FPGA Editor Guide

Introduction

Getting Started

Using the FPGA Editor

Working with Physical Macros

Customizing the FPGA Editor

FPGA Editor Files

Glossary

FPGA Editor Guide

About This Manual

This manual describes the Xilinx FPGA Editor, a graphical application used for displaying and configuring Field Programmable Gate Arrays (FPGAs).

Before using this manual, you should be familiar with the Xilinx design flow, including design entry and implementation. For more information on the Xilinx tools, refer to the *Development System Reference Guide*, *Design Manager/Flow Engine Guide*, and the *Alliance Series 3.1i Quick Start Guide*.

Note This Xilinx software release is certified as Year 2000 compliant.

Manual Contents

This manual covers the following topics.

- Chapter 1, "Introduction," provides an overview of the FPGA Editor.

- Chapter 2, "Getting Started," includes information on how to start and exit the FPGA Editor, and describes the user interface.

- Chapter 3, "Using the FPGA Editor," describes how to perform various operations on your design files.

- Chapter 4, "Working with Physical Macros," explains how to create and work with macros.

- Chapter 5, "Customizing the FPGA Editor," describes how to customize the FPGA Editor to suit your needs.

- Appendix A, "FPGA Editor Files," provides a listing of the files used by the FPGA Editor.

- • "Glossary" describes the basic terminology used in the FPGA Editor manual.

Additional Resources

For additional information, go to `http://support.xilinx.com`. The following table lists some of the resources you can access from this Web site. You can also directly access these resources using the provided URLs.

Resource	Description/URL
Tutorials	Tutorials covering Xilinx design flows, from design entry to verification and debugging `http://support.xilinx.com/support/techsup/tutorials/index.htm`
Answers Database	Current listing of solution records for the Xilinx software tools Search this database using the search function at `http://support.xilinx.com/support/searchtd.htm`
Application Notes	Descriptions of device-specific design techniques and approaches `http://support.xilinx.com/apps/appsweb.htm`
Data Book	Pages from *The Programmable Logic Data Book*, which contain device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging `http://support.xilinx.com/partinfo/databook.htm`
Xcell Journals	Quarterly journals for Xilinx programmable logic users `http://support.xilinx.com/xcell/xcell.htm`
Technical Tips	Latest news, design tips, and patch information for the Xilinx design environment `http://support.xilinx.com/support/techsup/journals/index.htm`

Conventions

This manual uses the following conventions. An example illustrates each convention.

Typographical

The following conventions are used for all documents.

- `Courier font` indicates messages, prompts, and program files that the system displays.

  ```
  speed grade: - 100
  ```

- **`Courier bold`** indicates literal commands that you enter in a syntactical statement. However, braces "{ }" in Courier bold are not literal and square brackets "[ ]" in Courier bold are literal only in the case of bus specifications, such as bus [7:0].

  `rpt_del_net=`

  `Courier bold` also indicates commands that you select from a menu.

  `File` → **`Open`**

- *Italic font* denotes the following items.

  - Variables in a syntax statement for which you must supply values

    `edif2ngd` *`design_name`*

  - References to other manuals

See the *Development System Reference Guide* for more information.

♦ Emphasis in text

If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.

• Square brackets " [ ]" indicate an optional entry or parameter. However, in bus specifications, such as bus [7:0], they are required.

```
edif2ngd [option_name] design_name
```

• Braces " { }" enclose a list of items from which you must choose one or more.

```
lowpwr ={on|off}
```

• A vertical bar " | " separates items in a list of choices.

```
lowpwr ={on|off}
```

• A vertical ellipsis indicates repetitive material that has been omitted.

```
IOB #1: Name = QOUT'

IOB #2: Name = CLKIN'

.

.

.
```

• A horizontal ellipsis "...." indicates that an item can be repeated one or more times.

```
allow block  block_name loc1 loc2locn;
```

Online Document

The following conventions are used for online documents.

• Red-underlined text indicates an interbook link, which is a cross-reference to another book. Click the red-underlined text to open the specified cross-reference.

- Blue-underlined text indicates an intrabook link, which is a cross-reference within a book. Click the blue-underlined text to open the specified cross-reference.

Contents

Chapter 3   Using the FPGA Editor

Chapter 4   Working with Physical Macros

Chapter 5   Customizing the FPGA Editor

Appendix A FPGA Editor Files

Chapter 1

Introduction

This chapter provides an overview of the FPGA Editor and includes the following sections.

- "Overview"

- "Design Flow"

- "FPGA Editor Files"

- "Architecture Support"

Overview

The FPGA Editor is a graphical application for displaying and configuring Field Programmable Gate Arrays (FPGAs). The FPGA Editor reads from and writes to Native Circuit Description (NCD) files, macro files (NMC), and Physical Constraints Files (PCF).

The following is a list of a few of the functions you can perform on your designs in the FPGA Editor.

- Place and route critical components before running the automatic place and route tools.

- Finish placement and routing if the routing program does not completely route your design.

- Add probes to your design to examine the signal states of the targeted device. Probes are used to route the value of internal nets to an IOB (Input/Output Block) for analysis during the debugging of a device.

- Run the BitGen program and download the resulting bitstream file to the targeted device.

- View and change the nets connected to the capture units of an Integrated Logic Analyzer (ILA) core in your design.

- Create an entire design by hand (advanced users).

Design Flow

The following steps are an overview of the FPGA Editor design flow.

1. Save your design in EDIF (Electronic Data Interchange Format).

2. Run NGDBuild, which creates an NGD (Native Generic Database) file.

3. Run the MAP program, which creates an NCD (Native Circuit Description) file.

4. Load your design into the FPGA Editor, make necessary changes, and save the modified design as an NCD file.

5. Run the PAR (Place and Route) program on the modified NCD file.

The following figure shows the complete Xilinx design flow.
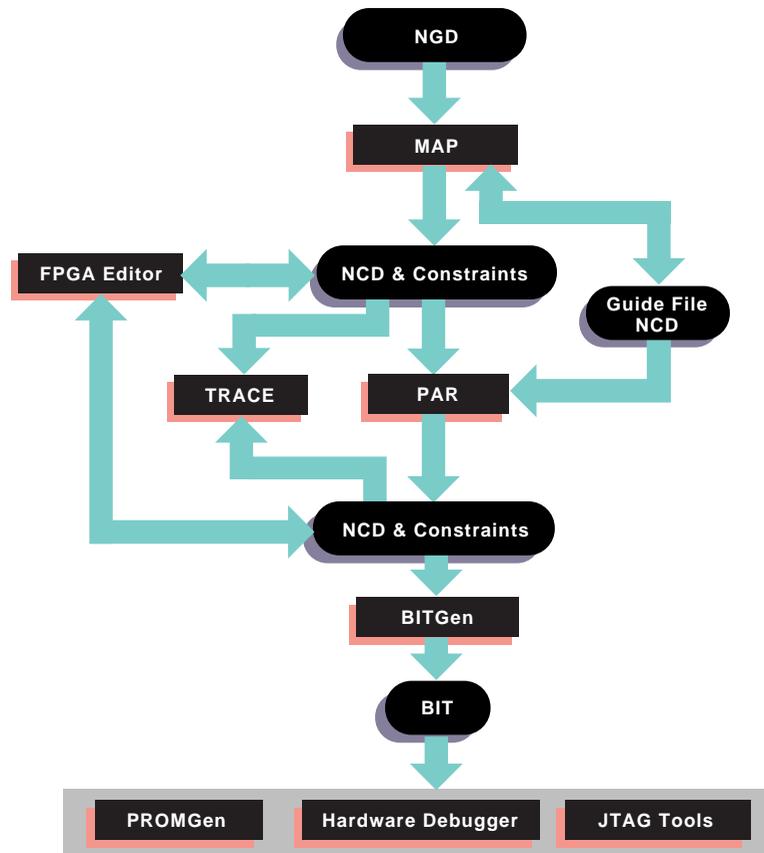
X7214

Figure 1-1 Design Flow

FPGA Editor Files

The following files are used with the FPGA Editor. See the "FPGA Editor Files" appendix for more information on these files.

NCD File

The NCD file is the output file from the MAP program and represents the physical design. This file describes a design in terms of CLBs

(Configurable Logic Blocks) and IOBs (Input/Output Blocks). You can edit a design in the FPGA Editor, save it, and then route it with PAR (Place and Route).

PCF File

The PCF file is an ASCII file containing physical constraints created by the MAP program, as well as physical constraints entered by you in your design and in the UCF (User Constraints File). You can edit the PCF file in the FPGA Editor.

NMC File

The NMC file is a physical macro library file. This file contains a physical macro definition that can be instantiated into an FPGA Editor design.

Architecture Support

The software supports the following architecture families in this release.

- Spartan™/XL/-II

- Virtex™/-E/-II

- XC4000™E/L/EX/XL/XV/XLA

- XC3000™A/L

- XC3100™A/L

- XC5200™

Chapter 2

Getting Started

This chapter describes how to start and exit the FPGA Editor. It also explains the basic elements and operations of the FPGA Editor interface. This chapter includes the following sections.

- "Starting the FPGA Editor"

- "Exiting the FPGA Editor"

- "Configuring Xprinter"

- "Using the Interface"

Starting the FPGA Editor

The FPGA Editor runs on PCs and workstations. You can start the FPGA Editor from the Windows Program Manager, the Xilinx Alliance Design Manager, the Xilinx Foundation Project Manager, or the command line.

This section contains the following topics.

- "From the Design Manager (Alliance Series)"

- "From the Project Manager (Foundation Series)"

- "Stand-Alone Tool (PC Only)"

- "From the Command Line"

From the Design Manager (Alliance Series)

To start the FPGA Editor from the Design Manager window (PC or workstation), select **Tools** → **FPGA Editor** or click on the FPGA Editor icon as shown in the following figure.

From the Project Manager (Foundation Series)

To start the FPGA Editor from the Project Manager window, select
Tools → **Implementation** → **FPGA Editor**.

Stand-Alone Tool (PC Only)

Note To run the FPGA Editor as a stand-alone tool on a PC, you must
have the XILINX environment variable defined and the path must
include $XILINX/bin/nt.

If you installed the FPGA Editor as a stand-alone tool on a PC, click
on the FPGA Editor icon (shown in the previous figure) on the
Windows desktop. Optionally, you can select the Run command from
the Windows Start button menu, and specify fpga_editor.exe in the
Run dialog box. See the following section for information on the
various options you can specify in the Run dialog box.

From the Command Line

Enter the following command to start the FPGA Editor from the
command line.

```
fpga_editor
```

Syntax

This section provides examples for starting the FPGA Editor from the
command line. The "Variables" and "Options" sections that follow
provide definitions of the command line variables and options.

To run the FPGA Editor on an existing design, use the following
command.

```
fpga_editor [-r | -e | -n] design_name.ncd
[pcf_file_name.pcf]
```

To run the FPGA Editor and create a new design, use the following
command.

```
fpga_editor -e design_name.ncd [pcf_file_name.pcf]
[arch device package speed]
```

To run the FPGA Editor on an existing physical macro, use the following command.

```
fpga_editor [-e|-r] -m macro_file_name
```

To run the FPGA Editor to create a new physical macro, use the following command.

```
fpga_editor -e -m macro_file_name [arch device package
speed]
```

Variables

This section describes the command line variables.

- Design_name.ncd

  The name of your new or existing design file. The .ncd extension is optional.

- Macro_file_name

  The name of your new or existing macro file.

- Pcf_file_name.pcf

  The name of the new or existing constraints file that you want applied to your design file. The .pcf extension is optional.

- Script_file_name.scr

  Specifies the name of a command file that includes command line arguments. The .scr extension is optional.

- Arch

  Specifies the architecture (product family) of your design (for example, Virtex).

  Note To see what devices, packages, and speeds are available, select **File** → **New** from within the FPGA Editor. Click **Select Part** to display the Part Selector dialog box. See Figure 3-2 of the "Using the FPGA Editor" chapter.

- Device

  Specifies a device within the selected architecture (for example, 4036 or 4028).

- Package

  Specifies a package within the selected architecture and device (for example, PC84 or PQ100).

  Note You can only enter a device and package from a part library that is installed on your system. For example, if you have not installed the Xilinx 4036EX series part library, you cannot create a design using the 4036EX device and the package.

- Speed

  Specifies the speed grade of the selected part. Allowable speeds are listed in the online Data Book at `http://support.xilinx.com/partinfo/databook.htm`.

Options

Note See the "Opening an Existing Design File" section of the "Using the FPGA Editor" chapter for more information on command options.

Table 2-1 Command Line Options

Option	Definition
–r	Read Only Mode—prevents overwriting your design. You can open and edit your design, but you cannot save changes to your original file. However, you can save your changes to a file with a different name.
–e	Read-Write Mode—allows changes to your design, including changes to the logical configurations.
–n	No Logic Changes Mode—allows placement and routing changes, but not changes to the logical configurations. For example, you cannot add or delete nets and components or reprogram programmable components. This mode ensures that the design database is consistent with the schematic or netlist used to create it.

Table 2-1 Command Line Options

Option	Definition
−p *script_file_name*	Use this option to execute the command line arguments in the specified file after your design or macro is opened or created.
−m *macro_file_name*	Use this option to edit or create a macro.

Exiting the FPGA Editor

This section describes how to exit the FPGA Editor.

1. To exit the FPGA Editor, perform one of the following.

   ♦ Select **File** → **Exit**.

   ♦ Select **Exit** from the User toolbar.

   ♦ Enter the following in the Command Line toolbar.

   exit

   The following dialog box appears.

