . The frequency value of f-end is automatically set to  $(1.8\text{MHz} + 3/2 \text{ MHz}) = 3.3\text{MHz}$ .
  - Since the amplitude of the component is around  $(8*10)/2 = 40\text{V}$ , set the amplitude range to 100 V in “LIN” mode.
4. Run the simulator.
5. Double-click on the spectrum analyzer.
6. Click **Start** and wait until the signal stabilizes

The spectrum analyzer starts performing the Fourier Transform of the input signal in time domain. However, since it begins with only a few samples, it does not provide accurate results initially. You must wait until the screen is refreshed a few times to obtain accurate readings of the frequency components and their magnitudes. At this time, the internal frequency resolution is equal to the user-defined frequency resolution. Both of these values are shown on the device. Using the cursor on the screen, you can read the amplitude and frequency of each component. In the example, the readings are the same as the calculated values, that is, two frequency components at 2MHz and 0.4 MHz, with 40 V magnitude.

## Example 2

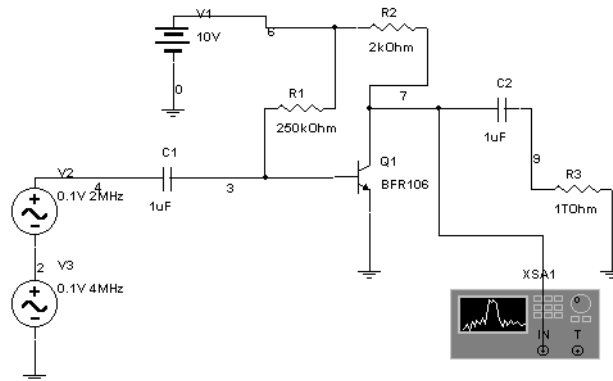
In order to run another example, you need to choose **Run** and stop the simulation running for the previous example. The second example has the same circuit structure, as shown below.



However, DC offsets (1 V) to the input and the output signals are introduced. Due to the offsets, you will have components at DC, as well as 0.8 MHz and 1.2 MHz. The spectrum analyzer shows the components and their amplitude on the screen if the same setting used in the previous example is also used here.

### Example 3

This example considers an active device that works in saturation. This structure is shown below:



➤ To initialize the spectrum analyzer:

1. Double-click on the spectrum analyzer.
2. Set **Start** to 1kHz and **End** to 11MHz.
3. Click **Enter**. The frequency value of f-span is  $(11\text{MHz} - 1\text{kHz}) = 10.999\text{kHz}$ . The frequency value of f-center is  $(11\text{MHz} + 1\text{kHz})/2 = 5.5005\text{ MHz}$ .
4. Set the range to 2 dB/division, and set the reference to 4 dB to demonstrate the application of reference dB level.

The frequency values of f-start and f-end are set so that the frequency components of interest are captured and their magnitudes can be studied. Using the spectrum analyzer, you can verify that there are more than two frequency components present at the output node. There are three components above 4 dB—zero frequency, 2MHz, and 4 MHz. Other frequency components are at higher frequencies and have dB level less than 4 dB.

## 14.4.2 Network Analyzer

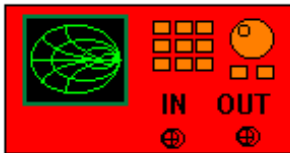
### 14.4.2.1 About the Network Analyzer

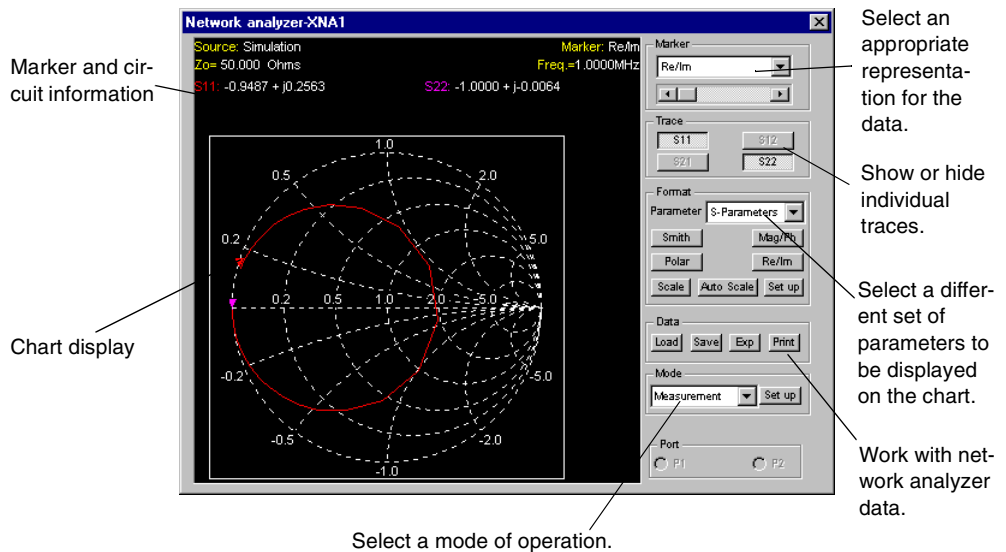
The network analyzer is used to measure the scattering parameters (or S-parameters) of a circuit, commonly used to characterize a circuit intended to operate at higher frequencies. These S-parameters are used for a variety of purposes, including in Multisim RF analyses. The network analyzer also calculates H, Y, Z parameters.

### 14.4.2.2 Using the Network Analyzer

The circuit is idealized as a two-port network. To properly use the network analyzer, the circuit must be left open at its input and output ports. During simulation the network analyzer completes the circuit being analyzed by inserting its sub-circuits. You need to remove these sub-circuits from the circuit before performing other analysis and simulation.

When you start simulation, the network analyzer automatically executes two AC analyses. The first AC analysis is applied at the input port to compute the forward parameters  $S_{11}$  and  $S_{21}$ . The second analysis is applied at the output port to compute the reverse parameters  $S_{22}$  and  $S_{12}$ . After the S-parameters are determined, you can use the network analyzer to view the data in many ways and perform further analyses based on the data.





The display of the network analyzer is divided into two regions:

- The left region contains a chart display area and a text display area that shows marker and circuit information.
- The right region contains the controls for the network analyzer.

### 14.4.2.3 Marker Controls

Select from the **Marker** drop-down list how you want data represented:

- Real/Imaginary
- Magnitude/Phase
- dB (Magnitude)/Phase.

Use the scroll bar below the list to step through each data point acquired for the circuit. Initially, the frequency points are selected in decade. The default frequency sweep range is between 1 MHz and 10 GHz. You can also use the PAGE UP and PAGE DOWN keyboard keys to navigate the data set.

### 14.4.2.4 Trace Controls

Use the **Trace** buttons to show and hide each individual trace on the chart. When the network analyzer is opened, S11 and S22 are shown in the trace area. The trace buttons change depending on what parameter or analysis is being looked at. The possible sets in “Measurement” mode are {S11, S12, S21, S22}, {Z11, Z12, Z21, Z22}, {H11, H12, H21, H22}, {Y11, Y12, Y21, Y22}, {K, |Δ|}. The possible sets in “RF Characterizer” mode are {P.G., T.P.G., A.P.G.}, {V.G.}, and {Zin, Zout}. For more about these modes, see “14.3.2.7 Mode Controls” on page 14-21.

### 14.4.2.5 Format Controls

#### Parameter Options

Select the set of parameters to be displayed on the chart. The parameters available depend on the network analyzer’s mode. In “Measurement” mode, you can choose from S-, Y-, H-, Z-parameters, and stability factors K and |Δ|. In “RF Characterizer” mode, select from Power gains, Voltage gains, and Impedances.

#### Format Buttons

Use these buttons to display data using different chart formats. The formats available depend on the selected parameter group.

Use the **Scale** button to change the scaling of the current chart. Only Polar plot, Real/Imaginary plot and Magnitude/Phase plot can be changed.

Use the **Auto Scale** button to automatically scale the data so that it can be displayed within the current chart.

Use the **Setup** button to change the various display properties of the network analyzer.

### 14.4.2.6 Data Controls

Use the **Save** button to save the current S-parameter data set to file.

Use the **Load** button to load a previously saved S-parameter data set into the network analyzer. Once the data is loaded, you can use all the functions provided by the network analyzer to view and analyze the data. The saved S-parameter file has the file extension .sp.

Use the **Exp** button to export the data set of the selected parameter group to a text file. For example, if the selected parameter group is Z-parameters displayed in magnitude/phase chart format, Z-parameters will be exported, and their values will be in magnitude/phase format.

Use the **Print** button to print the selected chart.

### 14.4.2.7 Mode Controls

From the **Mode** drop-down list, select the network analyzer mode:

- measurement mode, which provides the parameters in different formats
- RF Characterizer mode, which provides the power gains, voltage gains, and impedances seen from input and output ports
- “Match Net. designer” mode, which opens a new window, explained in “14.4.2 Matching Network Analysis” on page 14-23.

Use the first of the above choices to use the Network analyzer to perform measurement (its normal application). Use the second and third options to access the two RF analyses explained in “14.4 RF Analyses” on page 14-21.

Use the **Setup** button to enter the measurement settings for computing the circuit’s S-parameters.

## 14.5 RF Analyses

### 14.5.1 RF Characterizer Analysis

Multisim’s RF characterizer analysis tool helps designers study RF circuits in terms of the power gains, voltage gain, and input/output impedances. A typical application is an RF amplifier. The source signal at the input of an amplifier is usually provided by a receiver and its power is relatively small. The RF designer often intends to magnify the input signal and provide an output signal in terms of both voltage and current: i.e., the output power delivered to the load is considerably higher than that of the input signal. That is why the power transferability of the designed circuit is of interest. The power gains in Multisim are calculated by assuming that source and load impedances are 50 Ohm. You can change these values by clicking the **Setup** button next to the **Mode** drop-down list to specify that the RF simulator assumes

$$Z_I = Z_o \text{ and } Z_s = Z_o \text{ or } \Gamma_s = \Gamma_l = 0.$$

Another aspect of a circuit is the input and output impedances of the amplifier. An RF amplifier usually has more than one stage of amplification. Each stage of the amplifier is loaded by the input port of the next stage.

The loading effect is best understood by studying the input/output impedances. Most engineers would like to design an amplifier which has maximum input impedance in low RF frequencies, to reduce its loading effect on previous stage. On the other hand, the smaller the output impedance is, the better the output signal would be delivered. In higher RF frequencies, it is desirable to have an output impedance matching that of the load to minimize the reflection of signals. The Multisim RF characterizer analysis toolbox helps designers to study these impedances and choose the most appropriate frequency of operation.