   Note If you have not made any changes since you last saved the design, the FPGA Editor window closes without displaying this dialog box.
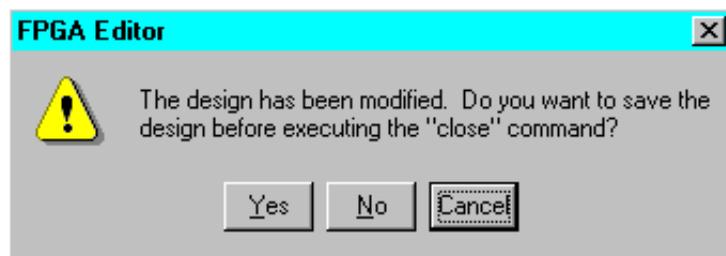
Figure 2-1 Exit Dialog Box

2. Click **Yes** to save changes and exit, **No** to discard changes and exit, or **Cancel** to return to your design without exiting.

Configuring Xprinter

Note This section applies only to workstation applications.

For detailed instructions on configuring a printer so you can print from the FPGA Editor, see the *Alliance Series 3.1i Quick Start Guide.*

Using the Interface

This section describes the FPGA Editor interface and how to use it.

Note Menus, dialog boxes, and parts of the application window are documented as they appear on a PC. Differences between the PC and the workstation applications are documented if there is a difference in operation between the two platforms.

This section contains the following topics.

- "Main Window"
- "Using Dialog Boxes"
- "Using Help"
- "Using the Mouse and Keyboard"

Main Window

This section describes the FPGA Editor main window. To work in a section of the interface, click the left mouse button in that section. The default layout of the main window is shown in the following figure. You can execute commands from the menus, toolbars, and command line. The client or work space area includes the Array window, List window, and World window.

Note If you do not have a design open in the Array window, an abbreviated menu bar is displayed.
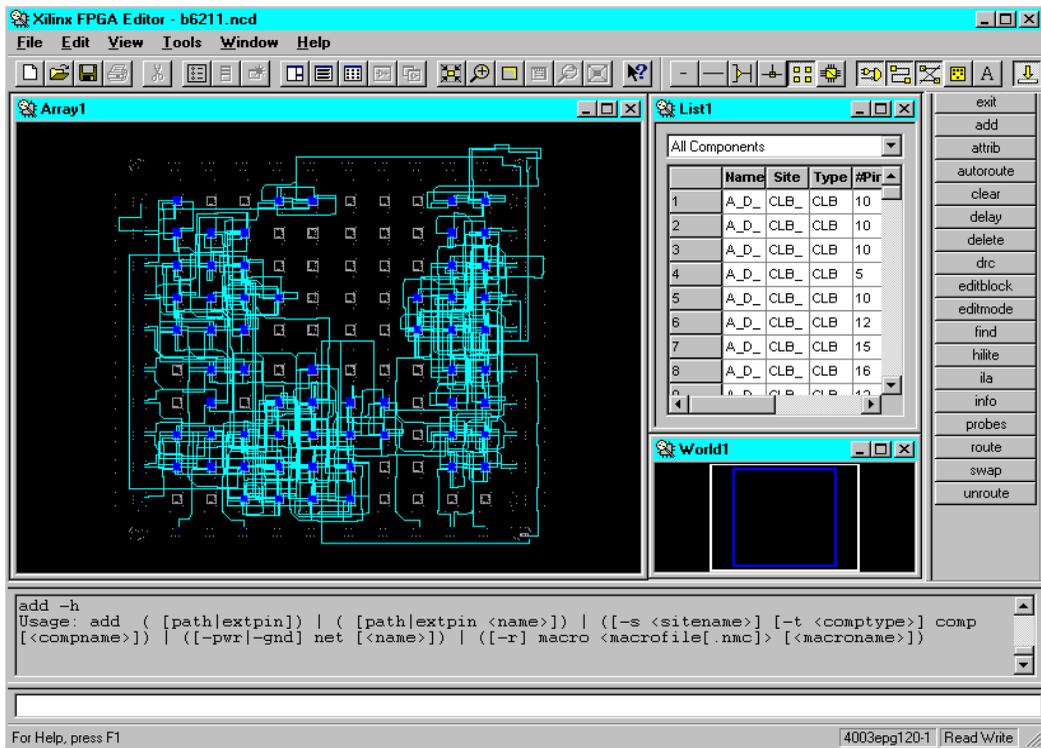
Figure 2-2 FPGA Editor Main Window

The main window includes the following areas.

Note Each window area is described in more detail in this section.

- Title Bar
- Menu Bar
- Standard Toolbar
- Layer Visibility Toolbar
- Array Window
- List Window
- User Toolbar
- World Window
- Window Tabs

Note The Window Tabs are displayed at the bottom of the client area when you select **View → Window Tabs**.

- History Toolbar

- Command Line Toolbar

- Status Bar

- Block Window

    Note The Block Window and its command toolbar are displayed when you double click the left mouse button on a logic block.

- Block Window Toolbar

Title Bar

The title bar at the top of the main window displays the program name (Xilinx FPGA Editor) and the name of the currently loaded design.

Menu Bar

The menu bar is located above the Array window. Most of the FPGA Editor commands are available in the pull-down menus of the FPGA Editor window after a design is loaded.

| File | Edit | View | Tools | Window | Help |
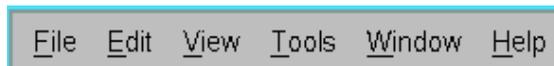
You can select menu commands with the mouse or the keyboard. With the mouse, click the left mouse button on the command. With the keyboard, press the Alt key and type in the letter underlined in the menu for that command. When you select a menu command with either method, a brief description of the command appears in the Status Bar at the bottom of the FPGA Editor window.

Some menu commands include an ellipsis (...). When you select one of these commands, a dialog box appears. You can enter additional information for that command in the dialog box. Refer to the FPGA Editor online help for detailed information on the commands in each menu. To access the online help within the FPGA Editor, select **Help → Help Topics**.

Standard Toolbar

The Standard toolbar contains commands for performing common operations on your designs, such as opening a design file, deleting objects from a design, and displaying a more detailed view of a design. Click once on a button in the toolbar to access a command. When you position the mouse pointer over a toolbar button, a short description (a tool tip) appears next to the button and a longer description appears in the status bar at the bottom of the main window. See the FPGA Editor online help for more information on the Standard toolbar.

Layer Visibility Toolbar

The Layer Visibility toolbar allows you to specify which objects are displayed in the Array window. When you position the mouse pointer over a toolbar button, a short description (a tool tip) appears next to the button and a longer description appears in the status bar at the bottom of the main window. See the FPGA Editor online help for more information on the Layer Visibility toolbar.

Use the left mouse button to select or deselect the layers you want displayed or hidden. You can control the display of the following objects.

Note See the "Glossary" for definitions of these objects.

- Local lines
- Long lines
- Pin wires
- Pips
- Sites
- Switch boxes
- Components

- Routes

- Ratsnests

- Macros

- Text (reference designators)

- For Virtex and Virtex2 designs, double and hex lines can be displayed

Array Window

The Array window displays a graphical representation of the FPGA device. The following figure shows a section of the Array window.

Figure 2-3 Section of Array Window

When you position the mouse pointer over an object in the Array window, a description or data tip appears in a small floating window. This functionality is similar to using the pick –q command (see the FPGA Editor online help for more information on the Pick command). The following is an example of a data tip for a site.

site "CLB_R9C12.S1", type = SLICE.

To turn off the data tips, enter the following in the Command Line toolbar.

```
setattr main array-datatips off
```

Note See the FPGA Editor online help for a complete description of the Setattr command.

The device components and the interconnections (both logical and routed) between these components are displayed in this window. When you edit the internal logic of a programmable component such as a logic block, a schematic of the interior of the component is displayed in the Block window (see the"Block Window" section for more information).You can control the display of objects in the Array window with the Layer Visibility toolbar. You can turn off the display of individual object layers, such as switches, wires, and routed connections, to make your design easier to view. Selectively viewing objects also reduces the duration of each screen redraw.

Note Symbols used in the FPGA Editor Array window for specific devices are described in *The Programmable Logic Data Book*.

A site is a programmable logic element (used or unused) located within the device. Sites are potential locations for components and are displayed in the Array window as outlines of components. Components are CLBs, IOBs, tristate buffers, pull-up resistors, oscillators, or clocks. When you place a component in a site, the outline is filled in as shown in the following figure.

X8971
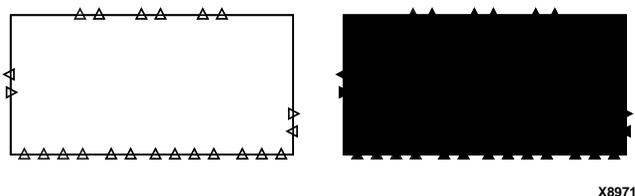
Figure 2-4 Sites and Components

Components are connected by nets. A net is a set of component pins that are electrically connected in the finished design. When a component is placed, but not yet routed, the connection between its net pins (those pins on the net) and net pins from other components is logical, not physical. The pins are associated with each other, even though there is no electrical connection between them. In the Array window, the logical connections that make up a net are displayed as ratsnest

lines or direct point-to-point connections between net pins, as shown in the following figure.
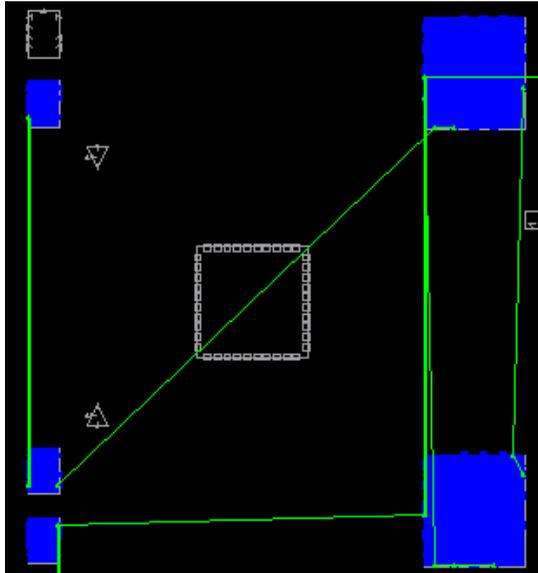
Figure 2-5 Ratsnest Display (Unrouted Net)

When the net is routed, electrical connections are made between the net pins. In the Array window, the routed connections appear as lines following the routing resources available on the device (long lines, pinwires, switch boxes, and so on). The following figure shows a routed net.
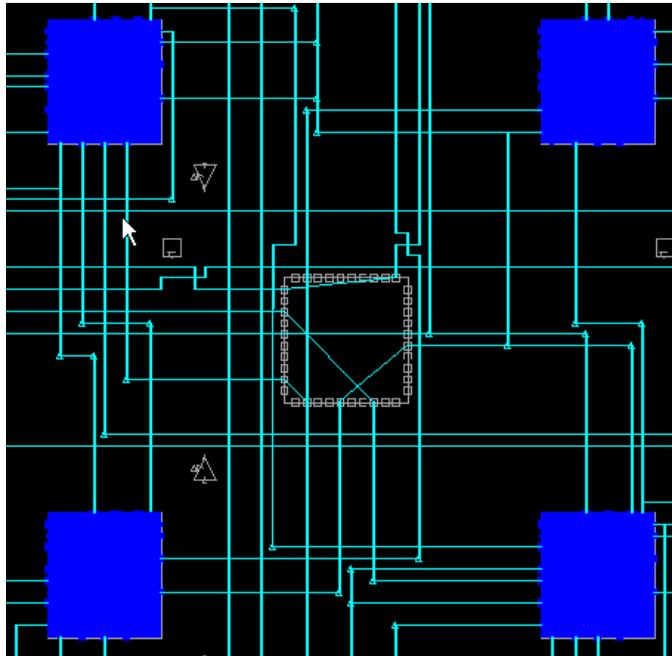
Figure 2-6 Display of Routed Net

List Window

The List window displays a list of the components, nets, layers, paths, macros, constraints, and bels in your design. Use the pull-down list box at the top of the window to specify the items you want displayed in the List window. You can use the Maximize List Window toolbar button to maximize this window. To print the List window, first select the window with the mouse and then use the Print command in the File menu. To save the List window data to a Comma Separated Value (.csv) file, use the Export command in the File menu (see the FPGA Editor online help for more information on the Export command).

Figure 2-7 List Window

Note See the "Selecting Objects with the List Window" section of the "Using the FPGA Editor" chapter for more information on the List window.

User Toolbar

The User toolbar, shown in the following figure, provides a convenient way to perform frequently used commands. To use a command, select the appropriate command button with the left mouse button.

exit
add
attrib
autoroute
clear
delay
delete
drc
editblock
editmode
find
hilite
ila
info
probes
route
swap
unroute

Figure 2-8 User Toolbar

You can customize the User toolbar with the Button and Unbutton commands described in the FPGA Editor online help. Alternatively, you can edit the fpga_editor.ini or the fpga_editor_user.ini file (see "Customizing the FPGA Editor" chapter for more information on these files). These files define the default User toolbar buttons that appear when the FPGA Editor window opens. You can define up to fifty buttons; however, the height of the toolbar may limit the number of buttons that are visible.

Table 2-2 User Toolbar Command Summary

Command	Description
exit	Exits FPGA Editor. If you edited your design, a dialog box appears to allow you to save changes.
add	Adds selected sites as components to your design.

Table 2-2 User Toolbar Command Summary

Command	Description
attrib	Posts a dialog box containing property information for all selected items or for the main window if no item is selected.
autoroute	Automatically routes a selected component, macro, net, ratsnest, or net pin between selected routing resources.
clear	Deselects all selected objects.
delay	Calculates and lists the path delay for a selected net or selected pins on a net. This information appears in the History toolbar and in the List window when nets are selected.
delete	Deletes selected objects from the design.
drc	Performs a DRC (Design Rule Check), which checks for logical and physical errors in the design. The results appear in the History toolbar and also in the log file.
editblock	Opens the Block window, which allows you to view or edit the internal logic of a selected logic block.
editmode	Toggles between Read Only, No Logic Changes, or Read Write edit modes.
find	Posts a dialog box that allows you to find a specified component, macro, site, pin, or net.
hilite	Highlights a selected object.
ila	Posts a dialog box that allows you to view and change the nets connected to the capture units of the ILA core in your design.
info	Lists the properties for a selected object in the History toolbar.
probes	Posts a dialog box that allows you to add probes to your design.
route	Routes selected objects (manual route).

Table 2-2 User Toolbar Command Summary

Command	Description
swap	Moves a component to another site or swaps locations between two components or two logic block pins.
unroute	Unroutes selected objects.

World Window

The World window, as shown in the following figure, shows the area of the device that is currently displayed in the Array window.
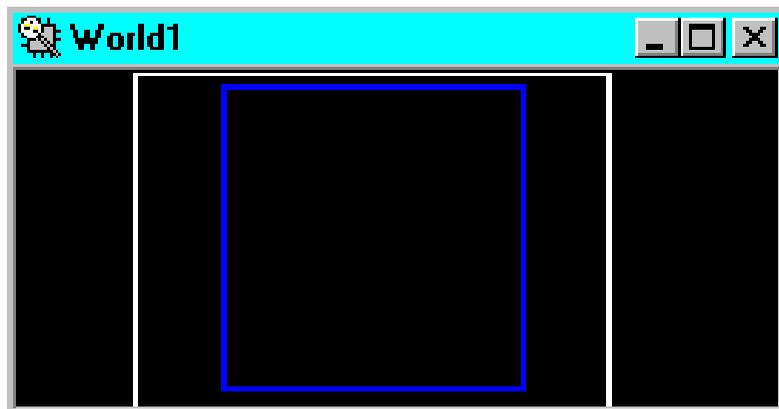
Figure 2-9 World Window

As you pan and zoom the Array window, notice the corresponding changes in the size and position of the white rectangle within the World window. Also, any objects selected in the Array window appear in the World window. You can drag the white rectangle with the mouse button to pan the display to the desired position. If you have multiple Array windows, the World window displays a rect-angle for each Array window.

Window Tabs

The Window Tabs, as shown in the following figure, are displayed at the bottom of the client area when you select **View → Window Tabs**. These tabs display the windows that are open even if they are mini-mized.

Figure 2-10 Window Tabs

History Toolbar

The History toolbar, shown in the following figure, is located below the Array window and displays commands and responses. All error messages, warnings, and command responses are written to the History toolbar. Information in the History toolbar is especially useful for deciphering unexpected command results.

Use the scroll bar located to the right of the History toolbar to view all the commands and responses recorded during an editing session. You can resize this area by selecting the top of the toolbar, and using the left mouse button to move the double-headed arrow that appears up or down.

The History toolbar accommodates 32,000 characters of output on Windows 98® platforms, and a much larger number of characters on Windows NT® and UNIX® platforms. To examine text beyond this limit, use any text editor to view the contents of the FPGA Editor log file while you are editing your design. The log file is named *design_name*_fpga_editor.out and is described in the "Recovering a Terminated Session" section of the "Using the FPGA Editor" chapter. You must remain in the FPGA Editor while you view this file because the file is renamed when you exit the program. The .out file is

renamed *design_name*_fpga_editor_YYMMDD_HHMMSS.log, where Y is year, M is month, D is day, H is hour, M is minute, and S is second.

Command Line Toolbar

Use the Command Line toolbar (located directly below the History toolbar) to enter commands from the keyboard.

The following is a list of tips for using the Command Line toolbar.

- To enter multiple commands on the same line, use semicolons to separate each command.

- Use a backslash at the end of the line to continue a command on the next line.

- Commands are case-sensitive.

- For command arguments that contain special characters (" ' * ? \ ; #), embedded spaces, or leading dashes, you must place the argument in quotation marks (for example, button "zoom in twice" "zoom in; zoom in"). Alternatively, you can precede the reserved character with a back slash (for example, select net \ ;* selects a net with the name ;*).

  On PC's, since the back slash serves as the FPGA Editor escape character, it affects DOS™-style path names. For example, the following command will not produce the desired effect.

  ```
  load design \data\mydesign.ncd
  ```

  The system responds with the following.

  ```
  ERROR – load: file "datamydesign.ncd" not found
  ```

  Use one of these solutions to solve this problem.

  - Use forward slashes (UNIX style) in the path name as in the following example.

    ```
    load design /data/mydesign.ncd
    ```

  - Escape the escape characters as in the following example.

    ```
    load design \\data\\mydesign.ncd
    ```

- If you have trouble entering text, make sure the keyboard focus is in the Command Line toolbar. Press the F2 key to move the keyboard focus to the Command Line toolbar.

- Use the up and down arrow keys to display the values of previous and next commands.

- Use the following keystrokes to move the cursor in the Command Line toolbar.

Table 2-3 Command Line Toolbar Keystrokes

Keystroke	Description
Left arrow (←) or Control–b	Move back one character
Right arrow (→) or Control–f	Move forward one character
Home or Control–a	Move to the beginning of the line
End or Control–e	Move to the end of the line
Backspace	Delete the character to the left of the cursor
Control–x	Cut the selected characters to the clipboard
Control–v	Paste the clipboard contents to the right of the cursor or to replace selected characters
Control–c	Copy selected characters to the clipboard

Status Bar

The status bar appears at the bottom of the main window (below the Command Line toolbar). When you select a menu command, a brief description of its function appears in the status bar.

Block Window

The Block window is used to edit logic blocks. This window is displayed when you double click the left mouse button on a logic block. You can use only one Block window at a time for editing; however, you can have additional Block windows open for viewing. See the "Editing Component Logic" section of the "Using the FPGA Editor" chapter for more information. The Block window is shown in the following figure.
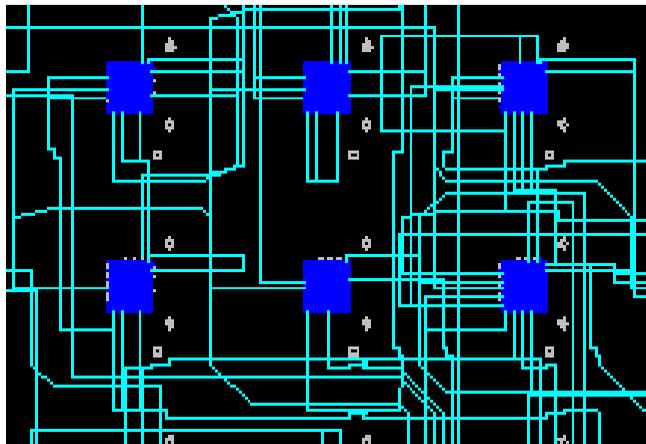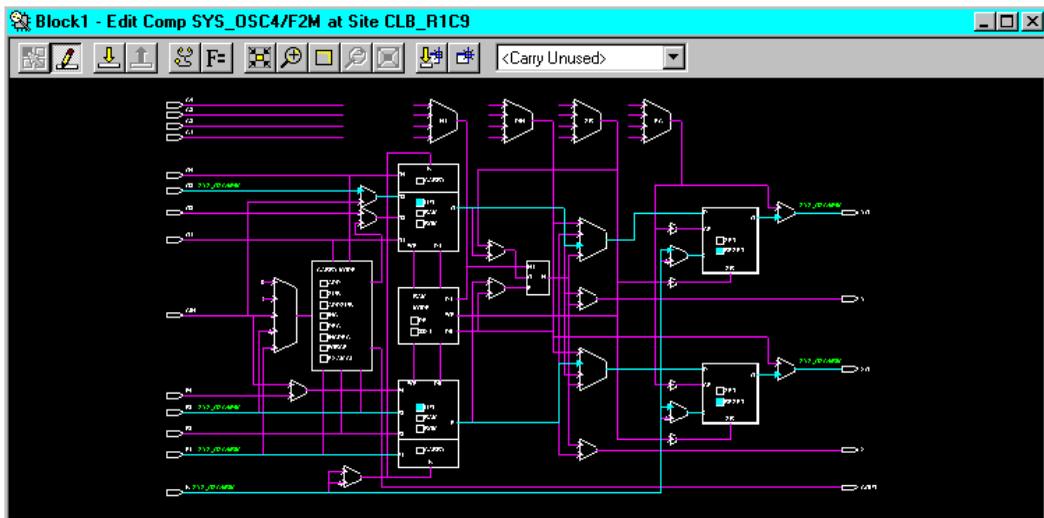
Figure 2-11 Block Window

When you position the mouse pointer over an object in the Block window, a description or data tip appears in a small floating window. To turn off the data tips, enter the following in the Command Line toolbar:

```
setattr main block-datatips off
```

Note See the FPGA Editor online help for a complete description of the Setattr command.

Using Dialog Boxes

Many menu commands display dialog boxes in which you can enter information and set options.

Using Common Fields

The fields shown in the following table are common to most dialog boxes.

Table 2-4 Common Dialog Box Fields

Dialog Box Field	Function
OK	Closes the dialog box and implements the intended action according to the settings in the dialog box
Apply	Implements the intended action without closing the dialog box
Cancel	Closes the dialog box without effecting any action
Help	Displays information on the dialog box

Browse Buttons

Many dialog boxes contain browse buttons to allow you to navigate through your directory structure to find a particular file or to save a file to a specific location.

Moving Items within List Boxes

Many of the dialog boxes feature list boxes. You can select and move items from one list box to another using either the mouse or the keyboard, or a combination of these methods.

Use the mouse to select items in list boxes as follows.

- To move an item to a list box on the right, select it with the left mouse button. Click the `>` button. To move an item to a list box on the left, select it, then click the `<` button.

- To move all items to a list box on the right, click the `>>` button. To move all items to a list box on the left, click the `<<` button.

- To move sequential items, select the first one with the left mouse button. Hold down the `shift` key. Select the last item and release the `shift` key. Click the `>` or `<` button to move the selected items to the right or left list box, respectively.

You can also select the first item; drag the mouse up or down until all the desired items are highlighted; and then click the `>` or `<` button.

- To move multiple items in any order, hold down the `Ctrl` key while clicking individual items. When you finish selecting, release the `Ctrl` key and click the `>` or `<` button.

Use the keyboard to select items in list boxes as follows.

- To move an item in a list box, press the `Tab` key until the first item in the list box is highlighted. Press the down arrow key to select the desired item. Tab to the `>` or `<` button to move it to the right or left list box, respectively. Press `Enter`.

- To move all items to a list box on the right, tab to the `>>` button and press `Enter`. To move all items to a list box on the left, tab to the `<<` button and press `Enter`.

- To move consecutive items, tab to the first item in the list box. Use the down arrow key to highlight the first desired item. Press and hold the `Shift` key while using the down arrow key to select the other items in the sequence. Tab to the `>` button. Press `Enter`.

- *On workstations only:* To move multiple items in any order, tab to the list box and press `Shift F8`. Use the up and down arrow keys to navigate within the list box. Press the space bar to select each item. Tab to the `>` button. Press the `Enter` key.

Using Filters with Commands

Many menu commands have dialog boxes that allow you to filter a list of choices, that is, display a subset of the listed items. These dialog boxes contain a filter field, Apply buttons, and Clear or Reset buttons.

In filter fields, you can enter a text string consisting of characters and wildcards. You cannot enter a range of items in filter fields.

- Characters can be any alphanumeric characters, text spaces, and the characters that appear on the top of the number keys on a keyboard. Alphabetic characters are case-sensitive. Control characters are not permitted.

- A wildcard can be an asterisk (*), which can represent any number of characters, or a question mark (?), which represents a single character.

The software does not strictly match patterns; it matches entire text strings. It does not find a string if it is embedded in a larger string unless you use wildcards. For example, it does not find $1N36 if it is embedded in ABC$1N36XYZ. However, if you searched for *$1N36*, it would find that string in ABC$1N36XYZ.

To use the Filter dialog boxes, follow these steps.

1. Specify the pattern to include in your list by typing the characters.

2. Include one or more wildcards (*) to do a global search on the specified string.

    Precede the character string with a wildcard to retrieve all signal names that end with the string of specified characters.

    Append the wildcard to the character string to retrieve all signal names that start with the specified character string.

3. Click **Apply**.

    The list displays only the choices that match the selection criteria.

4. To clear the filter, click **Clear** or **Reset** or backspace over the information specified in the filter text box.

Using Help

You can obtain help on commands and procedures with the Help menu or by selecting the Help toolbar button. In addition, the dialog boxes associated with many commands have a Help button that you can click to obtain context-sensitive help.

Note The online help provides detailed information on menu commands, dialog boxes, and command line commands.

Help Menu

Use the following Help menu commands to obtain help.

•   The Help Topics command opens Help and lists the online help topics available. From the Contents page, you can jump to command information or step-by-step instructions. After you open help, you can click the Help Topics button in the Help window whenever you want to return to the help topic list.

- The Online Documentation command opens the software manuals in the default Web browser.