- To use the simulator in order to read the desired variable:
    1. Connect the network analyzer to the amplifier.
    2. Run the simulator. Ignore the DC warnings and wait until the AC analyses are complete.
    3. Double click on the network analyzer.
    4. At the bottom of the right side of the control panel of the network analyzer, select “RF characterizer” from the **Mode** drop-down list.
    5. Under the **Trace** options, set the desired variable, from among T.G., A.P.G., and T.P.G. While the curves are plotted versus frequency, the numerical values are displayed at the top of curves for each frequency point.
    6. From the **Parameter** drop-down list, select “Gains”. Here, the voltage gain (V.G>) is plotted versus frequency and its value is given at the top of the curve.
- Note** Use Auto Scale each time you change the parameters to get a better reading.
7. In the **Parameter** drop-down list, select “Impedances”. The input/output impedances are provided in the form of a curve as well as printed out at the top of the curves.
  8. Use the frequency scroll bar to select the desired frequency for a specific variable.

## Power Gains

The Multisim RF Simulator calculates the General Power Gain (PG), Available Power Gain (APG) and Transducer Power Gain (TPG) for  $Z_o = 50\Omega$  at a given frequency. The dBMag is derived as  $10\log_{10} |PG|$ .

PG is defined as the ratio of the power delivered to the load and the average power delivered to the network from the input, and is given as  $PG = |S_{21}|^2 / (1 - |S_{11}|^2)$ .

The Transducer Power Gain, TPG is the ratio of the power delivered to the load to the power available from the source. For  $G_s = G_L = 0$ ,  $TPG = |S_{21}|^2$ .

The Available Power Gain, APG is the ratio of the power available from the output port of the network to the power available from the source and it is expressed as

$$APG = |S_{21}|^2 / (1 - |S_{22}|^2)$$



## Voltage Gain

Voltage Gain, VG, is obtained for  $\Gamma_s = \Gamma_l = 0$  and is expressed as  $VG = S_{21}/(1 + S_{11})$ .

Voltage Gain expressed in dBMag is calculated as  $20 \log_{10} |VG|$ .

If you observe the time domain signals of the input and output while the transistors are operating in the linear region, you find that the amplitude of the output voltage signal (when 50 Ohm load and source impedances are used) to the amplitude of the input voltage signal is the same as V.G. given by Multisim. Note, however, that V.G. is calculated using S-parameters.

## Input/Output Impedances

These values are calculated assuming  $\Gamma_s = \Gamma_l = 0$ . For this condition, we have:

$$Z_{in} = (1 + \Gamma_{in}) / (1 - \Gamma_{in}) \quad \text{where } \Gamma_{in} = S_{11} \quad \text{and}$$

$$Z_{out} = (1 + \Gamma_{out}) / (1 - \Gamma_{out}) \quad \text{where } \Gamma_{out} = S_{22}.$$

One must note that these values are normalized. The simulator prints denormalized values of  $Z_{in}$  and  $Z_{out}$ .

## 14.5.2 Matching Network Analysis

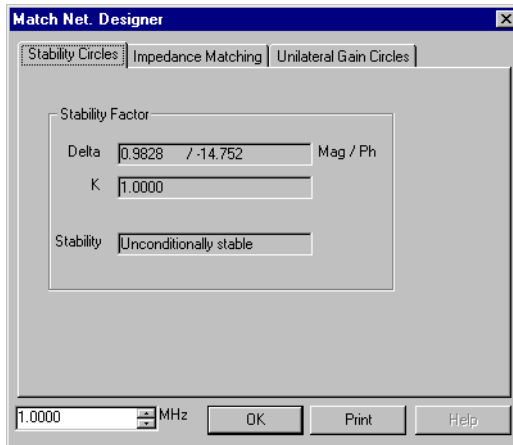
While designing RF amplifiers using Multisim, RF engineers need to analyze and, if necessary, modify circuit behavior. The Matching Network Analysis provides three options for analyzing circuit behavior:

- Stability circles
- Unilateral gain circles
- Impedance matching.

These options are described in detail in this section. Depending on the application, one or more of the options is used. For example, to design oscillators only stability circles are used. On the other hand, to match an unconditionally stable circuit, the simulator first analyzes the stability properties of the circuit then uses automatic impedance matching.

The three options are accessed from the Match Net. Designer window.

- To open the Match Net. Designer window:
  1. Double-click the Network Analyzer on the circuit window.
  2. From the **Mode** drop-down list, select “Match Net. Designer”. The Match Net. Designer screen appears:



## Stability Circles

Stability circles are used to analyze the stability of a circuit at different frequency points.

In an ideal design, when an input signal is delivered to the input port of a two-port network, the entire source signal is delivered without any loss. In practice, however, part of the input signal bounces back to the source. Then, when the amplified signal is delivered to the load impedance, part of this signal bounces back to the output port of the amplifier. The amplifier, if it is not unilateral, transfers the reflected wave back to the source impedance. A circuit is considered unstable if the signal reflected is equal to the signal delivered in either the input or the output port.

An RF engineer aims to minimize this “bounce” effect and deliver maximum signal to the load. The stability circles in the network analyzer help achieve this goal.

- To perform the analysis:
  1. Connect the biased amplifier to the network analyzer using two series capacitors (usually 100 F). The values of these capacitors are selected to minimize the numerical errors. In practice, however, two capacitors must be used to isolate the amplifier from the pre- and post-stage amplifiers in DC mode. Note that the impedance of these capacitors should not contribute to the attenuation of the input or output signal. The impedance of a capacitor is frequency dependant and is calculated using  $X_c = 1/(j\omega C)$  where “w” is  $(2\pi f)$ .

2. To activate the network analyzer, click the Simulate button on the Design Bar and choose **Run/Stop** from the pop-up menu. Wait until the AC-analyses are complete. Ignore the warning for DC-analysis.
3. Double-click on the Network Analyzer icon on the circuit window.
4. From the **Mode** drop-down list select “Match Net. Designer.”
5. From the Match Net. Designer screen, click the Stability Circles tab.
6. Select the desired operating frequency from the field in the lower left corner of the screen, and click **OK**.

The result is a Smith Chart showing an input stability circle and an output stability circle. A stability circle represents the boundary between the values of source or load impedance that cause instability and those that do not. The perimeter of the circle thus represents the locus of points which forces  $K=1$ . Note that either the inside or the outside of the circle may represent an unstable region. Unstable regions are hashed on the Smith Chart.

There are three possible scenarios on the Smith Chart, as described below:

- **None of the Smith Chart is hashed** In this case the circuit is said to be “unconditionally stable”, meaning that any area of the Smith Chart represents a valid passive source or load impedance. The designer can, then, select the input or output impedances using other criteria (such as gain or noise criteria).
- **Parts of the Smith Chart are hashed** In this case the circuit is “potentially unstable”, meaning it is possible to select passive input or output impedance and still maintain the stability of the circuit. An input impedance should fall outside the hashed area of the input stability circle to achieve stability at the input port, while an output impedance should be selected outside the output stability circle to achieve stability at the output port.
- **The entire Smith Chart is hashed** In this case, the circuit is unstable regardless of input or output impedances. The designer has a number of options to achieve stability, including changing the frequency of operation, changing the DC biasing of the transistor, changing the transistor itself, or changing the entire structure of the amplifier.

In addition to stability circles, there are two numerical values printed on the “Match Net. Designer” screen. They are  $\Delta$  and “K”. The design is unconditionally stable if  $(|\Delta| < 1)$  and  $K > 1$ . For  $K < 1$ , for example, the circuit is potentially unstable, and will most likely oscillate with certain combinations of source or load impedance.

## Unilateral Gain Circles

This option is used to analyze the unilateral property of a circuit. A transistor is said to be unilateral when there is no “bounce” effect, meaning the signal reflected from the output port to the input port is zero. This occurs if the reverse transmission coefficient,  $S_{12}$  or the reverse transducer power gain,  $|S_{12}|^2$  is equal to 0. This means that the input section of the amplifier is completely isolated from the output section. (Note that passive networks are usually not unilateral.)

The unilateral property of a network is determined by calculating the “Unilateral Figure of Merit” (U). If necessary, the frequency can be adjusted to improve the unilateral property.

➤ To calculate the “Unilateral Figure of Merit”:

1. From the Match Net. Designer screen click the Unilateral Gain Circles tab.
2. Read the value of “U” or the “Unilateral Figure of Merit”.
3. Calculate the upper and lower limits of the following inequality using “U”.

$$1/(1 + U)^2 < G_T/G_{TU} < 1/(1-U)^2$$

where,  $G_T$  – **transducer power gain**, is defined as the ratio of the output power delivered to a load by a source and the maximum power available from the source, and  $G_{TU}$  represents the transducer power gain assuming unilateral property ( $S_{12}=0$ ) for the network. You need not calculate  $G_T$  or  $G_{TU}$  since only the limits are of interest here. If the limits are close to one, or “U” is close to zero, the effect of  $S_{12}$  is small enough to assume unilateral property for the amplifier. If it is not, go to the next step.

4. Change the frequency so that the minimum “U” is read. This frequency suggests an operating point for the amplifier where the unilateral property is best met.

**Note** The operating frequency to achieve the best unilateral property for the amplifier does not necessarily coincide with the maximum gain for the circuit. The unilateral gain circles are developed to identify the best load and source impedances to minimize the error due to unilateral assumption and maintain a satisfactory level of gain. The gain circles are also used for a trade off between the gain and the bandwidth.

## Wide Band Amplifier

- To design a wide band amplifier, first find the maximum gain delivered by the circuit:
1. Double-click the network analyzer icon on the circuit window.
  2. From the **Mode** drop-down list, select “RF characterizer”.
  3. Read the value of TPG (transducer power gain or  $G_T$ ). This value is printed in dB.
  4. From the **Mode** drop-down list, select “Match Net. Designer”.
  5. From the Match Net. Designer screen, click the Unilateral Gain Circles tab.

6. Change the value of  $G_s$  and  $G_l$  manually and individually until the circles become a dot on the Smith Chart.

7. Calculate the maximum transferable power.

$$G_{\max} = G_s \text{ (dB)} + \text{TPG (dB)} + G_l \text{ (dB)}$$

The maximum gain is achieved only for a narrow band. Since the slightest change in the circuit component would change its performance, the maximum gain is not achievable in reality. For a wider band of frequency, use a gain of less than the maximum.

➤ Knowing the level of gain you want to achieve, select input and output impedances:

1. Select the desired gain (should be less than the maximum gain calculated above).

2. Choose  $G_s$  and  $G_l$  so that these three conditions are met:

- $G_s + G_{TU} + G_l < G_{\max}$
- $G_s < G_{s\max}$
- $G_l < G_{l\max}$ .

3. Enter the selected values for  $G_s$  and  $G_l$  and observe the circles. Select points on the circles closest to the center of the Smith Chart. These points are shown on the Smith Chart and circles by two triangles. The circle for  $G_l = 0$  dB always passes through the center. Therefore, the best point of the  $G_l$  to produce  $G_l = 0$  dB.

Any point selected on the Smith Chart is a normalized point. These points provide the impedances for the input and output ports that you then design manually.

**Note** To ensure these points will not cause instability, we recommend you follow the instructions in “Stability Circles” on page 14-24. If the amplifier is “unconditionally stable”, it would be stable for any passive load or source network, so you need not check stability circles in this case.

## Impedance Matching

Occasionally, a design is considered “unconditionally stable”, meaning the amplifier does not oscillate in the presence of any passive load or source impedance. In this case, you can use the impedance matching option to automatically modify the structure of an RF amplifier to achieve maximum gain impedance.

To deliver maximum power, a circuit must match at both its input and output ports. In other words, there needs to be maximum matching between the output of the amplifier and the output impedance, and the input of the amplifier and the source impedance. There are eight possible structures for each port, although only a few of these provide complete matching.

➤ To use impedance matching to find a matching network:

1. Connect the network analyzer to your amplifier as described in “Stability Circles” on page 14-24. Do not forget the capacitors in order to prevent DC loading of the network analyzer.

2. Run the simulation.
3. Double-click the network analyzer icon.
4. From the **Mode** drop-down list, select “Match Net. Designer”.
5. On the Match Net. Design screen, click the Impedance Matching tab.
6. Change the frequency to the desired operating point.
7. Enable **Auto Match**.

The instrument provides the structure as well the numerical values of components. You can click on the left and right sides of the Impedance Matching window and change the structure. However, only a few of the eight structures can provide matching.

### 14.5.3 Noise Figure Analysis

A measure of signal quality is its signal/noise ratio. Noise accompanies any signal at the input to a two-part device such as an amplifier or attenuator. Designers are interested to know how much noise is added to the output signal of a two-part network since these networks contribute to the output noise. Passive components (i.e. resistors) add Johnson noise while active components add shot or flicker noise. A measure of this signal/noise degradation is given by the noise figure:

$$F = \frac{S_s/N_s}{S_o/N_o}$$

where  $S_s/N_s$  is the input signal to noise ratio and  $S_o/N_o$  is the output signal to noise ratio.

Multisim calculates the noise figure using the equation:

$$F = \frac{N_o}{GN_s}$$

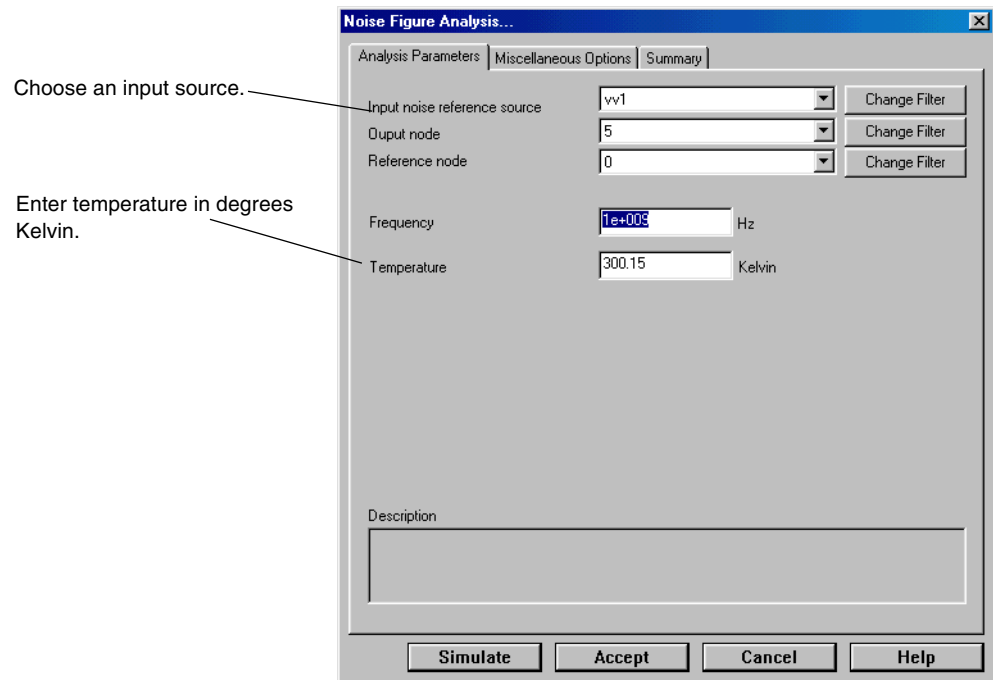
where  $N_o$  is the output noise power (which includes the noise due to two-part network and the magnified input noise)  $N_s$  is the thermal noise of the source resistor (this resistor generates noise equal to the output noise of the previous stage), and  $G$  is the AC gain of the circuit (the ratio of the output signal to the input signal of the two-part network). Note that the bandwidth of the signal has been considered in source resistor.

Finally, Multisim prints the Noise Figure in dB, that is  $10 \log_{10} (F)$ .

### 14.5.3.1 Noise Figure Analysis Tabs

Just as for other Multisim analyses, you need to fill in the appropriate fields from the Analysis Parameters tab.

Analysis parameters are shown below:



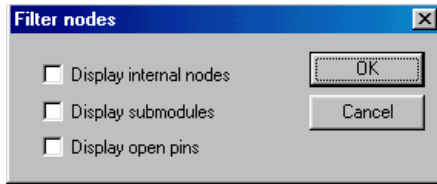
### Setting Noise Figure Analysis Parameters for Normal Use

For normal use, you only need to:

- select an input source from the **Input noise reference source** drop-down list
- select an output node from the **Output node** drop-down list
- select a reference node from the **Reference node** drop-down list
- enter a value in the **Frequency** field
- enter a value in the **Temperature** field. The default setting of 300.15 degrees Kelvin is equivalent to 27 degrees Celsius.

You can filter the variables displayed to include internal nodes (such as nodes inside a BJT model or inside a SPICE subcircuits), open pins, as well as output variables from any sub-modules contained in the circuit.

- To filter the variables displayed:
1. Click the **Change Filter** button. The Filter Nodes screen appears.



2. Enable one or more settings.
3. Click **OK**.

## 14.6 RF Model Makers

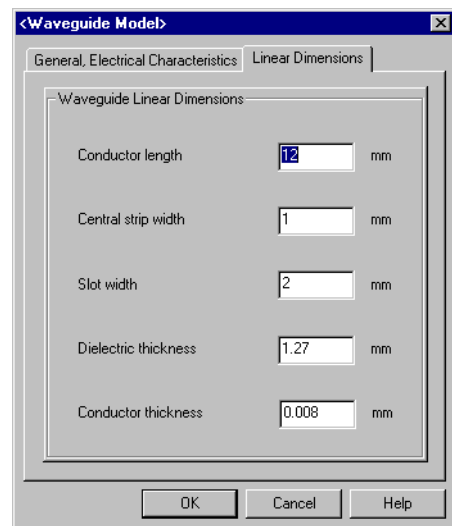
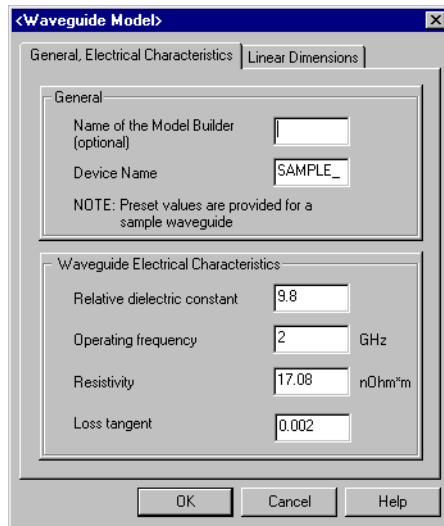
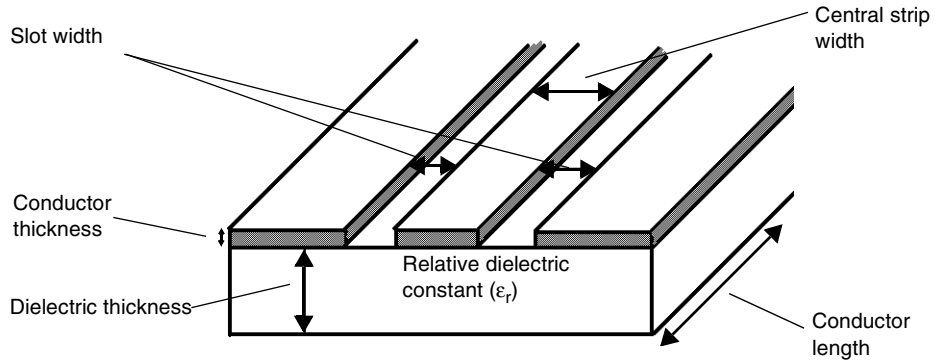
As with the other Multisim Model Makers, RF Model Makers automatically simulate models based on the input you provide. Whereas input for other model makers usually comes from data books, RF Model Makers can also receive other types of input, such as operating characteristics or physical dimensions, depending on the type of components you are modelling.