- The About FPGA Editor command opens a popup window that displays the version number of the tool and a copyright notice.

Toolbar Help Button

You can obtain context-sensitive help from the toolbar as follows.

1. Click the Help button in the toolbar.

The cursor changes to a question mark.

2. With the left mouse button, click the menu item or toolbar button for which you want help.

   Help appears for the selected command or option.

Note You can also press `Shift F1` to obtain context-sensitive help.

F1 Key

Pressing the F1 key on a dialog box displays help on that dialog box. Pressing the F1 key is the same as selecting Help Topics from the Help menu, if no dialog boxes are displayed.

Help Button in Dialog Boxes

Many of the dialog boxes have a Help button that you can click to obtain help for the dialog box with which you are working. You can also press `Alt H` on your keyboard while positioned over the dialog box to obtain help.

Using the Mouse and Keyboard

Use the mouse and keyboard to perform the following operations in the FPGA Editor.

Table 2-5 Using the Mouse and Keyboard

Mouse Action	Function
Left Click	To select toolbar buttons, menus, menu commands, dialog box options, design objects
Right Click	Display a popup menu at the top of the columns in the List window
Control + Right Click	To zoom in
Shift + Control + Right Click	To zoom out
Control + Right Drag	To pan
Right Drag	To move a placed component to an unused site
Left Drag	To move an unplaced component from the List window to an unused site
Middle Drag	To draw a rectangle around an area to zoom in on (available in Array and Block windows)

You can edit the fpga_editor.ini file to change the mouse button settings and customize them to perform operations other than those shown. For example, if you do not have a middle button on your mouse, you can change the settings in this file. The procedure for customizing mouse buttons is described in the "Customizing the FPGA Editor" chapter.

Zooming with the Mouse

To zoom in or out, click the appropriate button on your mouse; see Table 2-5. The Array window zooms by a preset factor; the number of available zoom levels is five by default.

Note See the fpga_editor.ini file for information on using the Setattr command to change the zoom settings for viewing dense designs. This file is described in the "Customizing the FPGA Editor" chapter.

Zoom Toggle

The zoom toggle feature allows you to toggle from an inner zoom level to the outermost zoom level and back. This feature is useful when you want to remain at a zoom level that provides good visibility for editing. For example, to move to a location outside your current editing area, but maintain the selected zoom level, use the zoom toggle to go to the outer zoom level, move the cursor to the desired location, and zoom back in.

To perform a zoom toggle, place the cursor in the Array window and type the letter "z" or "Z." If you begin at a zoomed-in level, you zoom out to the maximum zoom level. When you toggle again, you return to the previous zoom level.

Note The default letter "z" is defined in the fpga_initialization.ini file. You can edit this file and customize it for your use.

Panning

Panning keystrokes are defined in the FPGA Editor initialization (fpga_editor.ini) file and are summarized in Table 2-6. You can also use the World window to pan to a selected area. This panning method is described in the "World Window" section.

To pan, you must zoom in at least one level and position the mouse cursor in the Array window. You can perform the following pan operations.

- For smooth panning, hold down the right mouse button and drag the mouse. The window pans in the direction of the mouse movement. Panning continues until you reach the edge of the window or until you release the mouse button.

- When you reach the edge of the window with the cursor, the FPGA Editor will autopan the window until the edge of the FPGA device is reached. If you are near the edge of the window, the autopanning is slower than when you are far from the edge of the window.

- Use the keyboard arrow keys to pan left, right, up, or down.

- Use the Shift key with one of the arrow keys to pan the window to its furthest boundary in the direction of the arrow key.

- Press the space bar to center the display around the cursor position.

Table 2-6 Panning Keystrokes

Keystrokes	Result
Up arrow ($\uparrow$)	Pan up 75% of one window height.
Shift + $\uparrow$	Pan to upper edge of device.
Down arrow ($\downarrow$)	Pan down 75% of one window height.
Shift + $\downarrow$	Pan to lower edge of device.
Left arrow ($\leftarrow$)	Pan left 75% of one window height.
Shift + $\leftarrow$	Pan to left edge of device.
Right arrow ($\rightarrow$)	Pan right 75% of one window height.
Shift + $\rightarrow$	Pan to right edge of device.
Space bar	Center display around current cursor position.

Chapter 3

Using the FPGA Editor

This chapter includes information on performing various operations on your designs within the FPGA Editor. It includes the following sections.

Note You can select commands from the pull-down menus or the User toolbar, or you can enter commands in the Command Line toolbar. For more detailed information on Menu and Command Line commands, see the FGPA Editor online help.

- "Creating a New Design File"

- "Opening an Existing Design File"

- "Saving a Design File"

- "Recovering a Terminated Session"

- "Selecting Objects"

- "Adding Objects"

- "Deleting Objects"

- "Highlighting Objects"

- "Viewing and Changing Properties"

- "Moving and Swapping Components and Macros"

- "Editing Component Logic"

- "Placing and Unplacing Components"

- "Routing and Unrouting"

- "Constraints"

- "Using the ILA Command"

- "Using Probes"
- "Verifying Your Design"

Creating a New Design File

Note You must close your current design or macro before using the New command to create a new design.

To create a new design in the FPGA Editor, follow these steps.

1. Select **File → New** or click on the New toolbar button.
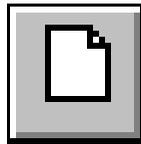
The New dialog box appears as shown in the following figure.
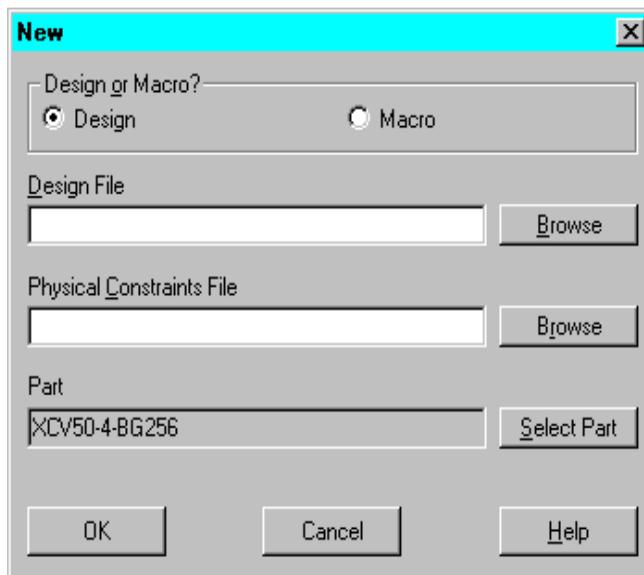
Figure 3-1 New Dialog Box

2. Select the **Design** option.

3. Enter the design name in the Design File field. If you do not enter the .ncd extension, it is automatically added when the file is created.

   You can open a new design under a directory other than the current working directory. Type in the path name of the target directory in the Design File field, or click **Browse** to select your target directory and specify your design file name.

4. In the New dialog box, enter the constraints file name in the Physical Constraints File field, or click **Browse** to specify the directory and file name. If you do not enter the .pcf extension, it is automatically added to your file name. If you do not enter a file name, a constraints file is created with the same name as the .ncd file but with a.pcf extension.

5. Click **Select Part** to display the Part Selector dialog box.
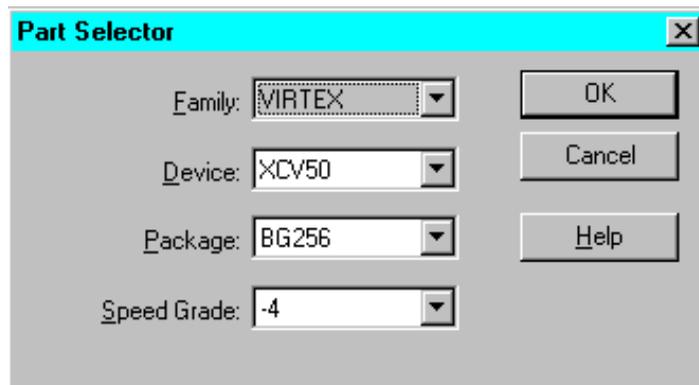
Figure 3-2 Part Selector Dialog Box

6. Select a Family, Device, Package, and Speed Grade for your design.

   Note You can only select a part number from a part library you have installed on your system.

7. Click **OK**. The Part field in the New dialog box is filled in with your selections.

8. Click **OK** to close the New dialog box.

   An empty (unprogrammed) design is loaded into the FPGA Editor window with the part number and speed as specified.

Opening an Existing Design File

Note You must close your current design or macro before using the Open command to open another design.

To open an existing design file from within the FPGA Editor, follow these steps.

1. Select **File → Open** or click on the Open toolbar button.

The Open dialog box is displayed.
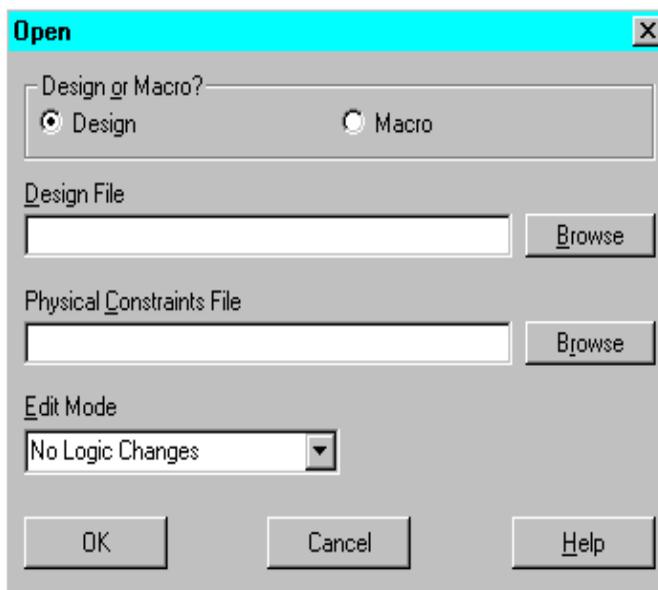
Figure 3-3 Open Dialog Box

2. Select the **Design** option.

3. Enter the name of the .ncd file you want to open in the Design File field, or click **Browse** to specify the design file name and directory.

4. Optionally, enter a constraints file name in the Physical Constraints File field, or click **Browse** to specify the constraints file name and directory.

   If you do not enter a file name in the Physical Constraints File field, the FPGA Editor looks for a constraints file with the same name as the .ncd file but with a.pcf extension.

5. Select an Edit Mode for the design from the pull-down list box.

   Select Read Only, No Logic Changes, or Read Write. To always edit an existing design file in Read Write Mode, enter the following command in the fpga_editor.ini or fpga_editor_user.ini initialization files.

   ```
   setattr main edit-mode read-write
   ```

6. Click **OK**.

   The specified design file is displayed in the Array window.

Saving a Design File

You can save your current design at any time during an FPGA Editor session in one of these variations.

- Save design under the same name

- Save design with a different name

- Save design as a macro library file

   Note The first two methods are described following. The procedure for saving a design as a macro library file is described in the "Working with Physical Macros" chapter.

With the Same Name

To save your current design, select **File** → **Save**. The current design file is written to disk.

With a Different Name

Use the following procedure to save your current design with a different name.

1. Select **File** → **Save As** to display the Save As dialog box.

Save As

Design or Macro?

● Design   ○ Macro

Design File

[   ]   Browse

Physical Constraints File

[   ]   Browse

OK   Cancel   Help
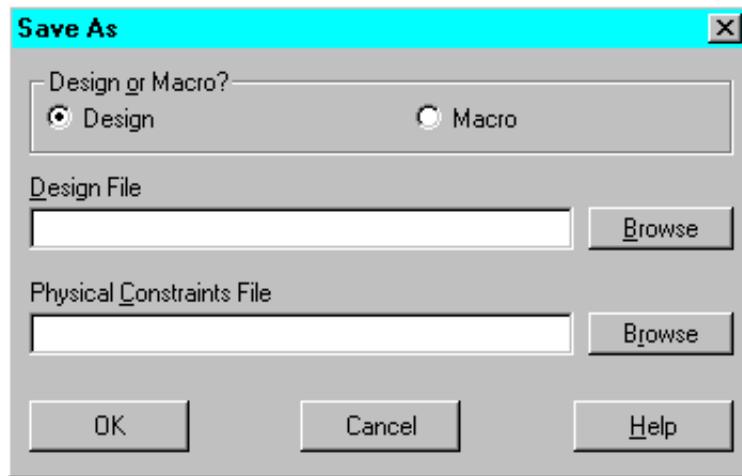
Figure 3-4 Save As Dialog Box

2.   Select the **Design** option.

3.   Enter the design name in the Design File field. If you do not enter the .ncd extension, it is automatically added to the file name.

You can save your design under a directory other than the current working directory. Type in the path name of the target directory in the Design File field, or click **Browse** to select your target directory and specify your design file name.

4.   If applicable, enter the constraints file name for the design in the Physical Constraints File field, or click **Browse** to select a directory and specify a file name.

If you do not enter a file name in the Constraints File field, a constraint file is saved with the same name as the .ncd file but with a.pcf extension. A constraints file is created only if there are existing constraints associated with your design.

5.   Click **OK**.

The dialog box closes, and the design file and constraints files are saved with the names you specified. The file name in the Title Bar at the top of the FPGA Editor window changes to indicate that this new file is currently displayed in the window. When you later save or exit the file, it is saved to the new file name, not the original file name.