Multisim has RF Model Makers for the following types of components:



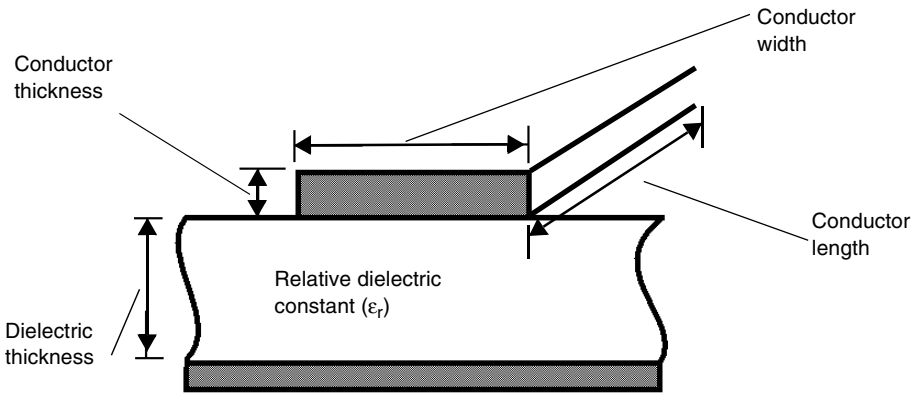
## 14.6.1 Waveguide

For Waveguide models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values:



# 14.6.2 Microstrip Line

For the Generic Microstrip Line models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values:



<Microstrip Line Model>

General, Electrical Characteristics | Linear Dimensions

General

Name of the Model Builder (optional)

Object of simulation

NOTE: Preset values are provided for a sample generic microstrip line

Microstrip Line Electrical Characteristics

Relative dielectric constant

Operating frequency  GHz

Resistivity  nOhm\*m

Loss tangent

OK Cancel Help

<Microstrip Line Model>

General, Electrical Characteristics | Linear Dimensions

Microstrip Line Linear Dimensions

Conductor length  mm

Conductor width  mm

Dielectric thickness  mm

Conductor thickness  mm

OK Cancel Help

## 14.6.3 Open End Microstrip Line

For Open End Microstrip models, enter values on the following tabs:

**<Open End Microstrip Line Model>**

General, Electrical Characteristics | Linear Dimensions

General

Name of the Model Builder (optional)

Device Name

NOTE: Preset values are provided for a sample generic microstrip line with an open end

Microstrip Line Electrical Characteristics

Relative dielectric constant

Operating frequency  GHz

Resistivity  nOhm\*m

Loss tangent

OK Cancel Help

**<Open End Microstrip Line Model>**

General, Electrical Characteristics | Linear Dimensions

Microstrip Linear Dimensions

Conductor length  mm

Conductor width  mm

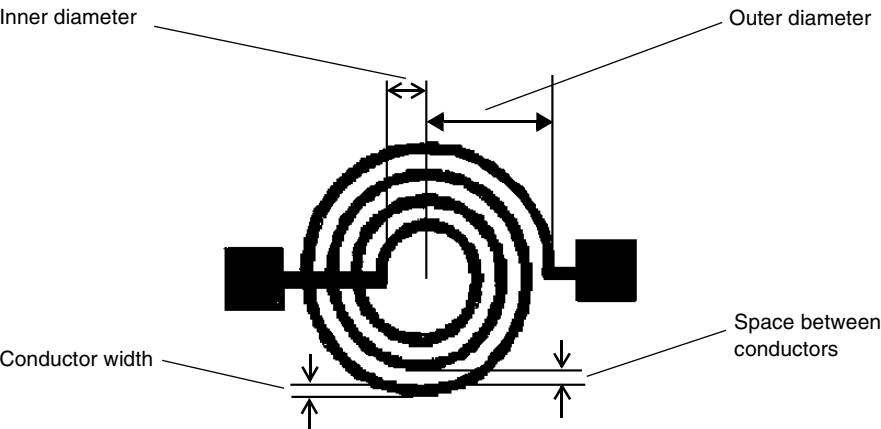
Dielectric thickness  mm

Conductor thickness  mm

OK Cancel Help

# 14.6.4 RF Spiral Inductor

For the RF Spiral Inductor models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values:



<RF Spiral Inductor Model>

General, Linear Dimensions | Electrical Characteristics

General

Name of the Model Builder (optional)

Device Name

NOTE: Preset values are provided for a sample RF Spiral Inductor

Spiral Inductor Linear Dimensions

Outer diameter  mm

Inner diameter  mm

Space between conductors  mm

Conductor width  mm

Conductor thickness  mm

OK Cancel Help

<RF Spiral Inductor Model>

General, Linear Dimensions | Electrical Characteristics

Spiral Inductor Electrical Characteristics

Relative dielectric constant

Operating frequency  GHz

Resistivity  nOhm\*m

Shunt capacitance at outer conductor  pF

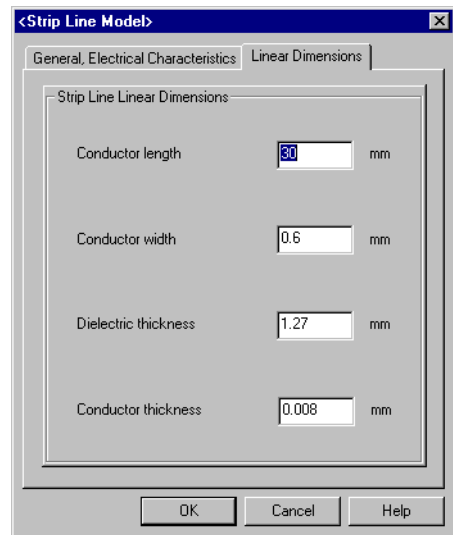
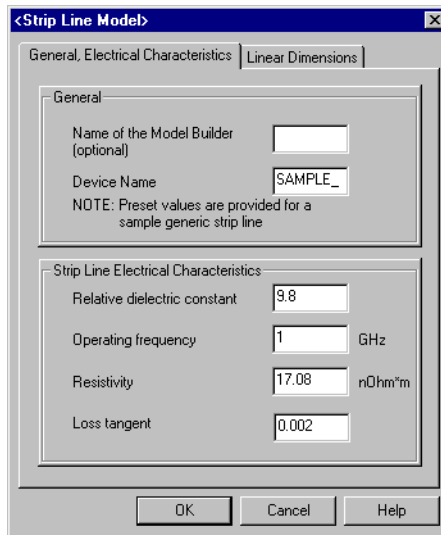
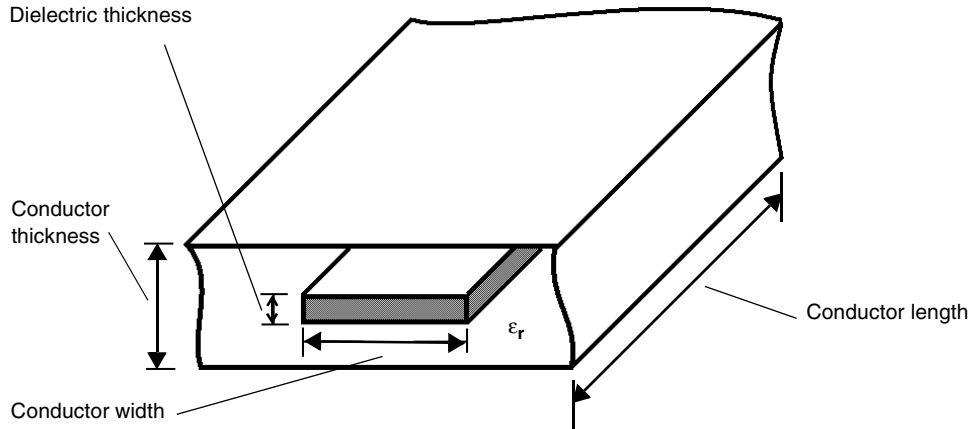
Shunt capacitance at inner conductor  pF

Total capacitance between conductors  pF

OK Cancel Help

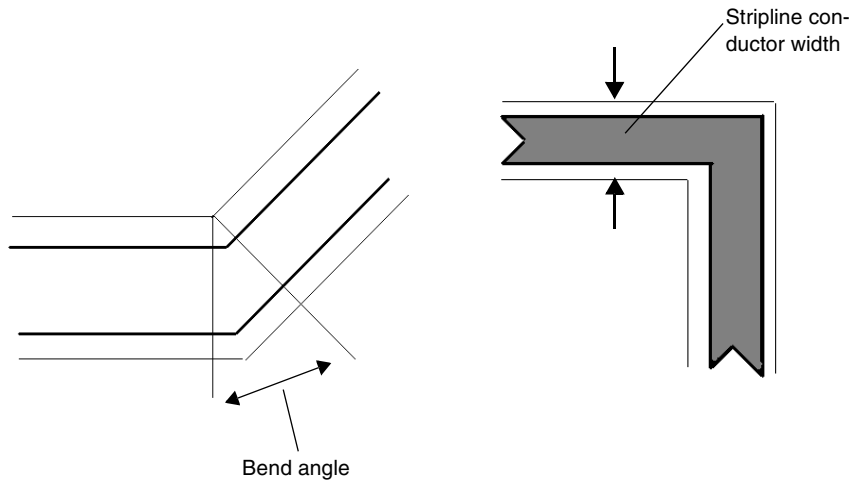
## 14.6.5 Strip Line Model

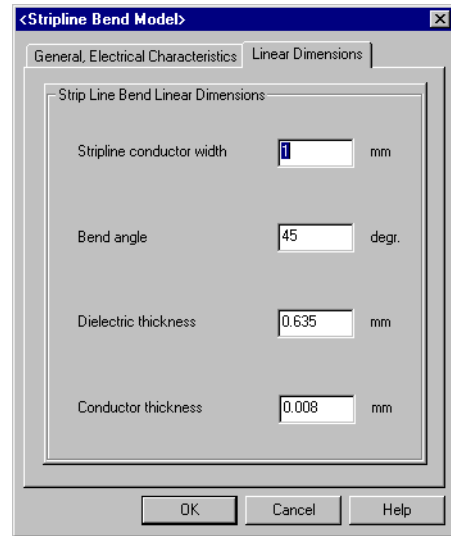
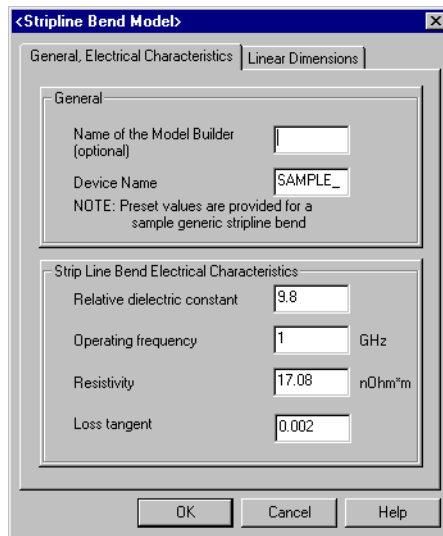
For the Strip Line models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values:





## 14.6.6 Stripline Bend

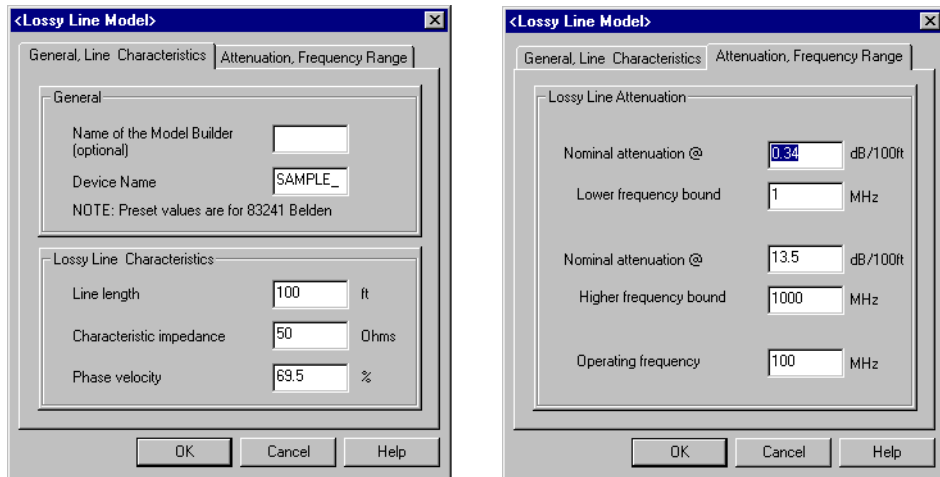
For the Stripline Bend models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values (refer to the Stripline diagram in “14.5.5 Stripline Model” on page 14-35 for Relative dielectric constance ( $\epsilon_r$ ), Dielectric thickness and Conductor thickness):





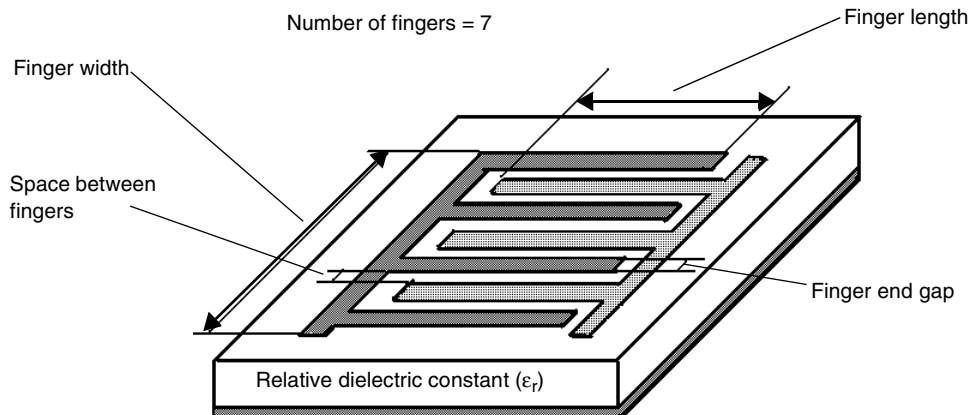
Phase velocity	 1-800-BELDEN-1	<h3>35 Ohm Transmission Cable</h3> RG-188A/U QPL 26 AWG																																
Characteristic impedance																																		
Line length	<table border="1"> <thead> <tr> <th>Trade Number</th><th>Std. Lgth. (ft.)</th><th>Std. Units (lbs.)</th></tr> </thead> <tbody> <tr> <td>83269</td><td>100 500 1000</td><td>1.4 6.4 12.4</td></tr> </tbody> </table>	Trade Number	Std. Lgth. (ft.)	Std. Units (lbs.)	83269	100 500 1000	1.4 6.4 12.4	<table border="1"> <thead> <tr> <th>Std. Units (lbs.)</th><th>AVG (stranded) Type (dia.) Nom. D.C.R.</th><th>Core O.D. Nom. O.D.</th></tr> </thead> <tbody> <tr> <td>(Kgs)</td><td>26 (7/16 0067) SCSSS 0.02 in. 84.1 ohms/MF</td><td>0.058 in. 0.108 in.</td></tr> <tr> <td>30.5 152.4 304.9</td><td>.508 mm 275.8 ohms/km</td><td>1.473 mm 2.743 mm</td></tr> </tbody> </table>	Std. Units (lbs.)	AVG (stranded) Type (dia.) Nom. D.C.R.	Core O.D. Nom. O.D.	(Kgs)	26 (7/16 0067) SCSSS 0.02 in. 84.1 ohms/MF	0.058 in. 0.108 in.	30.5 152.4 304.9	.508 mm 275.8 ohms/km	1.473 mm 2.743 mm	<table border="1"> <thead> <tr> <th>Shields</th><th>Nom. Imp. (ohms)</th><th>Vel. of Prop.</th><th>Nom. Cap.</th></tr> </thead> <tbody> <tr> <td>96% SC Braid Inner 8.5 ohms/MF</td><td>50.0</td><td>69.5%</td><td>29.2 pF/m</td></tr> <tr> <td>Inner 27.9 ohms/km</td><td></td><td></td><td>95.8 pF/m</td></tr> </tbody> </table>	Shields	Nom. Imp. (ohms)	Vel. of Prop.	Nom. Cap.	96% SC Braid Inner 8.5 ohms/MF	50.0	69.5%	29.2 pF/m	Inner 27.9 ohms/km			95.8 pF/m				
Trade Number	Std. Lgth. (ft.)	Std. Units (lbs.)																																
83269	100 500 1000	1.4 6.4 12.4																																
Std. Units (lbs.)	AVG (stranded) Type (dia.) Nom. D.C.R.	Core O.D. Nom. O.D.																																
(Kgs)	26 (7/16 0067) SCSSS 0.02 in. 84.1 ohms/MF	0.058 in. 0.108 in.																																
30.5 152.4 304.9	.508 mm 275.8 ohms/km	1.473 mm 2.743 mm																																
Shields	Nom. Imp. (ohms)	Vel. of Prop.	Nom. Cap.																															
96% SC Braid Inner 8.5 ohms/MF	50.0	69.5%	29.2 pF/m																															
Inner 27.9 ohms/km			95.8 pF/m																															
Nominal attenuation @ lower frequency bound		<table border="1"> <thead> <tr> <th colspan="4">Description:</th></tr> </thead> <tbody> <tr> <td>Insulation:</td><td>TFE Teflon</td><td colspan="2" rowspan="3">Coaxial High Temperature MIL-C-17 Cable. 26 AWG stranded silver coated copper coated steel (SCSSS) conductor with TFE Teflon® insulation. Silver coated copper braid, 96% coverage. White TFE Teflon® tape jacket. MIL-C-17D Temperature Rating : 200°C. Suggested Operating Temperature Range (Non-UL): -70°C to +200°C. Maximum Operating Voltage (Non-UL): 900 Volts RMS. Passes VW-1 Vertical Wire Flame Test. For cables manufactured to the latest government revision of other MIL-SPEC requirements, please contact your nearest Belden® Regional Sales Office. Spools may contain more than one piece. Lengths may vary ±10% from length shown.</td></tr> <tr> <td>Jacket:</td><td>TFE Teflon Tape</td></tr> <tr> <td>Plenum Version(s):</td><td>n/a</td></tr> </tbody> </table>			Description:				Insulation:	TFE Teflon	Coaxial High Temperature MIL-C-17 Cable. 26 AWG stranded silver coated copper coated steel (SCSSS) conductor with TFE Teflon® insulation. Silver coated copper braid, 96% coverage. White TFE Teflon® tape jacket. MIL-C-17D Temperature Rating : 200°C. Suggested Operating Temperature Range (Non-UL): -70°C to +200°C. Maximum Operating Voltage (Non-UL): 900 Volts RMS. Passes VW-1 Vertical Wire Flame Test. For cables manufactured to the latest government revision of other MIL-SPEC requirements, please contact your nearest Belden® Regional Sales Office. Spools may contain more than one piece. Lengths may vary ±10% from length shown.		Jacket:	TFE Teflon Tape	Plenum Version(s):	n/a																		
Description:																																		
Insulation:	TFE Teflon	Coaxial High Temperature MIL-C-17 Cable. 26 AWG stranded silver coated copper coated steel (SCSSS) conductor with TFE Teflon® insulation. Silver coated copper braid, 96% coverage. White TFE Teflon® tape jacket. MIL-C-17D Temperature Rating : 200°C. Suggested Operating Temperature Range (Non-UL): -70°C to +200°C. Maximum Operating Voltage (Non-UL): 900 Volts RMS. Passes VW-1 Vertical Wire Flame Test. For cables manufactured to the latest government revision of other MIL-SPEC requirements, please contact your nearest Belden® Regional Sales Office. Spools may contain more than one piece. Lengths may vary ±10% from length shown.																																
Jacket:	TFE Teflon Tape																																	
Plenum Version(s):	n/a																																	
Nominal attenuation @ higher frequency bound		<table border="1"> <thead> <tr> <th colspan="3">Attenuation</th></tr> <tr> <th>Freq MHz</th><th>Nom. Atten. (dB/100ft)</th><th>Nom. Atten. (dB/100m)</th></tr> </thead> <tbody> <tr><td>1.0</td><td>1.2</td><td>3.93</td></tr> <tr><td>10.0</td><td>2.7</td><td>8.65</td></tr> <tr><td>50.0</td><td>5.6</td><td>18.36</td></tr> <tr><td>100.0</td><td>8.3</td><td>27.2</td></tr> <tr><td>200.0</td><td>12.0</td><td>39.3</td></tr> <tr><td>400.0</td><td>17.5</td><td>57.4</td></tr> <tr><td>700.0</td><td>23.7</td><td>77.7</td></tr> <tr><td>900.0</td><td>27.3</td><td>89.5</td></tr> </tbody> </table>			Attenuation			Freq MHz	Nom. Atten. (dB/100ft)	Nom. Atten. (dB/100m)	1.0	1.2	3.93	10.0	2.7	8.65	50.0	5.6	18.36	100.0	8.3	27.2	200.0	12.0	39.3	400.0	17.5	57.4	700.0	23.7	77.7	900.0	27.3	89.5
Attenuation																																		
Freq MHz	Nom. Atten. (dB/100ft)	Nom. Atten. (dB/100m)																																
1.0	1.2	3.93																																
10.0	2.7	8.65																																
50.0	5.6	18.36																																
100.0	8.3	27.2																																
200.0	12.0	39.3																																
400.0	17.5	57.4																																
700.0	23.7	77.7																																
900.0	27.3	89.5																																