Recovering a Terminated Session

During an editing session, the system keeps track of all the commands used on your design. The commands are written into a temporary command log file, *design_name*_fpga_editor.out, in the same directory where your design is located. This file is updated after a certain number of commands are performed. You can set the rate at which commands are written to the log file with the Setattr command and the flush rate attribute (see the FPGA Editor online help for information on this command). The default is to update the file after ten commands have been performed. Each time you save a design file, the commands are erased from *design_name*_fpga_editor.out.

When you end an FPGA Editor session, the .out file is renamed *design_name*_fpga_editor_YYMMDD_HHMMSS.log, where Y is year, M is month, D is day, H is hour, M is minute, and S is second.

If an editing session is unexpectedly terminated (for example, if you lose power or a software error causes the session to end prematurely), the *design_name*_fpga_editor.out file remains in your current design directory. The next time you start the FPGA Editor, the *design_name*_fpga_editor.out file is renamed *design_name*_fpga_editor.rcv. A new *design_name*_fpga_editor.out file is created for the current FPGA Editor session. A message box appears in the Array window with a message similar to the following.

```
A crash recovery file was found for this design
   indicating that fpga_editor previously
   terminated abnormally. The changes that you made
   can be recovered. (This process will overwrite
   your current design file).
Do you want to run recovery?
```

If you want to run recovery, click **Yes**. Click **No** if you do not want to run the recovery process. The recovery process executes the commands in the *design_name*_fpga_editor.rcv file, restoring the session up to the point where it was terminated. If you do not run the recovery process, the *design_name*.rcv file is ignored and the commands in the file are not applied to the design file.

The recovery file records the number of valid commands in order to prevent repeating the command that caused the abnormal termina-

tion of the FPGA Editor. However, some commands issued before the command that caused the crash may not be repeated.

Selecting Objects

You can use the following methods to select an object in the Array window.

- Move the mouse over an object in the Array window, and click the left mouse button. This method is usually the easiest. See the "Selecting Objects with the Mouse and Keyboard" section for more information.

- Display an object list in the List window, and click the left mouse button on an object. See the "Selecting Objects with the List Window" section for more information.

- Type the Select command in the Command Line toolbar along with the appropriate arguments. See the FPGA Editor online help for a description of the Select command.

- Use **Edit** → **Find**. See the "Selecting Objects with the Find Command" section for details.

In some cases, you can only select from a list or from the command line because there is no graphical representation of the object in the Array window. You can select the following objects in the Array window.

- Bels

- Sites

- Site pins

- Components

- Component pins

- Nets

- Paths

- Macros

- Switch box pins

- Wires (pinwires, long lines, local lines)

- Route lines

- Object layers

- Constraints

- Ratsnest lines

When you select an object, it changes color and its name is displayed in the History toolbar. When you select a used wire (pinwire, long line or local line), the name of the net using the wire is displayed in the History toolbar. When you select a used component pin (or net pin), the History toolbar shows the pin name, the name of the net to which this pin belongs, and the delay time from the net source to this pin.

When you select additional objects, they also change color. You can select multiple objects of the same type and multiple types of objects. However, the objects you select must be appropriate for the action you are performing.

You can control whether design objects remain selected or are deselected after a command is performed with the Automatic Deselect option in the Main Properties property sheet.

After you select the appropriate object or objects, initiate the command with any of the methods available in the FPGA Editor (such as menu, User toolbar, command alias, and so on).

Notes On Selecting Objects

This section includes some additional information on selecting objects in your design.

- You cannot select unplaced components, paths, or macros directly in the Array window. However, you can select them using the List window and the Select command.

- You can select object layers, such as the components layer or the ratsnest layer. Layers are selected in order to change attributes for all objects in the layer, for example, to display all layer components in green.

- When you select a component, net, or macro name in the List window, it is highlighted and the corresponding object changes color in the Array window. When you select a component, net, or macro in the Array window, it changes color and its name appears selected in the List window. Selecting a constraint in the

List window does not cause a corresponding highlight in the Array window. The only time you may want to select a constraint in the List window is when you want to delete it.

- Switch box pins are selectable in the Array window. When you select a switch box pin, the allowable connections from the selected pin to other pins within the switch box are displayed in yellow. Bi-directional pins are solid and uni-directional pins are hollow. Downhill paths are displayed as bright yellow and uphill paths are displayed as dull yellow.

- Bring up the Block window for a component by double clicking the left mouse button on the component.

- When you select a ratsnest line, the entire net containing the ratsnest line is highlighted.

- Select a portion of a routed net by pressing the Control+Shift key while selecting the portion of the net with the mouse.

- For architectures that allow route-throughs, highlight all available route-through paths for a site or component pin by pressing the Control+Shift key while selecting the pin.

This section includes the following topics.

- "Selecting Objects with the Mouse and Keyboard"

- "Selecting Objects with the List Window"

- "Selecting Objects with the Find Command"

- "Selecting Objects with the Select Command"

- "Deselecting Objects"

Selecting Objects with the Mouse and Keyboard

The following table lists mouse and keyboard combinations for selecting and deselecting objects in your design as defined in the fpga_editor.ini file. To customize these settings, see the "Customizing the FPGA Editor" chapter.

Table 3-1 Selecting Objects with Mouse/Keyboard

Mouse/Keyboard	Function
Single left mouse button click	Selects an object, and deselects all previously selected objects
Control+left mouse button click	Selects/deselects more than one object in your design; toggles the selection status of the object, selecting it if not selected, and deselecting it if selected
Shift+left mouse button click	Selects object; use to select more than one object in your design
Left mouse button click on empty space	Deselects all selected objects
Double click left mouse button on component	Opens Block window
Control+Shift+left mouse button click on route segment	Selects the net
Control+Shift+left mouse button click on site or component pin	Selects route-throughs

Selecting Objects with the List Window

You can use the List window, shown in Figure 2-7, to display a list of the bels, components, nets, paths, layers, constraints, or macros in your design. You can open multiple List windows simultaneously for your design by selecting **Window → New → List Window**.

Use the pull-down list box at the top of the window to specify which of the following design objects appear in the list.

- Routed and unrouted nets
- Placed and unplaced components
- Placed and unplaced macros
- Layers

- Paths

- Constraints

- Bels

You can select items in the list with the following mouse and keyboard combinations.

Mouse/Keyboard	Function
Single left mouse button click	Selects a single object in the list
Control+left mouse button click	Selects multiple objects in the list that are not in sequential order
Shift+left mouse button click	Selects multiple objects in the list in sequential order

When you select an item in the List window, you are selecting the corresponding object in your design with that name. The selected object in the Array window is highlighted. You can then perform the same operations on the selected object that you can perform when you select the object in the Array window.

Customizing the List Window

Use the vertical and horizontal scroll bars if necessary to view the complete list of objects. You can also resize the window by positioning the cursor on any corner of the window. Press the left mouse button and drag the corner to the appropriate size. You can maximize this window with the Maximize List Window toolbar button. To print the List window, first select the window with the mouse and then use the Print command in the File menu. To save the List window data to a Comma Separated Value (.csv) file, use the Export command in the File menu (see the FPGA Editor online help for more information on the Export command).

To view the full name of an object in the list, place the mouse pointer over the name. The full name appears in a popup box over the name. Optionally, you can use the left mouse button to stretch or shrink the right side of any of the column headers to resize the column width.

You can sort the entries in the list by clicking on any of the column headers in the list. For example, click the left mouse button on the Name column and the list is sorted alphabetically. If you have All Components listed, and click the left mouse button on the Type column, the components are listed alphabetically by type. Clicking on

any of the column headers sorts the list based on what is displayed in that column.

Click the right mouse button on any of the column names in the List window to display a popup menu with the following choices.

Table 3-2 List Window Popup Menu Commands

Popup Menu Command	Function
Sort Ascending	Sorts the list in ascending order either alphabetically or numerically
Sort Descending	Sorts the list in descending order either alphabetically or numerically
Move Column Left	Moves the selected column to the left
Move Column Right	Moves the selected column to the right
Save Column Layout	Saves the current column layout
Restore Column Layout	Restores column layout to the layout used at the beginning of your editing session, or to the last saved layout
Default Column Layout	Restores the column layout to the Xilinx-defined layout

Selecting Objects with the Find Command

You can also select objects in your design with **Edit** → **Find**. This command displays the Find dialog box (shown in the following figure). Use the pull-down list to select the type of object you want to find in your design, and enter the name of the object in the Name field. If you are zoomed in when you select Find, the Auto Pan option automatically pans to the search object. The Auto Clear option clears all previous selections so only the item you are searching for is selected.
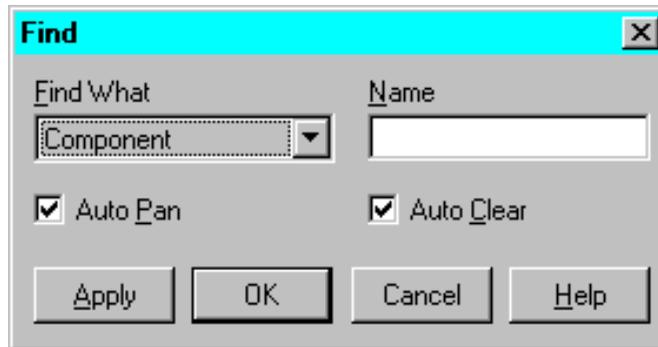
Figure 3-5 Find Dialog Box

As an alternative to using the Find dialog box, you can use the Select command with the Zoom Selection command in the Command Line toolbar. See the FPGA Editor online help for details on the Find, Select, and Zoom commands.

Selecting Objects with the Select Command

You can also select objects with the Select command in the Command Line toolbar. See the FPGA Editor online help for a description of the Select command.

Deselecting Objects

Use one of the following methods to deselect objects in your design.

- Select **Edit** → **Unselect All** to deselect all selected objects

- Click Clear in the User toolbar to deselect all selected objects

- Enter the Clear command in the Command Line toolbar to deselect all selected objects

   Use the Clear command before performing another command to deselect any selected objects in your design. You can also check which objects are selected by looking in the World window. This window displays selected objects even when they are not visible in the Array window.

Adding Objects

When you add objects to your design, these new objects become part of your design's database. Use the Add command to perform these functions.

- Create new components
- Create new nets
- Add a component pin to an existing net
- Add a macro to your design

You can add objects in any of the following ways.

- Select **Edit** → **Add**
- Select **Add** in the User toolbar
- Enter **add** in the Command Line toolbar

Note You cannot add objects if your design is in the No Logic Changes mode.

This section contains the following topics.

- "Adding Components"
- "Adding a Net"
- "Adding Component Pins to an Existing Net"

Adding Components

Use the following steps to add a component to your design.

1. In the Array window, select a vacant site for a new component.

2. Select **Edit** → **Add**.

   A component is created and placed at the site. You can connect this new component to other components in your design.

   The Component Properties property sheet may appear on the screen for each newly created component. Whether the property sheet appears or not is controlled by the Automatic Post option in the Main Properties property sheet (see the "Main Properties" section). Each property sheet shows the name of the new component and the name of the site in which the component is placed.

3. You can edit the component properties in the property sheet and click **OK** or **Apply** to apply them, or **Cancel** to close the property sheet and maintain the defaults. See the "Component Properties" section for details.

Notes on Adding Components

This section includes some additional information on adding components to your design.

*   When you add a component, the name assigned to the new component is in the format $COMP_*number*, where *number* is a number assigned to each new component. Numbering starts at 0 and increases by one for each new component. For example, if you already have 100 components and you add a component at site HF, the new component is named $COMP_101. The next component, added at site CF, is named $COMP_102, and so on. You can change the component's name by modifying the Component Properties property sheet.

*   Any characters special to the FPGA Editor command interpreter must be preceded by a backslash (\) escape character when used in a component name. Special characters are quotation marks (" or ') * ; ? # – (leading dash). The restriction only applies to commands entered at the command line or those in an FPGA Editor command script file; the special characters may be entered in the Component Properties property sheet without the escape character.

    Note If you use a tool set other than the Xilinx tools, additional naming restrictions may apply.

*   All components and nets have their own name space and all components and nets must have unique names within their own name space. For example, you can have a component and a net named "FRED," but you cannot have two components named "FRED" or two nets named "FRED." The FPGA Editor prevents you from entering a duplicate name in a name space.

*   When you add a component in the FPGA Editor, the component is assigned default programming. To determine the default programming, add the component, select it, and click **Info** in the User toolbar. The defaults are listed in the History toolbar.

Adding a Net

To add a new net to the database use this procedure.

1. Select all component pins comprising the new net. Each of the selected pins must be on a placed component (not a site), and the selected pins must not belong to any other nets.

2. Use the Add command in any of the following ways to add a net.

   ♦ Select **Edit** → **Add**.

   ♦ Select **Add** in the User toolbar.

   ♦ Enter **add** in the Command Line toolbar.

   The selected pins are joined as a net.

   After using the Add command, a Net Properties property sheet may appear for the newly created net. Whether the property sheet appears or not is controlled by the Automatic Post option in the Main Properties property sheet. See the "Main Properties" section for details. The property sheet shows the name of the new net, the number of pins and the routing status. You can edit the net properties in this property sheet or select **Cancel**. See the "Net Properties" section for details.

   If the Automatic Routing Option is enabled, the new net is routed as completely as possible. If Automatic Routing is Off, ratsnest lines will connect the pins on the new net. For a description of the Automatic Routing option, see the "Automatic Routing Option" section.

Notes on Adding Nets

This section includes additional information on adding nets to your design.

• When you add nets, the name assigned to the first new net is $SIG_0; subsequently added nets will be named $SIG_1, $SIG_2, and so forth. You can change this net name by using the Net Properties property sheet that appears when the net is added.

• Any characters special to the FPGA Editor command interpreter must be preceded by a backslash (\) escape character when used in a net name. Special characters are quotation marks (" or ') * ; ? # – (leading dash). The restriction only applies to commands

entered at the command line or those in an FPGA Editor command script file; the special characters may be entered in the Net Properties property sheet without the escape character.

Note If you use a tool set other than XILINX tools, additional naming restrictions may apply.

- All nets and components have their own name space and all nets and components must have unique names within their own name space. For example, you can have a net and a component named "FRED," but you cannot have two nets named "FRED" or two components named "FRED." The FPGA Editor prevents you from entering a duplicate name in a name space.

Adding Component Pins to an Existing Net

To add component pins to an existing net follow these steps.

1. In the Array window, select the component pins you want to add to the net.

   The selected pins must be on a placed component (not a site), and the selected pins must not belong to any other net.

2. Select the net to which you will add the pins in one of the following ways.

   - Display a list of nets in the List window, then select the net name from this list.

   - Select any net pin, ratsnest line, or routed segment on the desired net.

   - Enter **select** in the Command Line toolbar.

3. Select **Edit → Add**.

   The selected pins are added to the net. If the Automatic Routing option is enabled, the new net pins are routed to the net. If Automatic Routing is Off, ratsnest lines are drawn to the new net pins. For a description of Automatic Routing, see the "Automatic Routing Option" section.

Adding a Macro

Adding a macro to your design is described in the "Adding Macros to Your Design" section of the Working with Physical Macros chapter.

Deleting Objects

Deleting an object permanently removes it from the current design. You can delete these objects.

- Components
- Nets
- Net pins
- Macros
- Paths

Select the objects you want to delete, and then perform one of the following commands.

- Select **Edit** → **Cut**.
- Select **Delete** in the User toolbar.
- Enter **delete** in the Command Line toolbar.

This section contains the following topics.

- "Deleting Components"
- "Deleting Nets"
- "Deleting Net Pins"
- "Deleting Macros"
- "Deleting Paths"
- "Deleting Path Elements"

Deleting Components

When you delete a component, each net pin on the component is unrouted and the component is removed from the design database.

Note Do not use the Unplace command to delete an object.

Use the following procedure to delete components.

1. Select the components to delete.

2. Select **Edit** → **Cut**.

    The components are eliminated from the database. You cannot delete a component that is part of an instantiated macro.

Deleting Nets

When you delete a net, all net pins on the net are unrouted, each net pin is removed from the design database, and the net itself is removed from the database.

Use the following procedure to delete a net.

1. In the List window, display a list of net names.

2. Select the nets to delete.

    If you are unsure of the name of a net, position the mouse pointer over the net in the Array window to display the data tip for the net. The net name is displayed in a small floating window.

3. Select **Edit** → **Cut**.

    The nets are eliminated from the database.

Deleting Net Pins

When you delete a signal pin, the pin is unrouted from the net to which it is attached, and the pin is removed from the net. A deleted pin no longer has any connection (logically or physically) to the net from which it was deleted.

Use the following procedure to delete net pins.

1. Select the net pins to delete.

2. Select **Edit** → **Cut**.

    The net pins are removed from the nets to which they are attached and from the database.

Deleting Macros

Deleting a macro from the design is described in the "Deleting Macros from Your Design" section of the "Working with Physical Macros" chapter.

Deleting Paths

When you delete a path, you remove the Define Path constraint that defines the path. You also remove all constraints that directly depend on the Define Path constraint, such as Maxdelay Path.

Use the following procedure to delete a path.

1. Select the path to delete by selecting the path name in the List window or by entering the following in the Command Line toolbar.

    select path *path_name*

2. Select **Edit** → **Cut**.

    The next time you save the design, the Define Path constraint that originally defined the path is commented out in the constraints file.

Deleting Path Elements

You can delete nets or components from path definitions. When you delete them, they are not deleted from the design; they are only removed as elements in the path.

Use the following procedure to delete nets or components from a path.

1. Select the nets or components to remove from the path.

2. Enter the following in the Command Line toolbar.

    delete path *path_name*

    Path_name is the name of the path from which you will remove the nets or components.

    The selected nets or components are removed from the path definition. The next time your design is saved, the constraints file is modified, and the selected nets or components are removed from the Define Path constraint.

Highlighting Objects

When you edit a design in the FPGA Editor, you may want to change the color of (highlight) one or more objects for easy reference. Highlighting is especially useful in dense designs.

To use the highlighting feature, you must select the Automatic Hilite option in the Main Properties property sheet. See the "Main Properties" section for more information. When this option is enabled, delay paths are highlighted after a delay command, and nets are highlighted after a manual route. By default, the Automatic Hilite option is disabled. You can also set highlighting from the Command Line toolbar with the Setattr command. See the FPGA Editor online help for information on this command.

Here are some additional reasons for highlighting.

- Highlighting makes it easier to find objects in your design.

- After running a path delay, you may want to highlight the path you just checked.

- You may want to highlight the routes added during manual routing.

Before you can use the highlight feature, the layer containing the objects that you want to highlight must be selected in the Layer Visibility toolbar.

You can highlight selected objects in the following ways.

- Select `Hilite` in the User toolbar.

- Select `View` → `Highlight`.

- Enter the `Hilite` command in the Command Line toolbar.

To change the color of a selected object to yellow, enter the following in the Command Line toolbar.

```
hilite –c yellow
```

See the FPGA Editor online help for information on the Hilite command.

Selected objects are deselected based on the setting of the Automatic Deselect option in the Main Properties property sheet. If this option is not enabled, selected objects do not appear highlighted until you select another object. If this option is enabled, highlighted objects appear in the proper color automatically.

Viewing and Changing Properties

You can view and change various properties for the following design objects.

- FPGA Editor window

- Components

- Component pins

- Nets

- Sites

- Ratsnest lines

- Wires (pinwires, local lines, and long lines)

- Macros

- Paths

- Object layers

Some of the properties you view and change represent constraints applied to your design. If you change these properties in the FPGA Editor, the resulting constraints changes are written into the constraints file when you save your design.

Use one of the following methods to display a property sheet for a selected object.

- Select **Edit → Properties of Selected Items**.

- Select **Attrib** in the User toolbar.

- Enter **post attr** in the Command Line toolbar.

If you want to post properties for more than one object, select all of the objects first. The property sheets are displayed in the order in which the objects were selected.

You can modify some of the properties in the property sheet; some of the properties are read-only fields and cannot be modified. To activate any changes, click **OK** to close the property sheet, or click **Apply** to keep the property sheet open.

This section contains the following topics.

- "Getattr and Setattr Commands"

- "Main Properties"

- "Component Properties"

- "Pin Properties"

- "Net Properties"

- "Site Properties"

- "Wire Properties"

- "Macro Properties"

- "Path Properties"

- "Layer Properties"

Getattr and Setattr Commands

The Getattr and Setattr commands are also related to object properties. Use the Getattr command to display properties for selected objects in the History toolbar. You can also use the Info button in the User toolbar to display properties. Use the Setattr command to change properties for selected objects.

The information provided by the Getattr command is the same as that displayed in an object's property sheet. The properties changed with the Setattr command are the same properties you can modify in an object's property sheet. You may want to use these commands when you do not want to display the property sheet for the object. Getattr and Setattr are also described in the FPGA Editor online help.

The Getattr, Setattr, and Post Attr commands are all affected by the Automatic Deselect option in the Main Properties property sheet. When Automatic Deselect is enabled, selected objects are automatically deselected after a command is executed. If the Automatic Deselect option is disabled, the object remains selected.

Main Properties

To view or change properties for the FPGA Editor and the current design, perform one of the following commands.

- Make sure nothing is selected in the Array window or the List window, and select **File → Main Properties.**

- Make sure nothing is selected in the Array window or the List window, and select **Attrib** in the User toolbar.

- Make sure nothing is selected in the Array window or the List window, and enter **post attr** in the Command Line toolbar.

The Main Properties property sheet appears as shown in the following figure. Refer to the FPGA Editor online help for a detailed description of this property sheet.
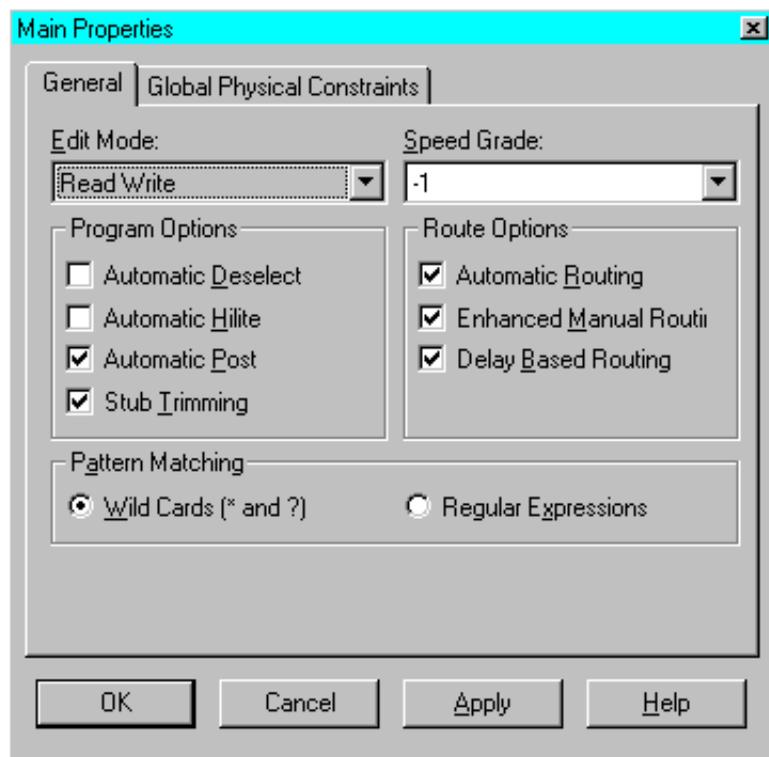
Figure 3-6 Main Properties Property Sheet

Component Properties

The Component Properties property sheet appears when you select a component and select **Edit → Properties of Selected Items**.

This property sheet contains the General, Configuration, and Physical Constraints pages.
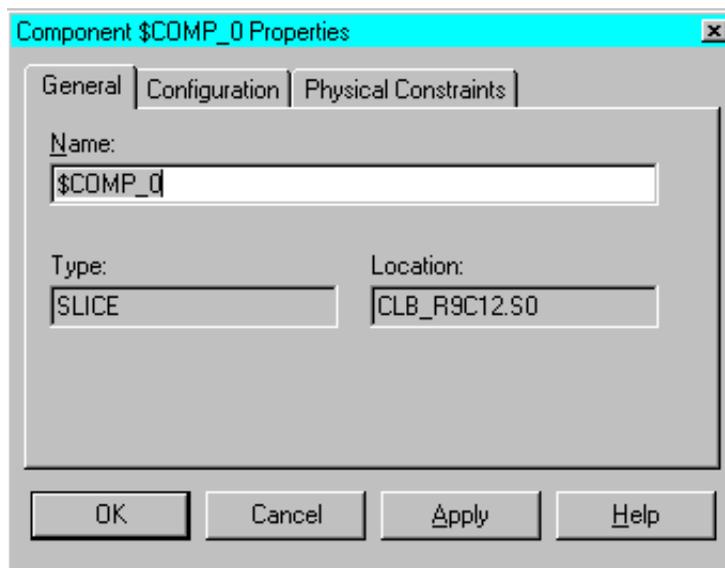
Figure 3-7 Component Properties Property Sheet

General Page

Refer to the following table for a description of the General page.

Property	Description
Name	Name of the component; there is no character limit
Type	Type of component, for example, CLB, TBUF, PULLUP
Location	Indicates the reference designator of the site in which the component is placed or indicates "unplaced"

Configuration Page

Note The Configuration page is not displayed if the selected component is a protected or confidential component.

This page displays the internal programming of the CLB or IOB. You cannot edit any of the properties displayed on this page, and the properties may differ between device families. Use the Block window to modify CLB or IOBs. Refer to the following table for a description of some of the possible properties displayed in the Configuration page.

Property	Description
Base	Base mode of the component
Config	State of the component, including multiplexers and other logic elements
Feqn	Boolean equation for the F output of the component's combinatorial logic section
Geqn	Boolean equation for the G output of the component's combinatorial logic section
Heqn	Boolean equation for the H output of the component's combinatorial logic section
Carry	Component carry strings

Physical Constraints Page

This page displays the constraints that are applied to the component. Refer to the following table for a description of the Physical Constraints page.