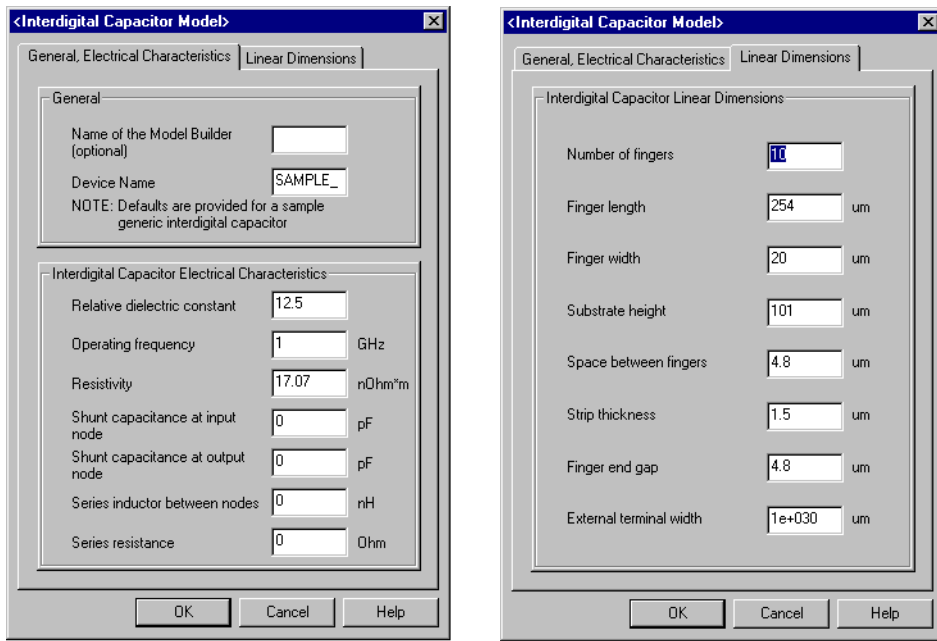




## 14.6.8 Interdigital Capacitor

For Interdigital Capacitor models, enter values in the two tabs shown below. Use the following diagram for assistance in identifying the values:





## 14.7 Tutorial: Designing RF Circuits

This tutorial is intended to:

- provide an introduction to simple RF circuit design
- demonstrate to engineers how to use Multisim for designing an RF circuit. Each design step is accompanied by the required simulation steps in Multisim.

The methodology that an RF engineer uses to design an RF circuit differs from that used for a low-frequency circuit design. An RF designer looks at performance parameters such as S-parameters, input/output impedances, power gain, noise figure, and stability factor. These design parameters are not directly available from a SPICE simulation. Impedance matching is a phase of RF circuit designs where the designer uses a Smith Chart, and calculates the values of matching elements such that maximum power is transferred to the load impedance. The Smith Chart or the calculations are not provided by SPICE simulation.

## 14.7.1 Selecting Type of RF Amplifier

Select the type of amplifier based on the application. Amplifiers designed for low-power applications are different than those for low-noise applications. Similarly, broad-band amplifiers are different in terms of design and structure than those for high-gain amplifiers. Some of the possible applications are:

- **Maximum Power Transfer** — These types of amplifiers operate in a very narrow band of frequencies.
- **Design for Specified Gain** — Designers may intentionally introduce mismatching at the input and/or the output ports to improve the bandwidth, even though the resulting power transfer is not maximal.
- **Low-Noise Amplifier Design** — In receiver applications, you need a pre-amplifier with as low a figure noise as possible since the first stage of a receiver front end has a dominant effect on the noise performance of the overall system. It is not possible to obtain both minimum noise figure and maximum gain for an amplifier.
- **Oscillators** — To produce a sinusoidal steady-state RF signal, you can use active elements and intentionally introduce negative resistance.

The network analyzer provided by Multisim is not intended for high-power RF amplifiers, because the network analyzer performs small-signal analyses of the RF network only.

In this tutorial, we will design the amplifier for maximum power transfer. Then, we will provide design steps for constant gains.

- To prepare for the tutorial, open a new circuit window.

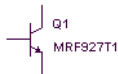
## 14.7.2 Selecting an RF Transistor

Different types of transistors are designed for a broad range of applications. For example, you may find many transistors for low noise applications. Cost of the transistor plays a significant role in choosing from those offered. In this tutorial, we will choose MRF927T1 because it is used for low power, low noise applications at relatively high frequencies.

- To select the MRF927T1 transistor in Multisim, do the following:
  1. Click the RF Parts Bin.
  2. From the **RF** Parts Bin, click the **RF\_BJT\_NPN bipolar transistor** button. The Browser screen appears.
  3. Scroll down in the component list until you find MRF927T1 and select it. Component data for that component appears in the screen.
  4. Click **OK**. The Product-RF screen closes and your cursor changes to indicate a transistor is ready to be placed.



5. Click to place the transistor on the circuit window. The results look similar to this:



### 14.7.3 Selecting a DC-operating Point

DC-operating point is referred to as  $V_{ce}$  and  $I_c$ . There are many reasons to select a specific DC-operating point. You need to consider “maximum swing” at the output, small/portable power source, and gain-bandwidth. Some DC-operating points are available in the data book, and others must be decided based on the application.

#### Vce Settings

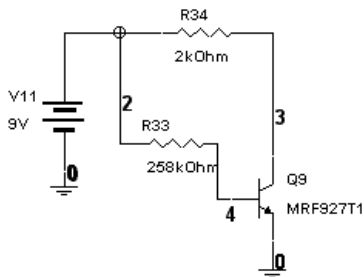
$V_{ce}$  is always less than  $V_{cc}$ , and it is usually around  $V_{cc}/2$  for maximum swing in a common-emitter configuration. For this tutorial  $V_{ce}=3V$  and  $V_{cc}=9V$  are selected.

#### Ic Settings

The nominal value of  $I_c$  for the selected transistor is 5mA. The power dissipated in transistor at any time is  $I_c \cdot V_{ce}$ . For this tutorial,  $I_c=3mA$  is selected to dissipate less power and to be close to the nominal value of  $I_c$ . This will let us achieve relatively good current-gain bandwidth and moderate voltage gain. (The voltage gain is maximum at  $I_c=1mA$  and the current gain-bandwidth is maximum at  $I_c=5mA$ .)

### 14.7.4 Selecting the Biasing Network

There are a number of possible structures to select from for proper DC-biasing of the network. It is important to note that the performance of the transistor and the amplifiers depends on DC-operating points. The following figure shows one possible biasing network.



This is the simplest structure for a biasing network. However, its thermal stability is poor. To find the resistor values for this structure you need to know five values:  $V_{ce}$ ,  $I_c$ ,  $V_{cc}$ ,  $V_{be}$ , and  $\beta$  (which is the DC-current gain of the transistor and is given in most data books).  $\beta$  relates  $I_c$  to  $I_b$  as  $\beta = I_c/I_b$ .  $V_{be}$  is the base-emitter voltage of transistor when it is active, and is typically 0.7 V. Both  $\beta$  and  $V_{be}$  depend on the values of  $I_c$  and  $I_b$ . The initial design process starts with typical values of  $R_c$  and  $R_b$  in the structure shown above. However, if accuracy is critical, you should use Multisim to ensure that the values of  $I_c$  and  $V_{ce}$  are as intended. In this tutorial, these selections are used:

- $V_{ce}=3V$
- $I_c=3mA$
- $V_{cc}=9V$
- $V_{be}=0.7V$
- $\beta=100$ .

The initial values of  $R_c$  and  $R_b$  are calculated as shown below.

$$R_c = (V_{cc}-V_{ce})/I_c = (9V - 3V)/ 3mA = 2 \text{ K}\Omega$$

$$I_b = I_c/\beta = 3mA/100 = 30 \mu A$$

$$R_b = (V_{cc} - V_{be})/I_b = (9V - 0.7V)/30 \mu A = 277 \text{ K}\Omega$$

➤ To select the DC operating points:

1. Draw the circuit shown above with  $R_b=277 \text{ K}\Omega$  and  $R_c=2\text{K}\Omega$ . Note that  $V_{cc}=9V$  and the transistor is MRF927T1.
2. Click the Analysis button on the Design Bar and choose **DC Operating Point**. The DC Operating Point Analysis screen appears.
3. Select the nodes representing the collector and base of the transistor.
4. Click **Plot during Simulation**.
5. Click **Simulate**.



The results will show a reading of  $V_c$  and  $V_b$ . In the DC-biasing network,  $V_c$  is  $V_{ce}$  and  $V_b$  is  $V_{be}$ . The first readings are  $V_{ce}=3.33V$  and  $V_{be}=0.8V$ . You can modify the value of either  $R_c$  or  $R_b$  or both, to achieve the desired DC operating point. After a number of iterations, you arrive at  $R_c=2\text{K}\Omega$  and  $R_b=258\text{K}\Omega$ . Reading the values of  $V_{ce}$  and  $V_{be}$  for the final simulation, you will note that

$$\beta = I_c/I_b = R_b \cdot (V_{cc}-V_{ce}) / [R_c \cdot (V_{cc}-V_{be})] = 94.36$$

which is close to the initial value of  $\beta$ .

### 14.7.4.1 Selecting an Operating Frequency Point

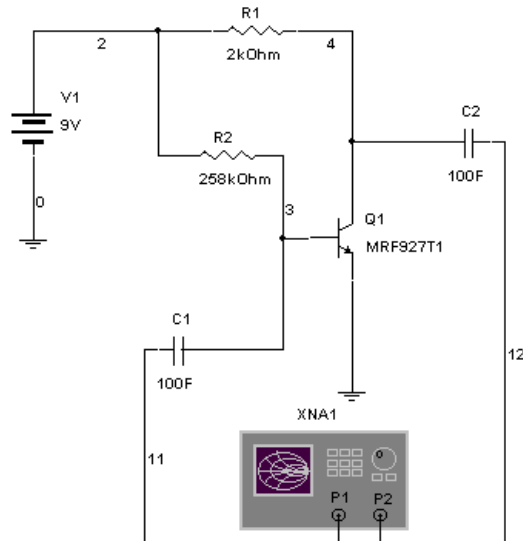
The operating frequency point chosen depends on the type of application, and is usually defined in the design specifications. For this tutorial, you can assume a single (center) frequency analysis of 3.02 GHz.