Property	Description
Lock Placement	Locks component to its current location
Requirement	Indicates whether the automatic placement range is an absolute requirement (Hard) or preferable (Soft)
Effort	Sets up a priority for the autoplacer to resolve constraint conflicts
Location Range	Specifies the location of the site or range of sites for the autoplacer
Block Paths	A timing constraint to block the enumeration of all timing paths that go through this component
TSid	Assigns a timing period or frequency to a timing specification

Pin Properties

The Pin Properties property sheet appears when you select a site or component pin and select **Edit → Properties of Selected Items**. If the selected pin is a macro external pin, a different property sheet appears. See the "Working with Physical Macros" chapter.
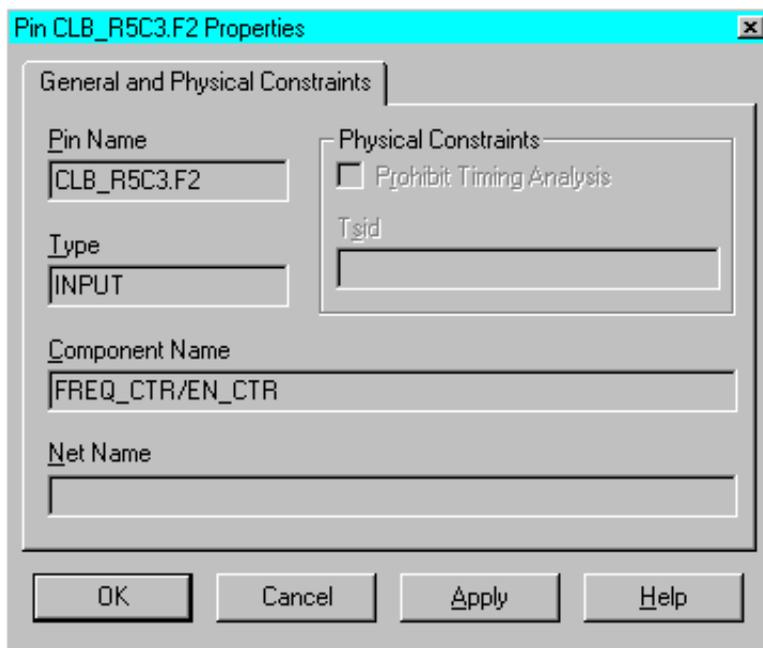
Figure 3-8 Pin Properties Property Sheet

Refer to the following table for a description of the General and Physical Constraints page properties. Only the Physical Constraints fields can be edited.

Property	Description
Pin Name	Name of the pin
Type	Indicates whether the pin is an INPUT, OUTPUT, TRISTATE, or CLOCK pin
Component Name	Name of the component containing the pin
Net Name	Name of the net to which this pin belongs. If the pin is unused, this field is empty.
Prohibit Timing Analysis	Applies a Timing IGnore (TIG) constraint to the pin, which prevents timing analysis on paths through that pin.
TSid	Assigns a timing period or frequency to a timing specification

Net Properties

The Net Properties property sheet appears when you select a net and select **Edit** → **Properties of Selected Items**. This property sheet contains the General, Physical Constraints, and Pins pages. A partially or fully routed net can be selected by pressing the Shift key while selecting any routed segment on the net. An unrouted net can be selected by selecting the ratsnest; by selecting its name from a list in the List window; or by using the Select command in the Command Line toolbar.
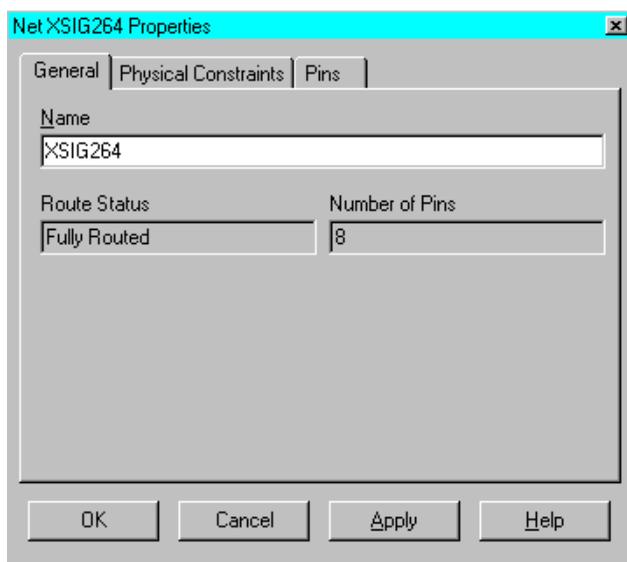
Figure 3-9 Net Properties Property Sheet

General Page

Refer to the following table for a description of the General page.

Property	Description
Name	Net name
Route Status	Indicates whether the net is Unrouted, Partially Routed, or Fully Routed. For VCC and Ground nets, the status is preceded with either PWR or GND.
Number of Pins	Indicates how many net pins this net contains

Physical Constraints Page

This page displays the constraints that have been applied to the net. Refer to the following table for a description of the Physical Constraints page.

Property	Description
Lock Status	Indicates whether the net is locked or unlocked. If net is locked, also indicates whether the net is locked due to a lock net or lock routing constraints. A locked net cannot be unrouted or deleted. Pins and components connected to the net cannot be unplaced, swapped, or deleted.
Lock Routing of this Net	Locks the net by applying a Lock Net constraint; this net cannot be unrouted or deleted.
Prohibit Timing Analysis	Applies a Block Net constraint to the net, which prevents timing analysis on the named net and on all paths through the named net.
Period (ns)	Applies a Period constraint to the net, which assigns a clock period for all sequential output and input pins clocked by the net. Enter a period or a frequency in this field, but not both.
First Pulse	Specifies whether the first pulse of the duty cycle will be High or Low. The field immediately to the right of this field specifies the duration of the first pulse of the duty cycle.
Duration (ns)	Specifies the duration of the first pulse of the duty cycle.
TSid	Assigns a timing period or frequency to a timing specification.
Phase (ns)	Generates a derived clock based on the period of an existing timing constraint. This constraint is the master and the derived constraint is the slave. Phase is the delay of the initial edge of the master to the slave.

Property	Description
Factor	Generates a derived clock based on the period of an existing timing constraint. This constraint is the master and the derived constraint is the slave. Factor is a multiplier period of the master used to create a relative one for the slave. For example, if the period for the slave is one half that of the master, use a value of 0.5.
Max Skew	Applies a Max Skew constraint to the clock net. The Max Skew constraint specifies the difference between the minimum and maximum load delays on the net.
Route Priority	Nets with a higher priority are routed first.
Delay (ns)	Applies a Max Delay constraint to the net, which specifies a maximum total delay for the driver-to-load connections on the net.
Priority	Applies a Prioritize Net constraint to the net. The constraint assigns a weighted importance (0–100) to the named net (with 0 as least important and 100 as most important). A net with a priority of 3 or less is not considered critical.

Pins Page

This page displays the net pins comprising the net. You cannot edit any of the information in the Pins page. The Pins page only displays information for net pins on placed components. To view a pin in the Array window, click the left mouse button on any of the pins listed, enable the Synchronize Pin Selection with Array Window option, and click Zoom. Refer to the following table for a description of the Pins page.

Property	Description
Name	Name of the pin, in the form, reference_designator.pin_name
Type	Indicates whether the pin is an INPUT, OUTPUT, TRISTATE, or CLOCK pin
Delay (ns)	Delay time from the net source to this pin. This field is blank if the pin is unrouted. For VCC and ground pins, the delay is indicated with a dash.

Site Properties

The Site Properties property sheet appears when you select a site and select **Edit → Properties of Selected Items**.
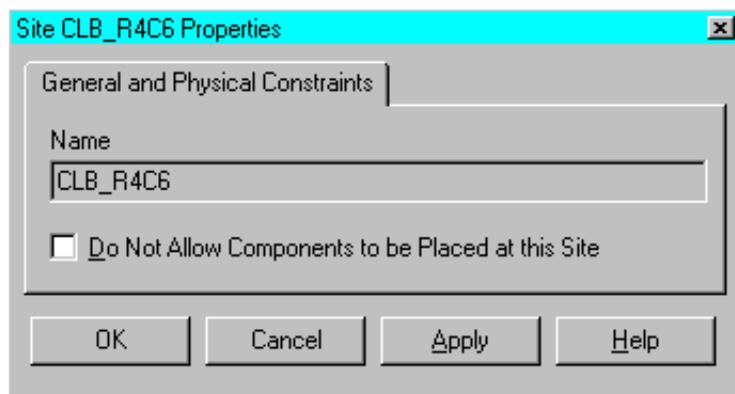
Figure 3-10 Site Properties Property Sheet

Use the Do Not Allow Components to be Placed at this Site option to apply a Prohibit Site constraint to the site. This constraint specifies

that a component cannot be placed at this site. You may want to reserve a site for future use.

Wire Properties

The Wire Properties property sheet, shown in the following figure, appears when you select a pinwire, a local line, or a long line and select **Edit → Properties of Selected Items**. The property sheet appears whether you select a routed or an unrouted portion of a wire. You cannot modify any of the properties in this property sheet.
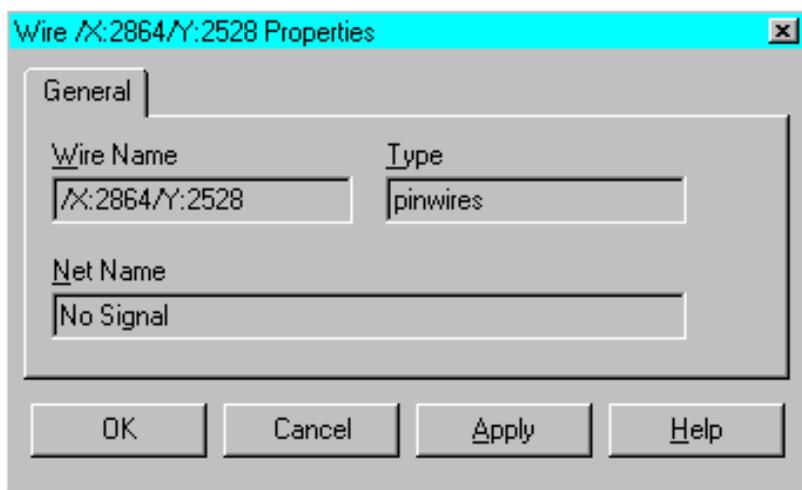
Figure 3-11 Wire Properties Property Sheet

Refer to the following table for a description of the Wire properties.

Property	Description
Wire Name	Name of the wire
Type	Indicates whether the selected wire is a PINWIRE, LOCAL line, LONG line, SWITCH BOX line, or PROGRAMMED SWITCH.
Net Name	Specifies the net routed along this wire.

Macro Properties

Macro properties are described in the "Viewing and Changing Macro Properties in Your Design" section of the "Working with Physical Macros" chapter.

Path Properties

The Path Properties property sheet appears when you select a path and select **Edit → Properties of Selected Items**.

Refer to the following table for a description of the Path properties.

Property	Description
Name	Name of the selected path
Elements	Names of the components and nets comprising the path.
Prohibit Timing Analysis	Indicates that a Block Path constraint is applied to the path, which blocks timing analysis on the path
TSid	Assigns a timing period or frequency to a timing specification
Maxdelay	Applies a Maxdelay constraint to the path, which specifies the maximum delay for the path
Priority	Lists the priority and the locked attribute for the Q1 net

Layer Properties

An FPGA Editor layer contains all of one type of object, for example, all long lines in the device or all components in the design database. To see a list of all available layers, select Layers in the List window.

The Layer Properties property sheet appears when you select any FPGA Editor layer and select **Edit → Properties of Selected Items**. Layers cannot be selected in the Array window; they must be selected from the layers list in the List window or by using the Select command in the Command Line toolbar.
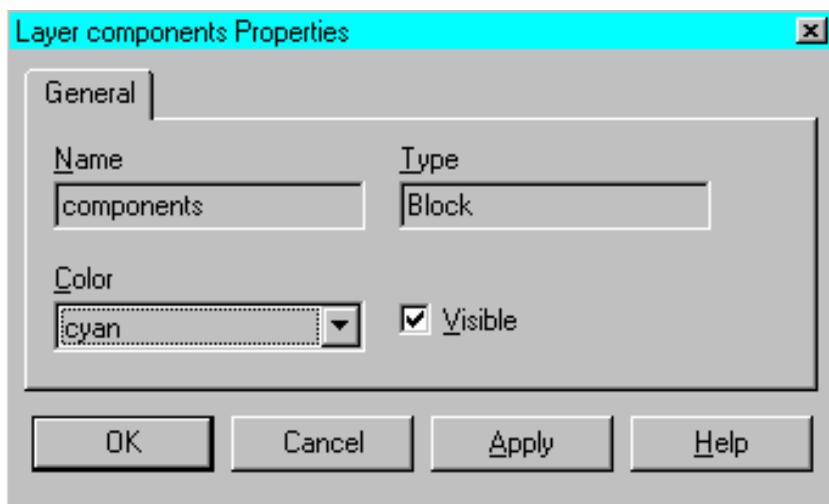
Figure 3-12 Layer Properties Property Sheet

Refer to the following table for a description of the Layer properties.

Property	Description
Name	Type of object in the layer
Type	Indicates what type of layer: BLOCK, WIRE, NODE, or OTHER
Color	Determines the color of objects in the layer; use the pull-down list box to select a color
Visible	If enabled, the layer appears in the Array window

Moving and Swapping Components and Macros

Use the Swap command to do any of the following.

- Move a placed component to an unused site

- Move a placed macro to an unused series of sites

- Swap the locations of two placed components

Moving Components with the Swap Command

To move a placed component to an unused site, perform the following procedure.

1. Select the component you want to move.

2. Press the Control key and the left mouse button to select the unused site for the component.

    The component and the site must be the same type of block: IOB and IOB, or CLB and CLB.

3. Select **Edit** → **Swap**.

    The component moves to the unused site. If Automatic Routing is enabled, the component is routed after it is placed.

Moving Components using Drag and Drop

You can use the drag and drop method to move *placed* components from one location to another within the Array window. You can also drag and drop *unplaced* components from the List window to an unused site in the Array window.

To move a placed component to an unused site using the drag and drop method, select the component with the right mouse button and drag it to the selected site. To move an unplaced component from the List window, select the component with the left mouse button and drag it to the selected site. If the Automatic Routing option is enabled, the component is routed after it is placed.