### 14.7.4.2 Analyzing the RF Network

➤ To perform the simulation:

1. Connect the biased transistor to the network analyzer using two series capacitors. These capacitors are used to isolate the network analyzer from the biasing network in DC mode. This step is necessary whenever the biasing network is important, that is, for active circuits only.

The connection should look like this:

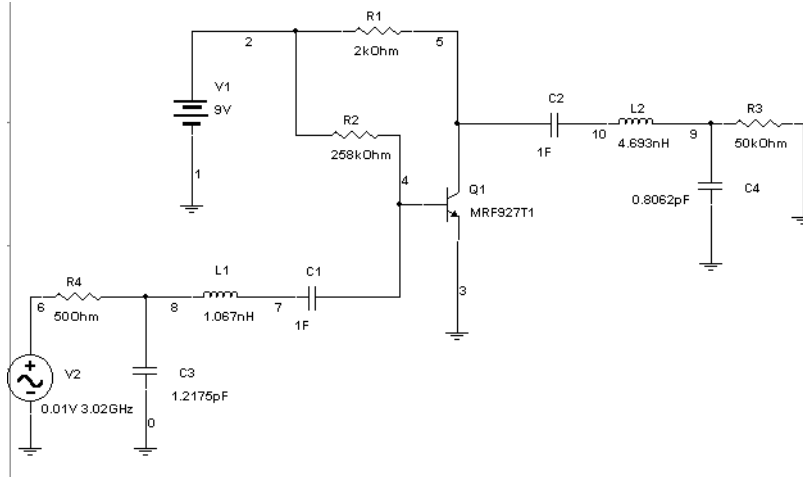


2. Choose **Simulate/Run** and wait until the AC analyses are complete. Ignore the warning for DC analyses.
3. Double-click on the Network Analyzer icon on circuit window, and from the **Mode** drop-down list, select “Match Net. designer”.
4. On the Match Net. designer window that appears, do the following:
  - set the frequency to 3.02 GHz
  - since the circuit is “unconditionally stable” for this frequency point, click **Impedance Matching**

- since the circuit is “unconditionally stable”, automatic impedance matching is possible. Click **Auto Match**.

The window provides the structure and the values necessary for conjugate matching; hence, maximum power transfer is achieved.

Below is our design for maximum power transfer at  $f=3.02$  GHz:



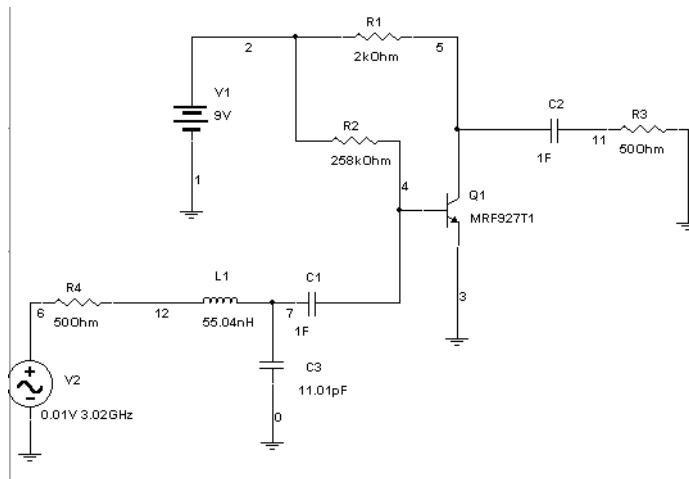
**Note** 1F is needed to isolate the active network from its matching network to keep the transistor in its biasing state.

Impedance matching yields maximum power transfer for a very narrow bandwidth. In real applications, however, you need to balance the power transfer and the bandwidth. For this reason, mismatching is intentionally introduced to the circuit.

- To design the amplifier for a constant gain less than the maximum possible gain, for better frequency response:
  1. Open the DC biased transistor circuit discussed previously.
  2. Open the network analyzer and change settings as follows:
    - select “RF Characterizer” from the **Mode** drop-down list
    - select “Power Gain” from the **Parameter** drop-down list
    - select “dB MAG” from the **Marker** drop-down list
    - set the frequency to 3.02 GHz
    - click **TPG** (Transducer Power Gain) under **Trace** and set its value to 4.3652 dB for our example.
  3. From the **Mode** drop-down list, select “Match Net. Designer” and, in the Match Net. designer screen, click **Unilateral Gain Circles**.
  4. Change the value of  $G_s$  and  $G_l$  manually and individually until the circles become a dot on the Smith Chart. The example shows  $G_s=0.042$  dB and  $G_l=1.2650$  dB.

5. Calculate the maximum transferable power.  $P_{max} = 0.042 + 4.3652 + 1.2650 = 5.6722$  dB.
6. Select the gain desired. This gain should be less than 5.6752 dB. 3.5302 dB was selected as the power gain.
7. Choose  $G_s$  and  $G_l$  so that  $G_s + 4.3652 + G_l = 3.5302$  dB and  $G_s < 0.042$  dB and  $G_l < 1.2650$  dB. You selected  $G_s = -0.08350$  dB and  $G_l = 0$  dB.
8. Enter the selected values for  $G_s$  and  $G_l$  and observe the circles. Select point or points on the circles which are closest to the center of the Smith Chart. The circle for  $G_l = 0$  dB always passes through the center. Therefore, the best point of the  $G_l$  is the center itself. This means that 50 Ohm load is sufficient to produce  $G_l = 0$  dB. The best point on  $G_s$  for the example is  $Z_l = 2$  (normalized). Using this value, you can design the matching network at the input port of the amplifier. You must make sure that the selected point or points are stable. Therefore, it is recommended that you go back to "Stability Circles", and confirm the stability of the design. Since the amplifier is unconditionally stable, it would be stable for any passive load or source network. Hence, you need not check the stability circles.

The complete amplifier is shown in the following figure:



**Note** The matching elements are calculated manually in this example.



# Index

## A

- AC analysis 8-11
- AC sensitivity analysis 8-27
- adding components 5-9
- adding subcircuits to circuit 3-27
- addressing, word generator 6-32
- algorithm
  - Gmin stepping 7-9
  - source stepping 7-10
- ammeter
  - multimeter measurement options 6-20
- analyses
  - about 8-1
  - AC 8-11
  - AC sensitivity 8-27
  - Analysis Parameters tab 8-3
  - audit trail 8-53
  - batched 8-51
  - cutting/copying/pasting pages, graphs, and charts 8-63
  - DC operating point 8-9
  - DC sensitivity 8-27
  - determining component use 3-20
  - distortion 8-23
  - Fourier 8-16
  - incomplete 8-9
  - miscellaneous options tab 8-6
  - Monte Carlo 8-47
  - noise 8-19
  - noise figure 14-18
  - options 8-65
  - output variables tab 8-3
  - parameter sweep 8-31
  - performing (general instructions) 8-2
  - pole zero 8-42
  - printing graphs and charts 8-64
  - results 8-53
  - summary tab 8-8

- temperature sweep 8-34
- transfer function 8-36
- transient 8-13
- user-defined 8-52
- viewing charts 8-63
- viewing graphs 8-57
- working with pages 8-56
- worst case 8-38
- analysis default settings for instruments 6-5
- analysis output, manipulating 9-1, 9-2, 9-7
- Analysis Parameters tab, about 8-3
- assigning faults to components 13-1
- audit trail 8-53
- Auto Fault option 13-3
- automatic wiring 3-11
- autosave 2-8
- axes, Bode plotter settings 6-7

## B

- batched analyses 8-51
- Bode plotter
  - about 6-6
  - axes settings 6-7
  - magnitude 6-7
  - phase 6-7
  - readouts 6-8
  - settings 6-7
- Boolean expressions, entering 6-14
- Browser screen 3-6
- browsing database 3-3
- BSpice model 5-28
- Bspice support 7-5

## C

- changing component value/model 3-18
- channel settings 6-26
- charts
  - cut/copy/paste 8-63

- printing 8-64
  - using in postprocessor 9-7
  - viewing 8-63
- circuit
  - adding instruments to 6-2
  - adding subcircuits 3-27
  - consistency, checking in simulation 7-4
  - controlling display 2-5
  - equation 7-7
  - printing files 3-28
  - setting up as subcircuit 3-27
  - simulation. *See* simulation
- circuit restrictions
  - setting 13-6
- circuit window
  - placing components 3-4
- clock
  - logic analyzer 6-17
- code model
  - about 76
  - creating 77
  - implementation file 84
  - interface file 78
- color
  - component 3-10
  - schemes 2-5
  - wire 3-14
- complex digital ICs, modeling 10-3
- Complex Programmable Logic Device. *See* CPLD
- component color 3-10
- component detail report 11-3
- Component Editing
  - about 5-1
  - Footprint tab 5-29
  - General tab 5-7
- component symbol
  - editing 5-16
- component value/model, changing 3-18
- component wizard 5-9
- components
  - about 4-1
  - adding 5-9
  - assigning faults 13-1
  - Auto Fault option 13-3
  - changing value/model 3-18
  - classification in database 4-3
  - copying model 5-27
  - copying symbols 5-17
  - creating model 5-25
  - creating symbols 5-18
  - determining use in analyses 3-20
  - displaying information 3-16
  - editing 5-5
  - editing model 5-25
  - editing symbol 5-16
  - flipping 3-15
  - general properties 5-7
  - information stored 4-23
  - labels, assigning 3-22
  - moving 3-8
  - package information 5-29
  - pins 5-29
  - placed, properties 3-16
  - placing 3-4, 3-8
  - placing on circuit window 3-4
  - reference ID, assigning 3-22
  - removing 5-14
  - rotating 3-15
  - searching for 4-21
  - using global 4-26
  - virtual 3-4
  - wiring 3-10
- connector, adding 3-14
- convergence assistance 7-9
- copying
  - component model 5-27
  - component symbol 5-17
- CPLD 10-3
- creating components 5-9
- customizing interface 2-4
- cut/copy/paste pages, graphs, and charts 8-63

## D

data ready 6-33

database

- classification of parts 4-3

- component classification 4-3

- family list 11-2

- levels 4-1

- selecting components 3-3

- structure of 4-1

DC operating point analysis 8-9

DC sensitivity analysis 8-27

decibels 6-21

default analysis, postprocessor 9-4

default instrument settings 6-5

default path 2-8

deleting components 5-14

Description window 3-25

description, adding to circuit 3-25

design bar 2-3

DIN symbols 3-3

diode model maker 45

display details 2-5

displaying information about placed components  
3-16

displaying or hiding grid, title block and page  
borders 3-2

distortion analysis 8-23

distortion analyzer 6-9

- harmonic distortion 6-10

- SINAD 6-10

dual-channel oscilloscope 6-24

## E

Edit menu 2-13

editing

- component symbol 5-16

- components 5-5

equation solution 7-7

error log/audit trail 8-53

Excel, exporting simulation results to 12-3

exporting

simulation results to Excel 12-3

simulation results to MathCAD 12-3

## F

faults, assigning to components 13-1

Field Programmable Gate Array. *See* FPGA

File menu 2-11

flipping components 3-15

Footprint tab 5-29

Fourier analysis 8-16

FPGA 10-3

frequency 6-33

function generator

- about 6-11

- rise time 6-12

- signal options 6-12

## G

general component properties 5-7

General tab 5-7

global components, using 4-26

global restrictions

- passwords 13-9

- setting 13-4

Gmin stepping 7-9

Grapher 8-54

Grapher window 8-54

graphs

- about Grapher window 8-54

- cut/copy/paste 8-63

- printing 8-64

- using in post processor 9-7

- viewing 8-57

grid

- displaying or hiding 3-2

- showing 2-6

grounding oscilloscope 6-25

## H

hardware description language *See* HDL

harmonic distortion

- distortion analyzer 6-10
- HDLs
  - about 10-2
  - in Multisim 10-3
- hierarchy 3-26
- horizontal Bode plotter settings 6-7
- I**
  - implementation file (code model) 84
  - importing model 5-28
  - In Use list 3-8
  - incomplete analyses 8-9
  - information stored for components 4-23
  - instruments
    - about 6-1
    - adding to a circuit 6-2
    - Bode plotter 6-6
    - default analysis settings 6-5
    - distortion analyzer 6-9
    - function generator 6-11
    - logic analyzer 6-15
    - logic converter 6-13
    - measurement options 6-20
    - multimeter 6-19
    - multiple 6-4
    - network analyzer 6-33
    - oscilloscope 6-24
    - spectrum analyzer 6-28
    - using 6-3
    - wattmeter 6-29
    - word generator 6-30
  - integration order, maximum 7-9
  - integration, numerical 7-8
  - interactive simulation 7-4
  - interdigital model 14-36
  - interface 2-2
    - customizing 2-4
    - design bar 2-3
    - elements 2-2
  - interface file (code model) 78
  - internal settings of multimeter 6-22

## J

- junction, adding 3-14

## L

- labels
  - assigning to components 3-22
  - assigning to nodes 3-23
- levels of database 4-1
- loading model 5-28
- logic analyzer
  - about 6-15
  - adjusting clock 6-17
  - reset 6-16
  - start 6-16
  - stop 6-16
  - triggering 6-18
- logic converter 6-13
  - constructing a truth table 6-14
  - deriving truth table from a circuit 6-13
  - entering Boolean expressions 6-14
- lossy line model 14-35

## M

- magnitude 6-7
- manual wiring 3-12
- MathCAD, exporting simulation results to 12-3
- maximum integration order 7-9
- measurement options
  - ammeter 6-20
  - decibels 6-21
  - ohmmeter 6-21
  - voltmeter 6-20
- menu
  - Edit 2-13
  - File 2-11
  - pop-up 3-30
  - Simulate 2-19
  - Symbol Editor 5-19
  - Tools 2-25
  - Transfer 2-24
  - View 2-17

- Window 2-26
- microstrip line model 14-29
- microstrip open end model 14-30
- miscellaneous options tab, about 8-6
- model
  - changing component 3-18
  - copying 5-27
  - creating *See* model creation
  - loading 5-28
- model creation
  - by importing 5-28
  - using code modeling 76
- model maker
  - diode 45
  - MOSFET 50
  - silicon controlled rectifier 67
- module
  - RF 14-1
- Monte Carlo analysis 8-47
- MOSFET model maker 50
- moving component 3-8
- multimeter 6-19
  - internal settings of 6-22
  - signal mode 6-22
  - using ammeter measurement option 6-20
  - using decibel measurement option 6-21
  - using ohmmeter measurement option 6-21
  - using voltmeter measurement option 6-20
- multiple instruments 6-4
- multiple traces, postprocessor 9-6
- Multisim
  - about 1-2
  - features 1-2
  - interface 2-2

## N

- network analyzer 6-33, 14-15
- nodes, assigning labels 3-23
- noise analysis 8-19
- noise figure analysis 14-18
- numerical integration 7-8

## O

- ohmmeter 6-21
- oscilloscope
  - about 6-24
  - channel settings 6-26
  - grounding 6-25
  - time base 6-25
  - trigger 6-27
- output variables tab, about 8-3

## P

- package information 5-29
- page borders, displaying or hiding 3-2
- page bounds, showing 2-6
- pages
  - cut/copy/paste 8-63
  - using in analyses 8-56
  - using in postprocessor 9-7
- parameter sweep analysis 8-31
- passwords, creating/changing 13-9
- phase 6-7
- pin information 5-29
- pins, adding to symbols 5-23
- placed components, properties 3-16
- placing components 3-4, 3-8
- PLD, about 10-2
- pole zero analysis 8-42
- pop-up menu 3-30
- postprocessor
  - about 9-1, 9-2, 9-8
  - basic steps 9-2
  - creating multiple traces 9-6
  - functions 9-8
  - pages, graphs and charts 9-7
  - screen 9-2
  - using the default analysis 9-4
  - variables 9-8
- Power Pro
  - using code modeling 76
- pre-defined fields in database 4-24
- print page setup 2-9

- printing
  - circuit files 3-28
  - graphs and charts 8-64
- Programmable Logic Device. *See* PLD
- properties of placed components 3-16
- PSpice model 5-28

## R

- readouts 6-8
- Recent Files 2-13
- reference ID, assigning to components 3-22
- Remote 13-10
- removing components 5-14
- reports
  - component detail 11-3
  - database family list 11-2
- resizing toolbars 2-9
- restrictions
  - about 13-3
  - setting circuit 13-6
  - setting global 13-4
- RF components
  - about 14-2
  - interdigital model 14-36
  - lossy line model 14-35
  - microstrip line model 14-29
  - microstrip open end model 14-30
  - RF model makers 14-27
  - RF spiral inductor model 14-31
  - strip line model 14-32
  - stripline bend model 14-33
  - waveguide model 14-28
- RF instruments
  - network analyzer 14-15
  - spectrum analyzer 14-9
- RF module 14-1
  - about 14-1
  - components (*see also* RF components) 14-2
  - instruments (*see also* RF instruments) 14-9
- RF simulation 7-10
- RF tutorial 14-37
- rise time 6-12

- rotating components 3-15

## S

- schematic capture 3-1
- search results 4-23
- searching for components 4-21
- setting circuit restrictions 13-6
- setting global restrictions 13-4
- setting up a circuit as a subcircuit 3-27
- settings, oscilloscope channel 6-26
- sheet size 2-6
- sheet size, setting up 3-2
- show
  - grid 2-6
  - page bounds 2-6
  - title block 2-6
- signal mode 6-22
- signal options 6-12
- silicon controlled rectifier model maker 67
- Simulate menu 2-19
- simulation
  - about 7-1
  - Bspice/Xspice support 7-5
  - checking circuit consistency 7-4
  - choosing type 7-1
  - circuit 7-6
  - circuit equation 7-7
  - equation solution 7-7
  - Gmin stepping 7-9
  - interactive 7-4
  - maximum integration order 7-9
  - miscellaneous SPICE capabilities 7-4
  - numerical integration 7-8
  - RF 7-10
  - source stepping 7-10
  - stages of 7-6
  - starting and stopping 7-3
  - supported types 7-2
  - using 7-3
  - VHDL 7-11
- SINAD 6-10
- solution, equation 7-7

- source stepping 7-10
- spectrum analyzer 6-28, 14-9
- SPICE simulation
  - BSpice/Xspice support 7-5
  - circuit 7-6
  - circuit equation 7-7
  - equation solution 7-7
  - Gmin stepping 7-9
  - maximum integration order 7-9
  - miscellaneous capabilities 7-4
  - numerical integration 7-8
  - source stepping 7-10
- spiral inductor, RF 14-31
- standard searching 4-21
- strip line model 14-32
- stripline bend model 14-33
- structure of database 4-1
- subcircuit
  - adding to circuit 3-27
  - setting up a circuit as 3-27
- subcircuits 3-26
- summary tab, about 8-8
- symbol
  - adding pins 5-23
  - copying 5-17
  - creating 5-18
  - editing 5-16
  - labels 5-21
  - pin 5-22
  - shape 5-21
- Symbol Editor
  - menus 5-19
  - palette 5-20
  - screen 5-18
- symbol set 2-8
  - choosing 3-3
- system toolbar 2-10

## T

- temperature sweep analysis 8-34
- time base 6-25
- title block

- about 3-24
  - displaying or hiding 3-2
  - showing 2-6
- tolerances 4-26
- toolbars
  - resizing 2-9
  - system 2-10
- Tools menu 2-25
- transfer function analysis 8-36
- Transfer menu 2-24
- transfer to Ultiboard PCB Layout 12-1
- transient analysis 8-13
- trigger, oscilloscope 6-27
- triggering
  - logic analyzer 6-18
  - word generator 6-33
- truth table
  - constructing 6-14
  - deriving from circuit 6-13

## U

- Ultiboard, transfer to 12-1
- user interface
  - about 2-2
  - customizing 2-4
  - design bar 2-3
  - elements 2-2
- user preferences, about 2-4
- user-defined analysis 8-52
- using instruments 6-3

## V

- value, changing component 3-18
- variables, postprocessor 9-8
- vertical Bode plotter settings 6-7
- VHDL
  - introduction to 10-4
  - simulation 7-11
- View menu 2-17
- virtual components 3-4
- voltmeter

multimeter measurement options 6-20

## **W**

wattmeter 6-29

waveguide model 14-28

Window menu 2-26

wire

color 3-14

re-shaping 3-13

wiring

automatic 3-11

combining automatic and manual 3-13

manual 3-12

wiring components 3-10

wiring options 2-8

word generator 6-30

addressing 6-32

enabling data 6-33

entering words 6-31

setting clock frequency 6-33

triggering 6-33

using word patterns in 6-32

worst case analysis 8-38

## **X**

Xspice

model 5-28

simulation 7-5

## **Z**

zoom level, default 2-6