Moving Macros

Moving macros is described in the "Moving Macros in Your Design" section of the "Working with Physical Macros" chapter.

Swapping Components

To swap the locations of two components, perform the following procedure.

Note The components you are swapping must be placed components.

1. Select one of the components to swap.

2. Press the Control key and the left mouse button to select the second component.

    Both components must be the same type of block: IOB and IOB, or CLB and CLB.

3.   Select **Edit** → **Swap**.

The two components are unrouted, and their locations are switched. If Automatic Routing is disabled, the components remain unrouted after the swap, even if they were routed before the swap. If Automatic Routing is enabled, the two components are routed after they are swapped.

Swapping Component Pins

You can also use the Swap command to swap pins on a component.

The FPGA Editor only allows you to swap pins that are permitted for the mode in which the component is programmed. When the pins are swapped, the internal component logic is modified to match the new pin positions.

To swap component pins, perform the following steps.

1.   Select one pin and then press the Control key and the left mouse button to select the second pin.

2.   Select **Edit** → **Swap**.

The two pins are unrouted, and their locations are switched. If Automatic Routing is disabled, the pins remain unrouted after the swap, even if they were routed before the swap. If Automatic Routing is enabled, the two pins are routed after they are swapped. The programming of the component changes to reflect the pin swap.

Editing Component Logic

You can use the Block window in the FPGA Editor to modify or create component logic. See Figure 2-11 of the "Getting Started" chapter. You can use only one block window at a time for editing; however, you can have additional block windows open for viewing.

To edit a internal logic of a component, perform one of the following.

•   Double click the left mouse button on a component, and then press the second to the left button in the Block window toolbar (the Begin Editing button). If it is disabled, check to make sure the edit mode is set to read-write.

- Select a component and select the **Editblock** command in the User toolbar

- Select a component and enter **editblock** in the Command Line toolbar

Notes on Editing a Component

This section provides additional information on editing components.

- Look-up tables (LUT) and flip-flops have several check boxes associated with them. You can enable a LUT or flip-flop by selecting the appropriate check box.

- You can also specify carry mode by selecting the appropriate check box.

- Multiplexer (MUX) symbols have triangular input pins. You can click on the appropriate pin to highlight the path through that pin.

- You can create LUT equations by using attributes for the component. In the Block window, click on the Show/Hide Attributes toolbar button. Equations are a boolean representation of the logic; for example, F4 + (~F3*F2). See the following table for symbol definitions.

Symbol	Logical Gate
~	NOT
+	OR
*	AND
@	XOR

Placing and Unplacing Components

Note For information on placing or unplacing macros, see the "Working with Physical Macros" chapter.

Components can be placed in the Array window automatically or manually. When you use automatic placement (AutoPlace), the system selects an appropriate site or sites for each component you select for placement. When you use manual placement, you select both a source component and the destination site in which to place it.

The automatic placement software selects sites based on routability. You can automatically place selected components or automatically place all remaining unplaced components in your design.

You can also unplace components. Unplacing unroutes each net pin on the components you select, and removes the components from the Array window and adds them to the list of your design's unplaced components.

This section contains the following topics.

- "Automatically Placing Selected Components"

- "Automatically Placing All Unplaced Components"

- "Manually Placing Selected Components"

- "Unplacing Components"

Automatically Placing Selected Components

You can automatically place selected components with the Auto Place command. To automatically place selected components, perform the following procedure.

1. In the List window, display a list of unplaced components.

2. In the list of unplaced components, select the components to automatically place.

3. Select **Tools** → **Place** → **Auto Place**.

The selected components are automatically placed in vacant sites in the Array window. If the Automatic Routing Option is enabled, the components are routed after they are placed.

Notes on Placing Selected Components

This section includes additional information on placing components.

- Before the selected components are automatically placed, any unplaced pull-ups or tristate buffers in your design are automatically placed.

- Placing certain components may cause other related components to be placed.

- Because the Place and Route program (PAR) unroutes and places components more efficiently than the FPGA Editor, when you

automatically place components in the FPGA Editor, existing routing is *not* unrouted and rerouted to achieve better placement.

Automatically Placing All Unplaced Components

To automatically place all unplaced components, perform the following procedure.

1. Select **Tools** → **Place** → **Auto Place All**.

    The Autoplace All Components dialog box is displayed as shown in the following figure. Refer to the FPGA Editor online help for a detailed description of the fields in this dialog box.
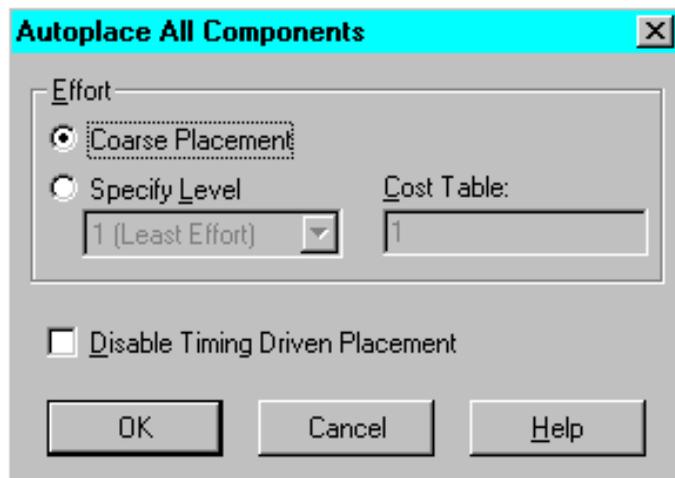
Figure 3-13 Autoplace All Components Dialog Box

2. Click **OK**.

    All of the unplaced components in the design are automatically placed in vacant sites in the Array window. If the Automatic Routing Option is enabled, the Autoroute All Nets dialog box appears and you are prompted for information on the type of automatic routing. If Automatic Routing is disabled, enter **autoroute -all** in the Command Line toolbar.

    Note When you autoplace all remaining components in the FPGA Editor, existing routing is *not* unrouted and rerouted to accommodate the new placement. Because PAR unroutes components if necessary, placement done with PAR produces different

results than placement done in the FPGA Editor. PAR places components more efficiently, and a densely packed design can more easily be placed and routed with PAR than with the FPGA Editor.

Manually Placing Selected Components

To manually place an unplaced component, follow this procedure.

1. In the List window, display a list of unplaced components.

2. Select a component from the list.

3. Select a vacant site in the Array window.

    The component and the site must be the same type of block (for example, IOB and IOB, or CLB and CLB).

4. Select **Tools** $\rightarrow$ **Place** $\rightarrow$ **Manual Place**.

    The component is placed in the site. If the Automatic Routing Option is enabled, the component is routed after it is placed.

Unplacing Components

The Unplace command removes all or selected components from their current sites. Before unplacing a component, the system unroutes each component pin on the component.

Note Locked components cannot be unplaced.

Selected Components

To unplace selected components, use the following procedure.

1. Select placed components from the List window.

2. Select **Tools** $\rightarrow$ **Place** $\rightarrow$ **Unplace**.

The selected components are unplaced. These components now appear as unplaced in the List window.

All Components

To unplace all placed components in your design, follow these steps.

1. Select **Tools** $\rightarrow$ **Place Unplace All**.

2. Select **Yes** in the confirmation box that appears to unplace all components.

    All unlocked placed components in your design are unplaced.

Routing and Unrouting

All or selected connections can be routed in the Array window automatically or manually. When you use automatic routing, the system selects the routing path for each connection you select. In manual routing, you specify the routing path.

You can enable the Automatic Routing option to automatically route any unrouted net pins created as a result of an editing action, such as placing a component or creating a new net. Automatic Routing, when enabled, occurs after add, place, autoplace, and swap operations.

You can also unroute connections. Unrouting removes the routing from any specified routes without eliminating the logical connections of the nets.

This section contains the following topics.

*   "Automatic Routing"
*   "Manual Routing"
*   "Automatic Routing Option"
*   "Unrouting"

Automatic Routing

Use the Auto Route command to automatically route selected design objects, including the following.

*   Nets
*   Net pins
*   Ratsnest lines
*   Components
*   Macros

In addition, you can autoroute between two resources including the following.

- Route segments

- Used wires

- Used or unused component pins

Note If both resources are used, they must be on the same net.

To route selected objects automatically, follow these steps.

1. Select the objects you want to route automatically.

   You can select any combination of nets, net pins, ratsnest lines, components, macros, and partially or fully routed nets in the Array window. Unrouted nets can only be selected in the List window.

2. Select **Tools → Route → Auto Route**.

   The objects you selected are routed (if possible). The display for each routed connection changes from a ratsnest display to a routed net display. If an object could not be completely routed, an error message appears in the History toolbar and the ratsnest remains.

Notes on Automatically Routing Selected Objects

This section includes additional information on automatically routing selected objects.

- When an output pin (a net driver) is automatically routed, the entire net is routed. When an input pin (a net load) is automatically routed, the connection between the pin and the net driver is routed.

- When a component is automatically routed, each net pin on the component is routed.

- When a net or a ratsnest line is automatically routed, the entire net is routed.

- Delay-based routing uses delay values when routing your design. If this type of routing is disabled, cost-based routing is performed instead. Delay-based routing takes longer, but can result in better routing, especially in a dense design. By default, delay-based routing is enabled; this option has no effect when you autoroute all nets.

- When selected objects are autorouted in the FPGA Editor, existing routing is *not* unrouted and rerouted to accommodate new routing. Also, component pins are not swapped for a better routing. Because the PAR program unroutes routes and swaps pins if necessary, routing with PAR is more efficient than routing in the FPGA Editor.

To automatically route your entire design use the following procedure.

1. Select **Tools** → **Route** → **Auto Route All**.

    The Autoroute All Nets dialog box appears as shown in the following figure. The fields in this dialog box are described in the FPGA Editor online help.
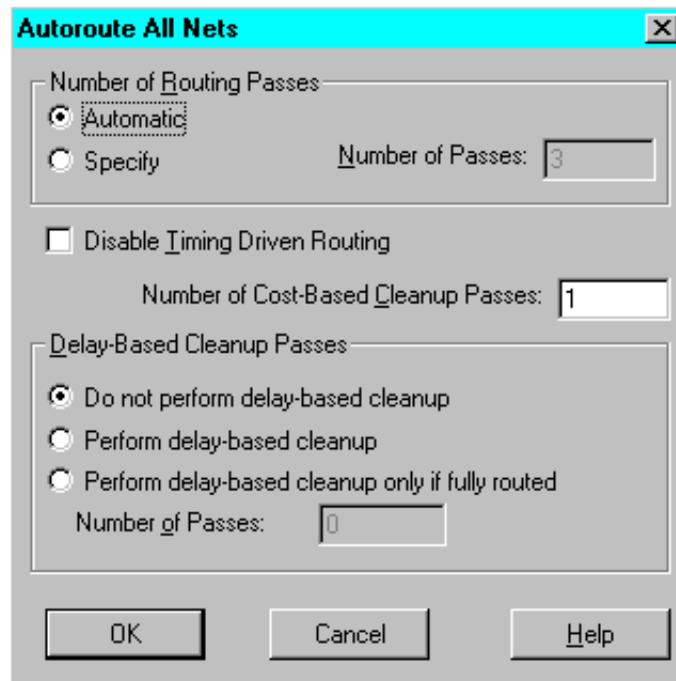
Figure 3-14 Autoroute All Nets Dialog Box

2. Specify the routing options you want and click **OK** to close the dialog box.

    All unrouted connections are routed if possible. The display for each routed connection changes from a ratsnest display to a

routed net display. If a net cannot be routed to completion, an error message appears in the History toolbar.

Notes on Automatically Routing the Entire Design

This section includes additional information on automatically routing your entire design.

- When your entire design is automatically routed, the following default routing options are performed.

  - Three iterations of the router

  - One cost-based cleanup pass

  - No delay-based cleanup passes

- To specify the number of routing iterations instead of accepting the default value (3), enter a different value in the Number of Passes field in the Autoroute All Nets dialog box. The number (1 to 999) specifies the maximum number of iterations performed. The router continues to perform iterations until one of the following events occurs.

  - Your design is routed to completion and constraints are met.

  - The router completes the number of iterations you specified.

  - The router cannot route your design to completion or meet constraints.

- When your entire design is routed, existing routing is *not* unrouted and rerouted to accommodate new routing. Because the PAR program unroutes routes and swaps pins if necessary, routing with PAR is more efficient than routing in the FPGA Editor.

Manual Routing

Use the Route command to manually route your design. Manual routing allows you to specify the path for the signal routing. Select the net pins or pinwires to connect and the routing resources to use (such as long lines or local lines), and the system routes the specified path. As part of the manual routing, you can select unused pins on placed components and then route these pins together to form a new net, or route them to an existing net.

To route your design manually, follow these steps.

1. Click the left mouse button to select the source pin wire.

2. Press the Control key, Shift key, and the left mouse button simultaneously to select additional routing resources.

    You must select objects in the order in which you want them routed. When you select the objects, consider that connections are routed one by one from each selected object to the next. For example, if you select three objects in the order A, B, C, you are indicating that you want to route A to B, then route B to C.

3. Select **Tools** → **Route** → **Manual Route**

    The selected objects are routed in the specified order.

Notes on Manual Routing

This section includes additional information on manually routing your design.

*   Selected objects must be part of the same net. For example, you cannot select a net pin that is part of one net and a long line that is part of another net.

*   If a connection does not exist between specified objects that connection is not routed, and an error message appears in the History toolbar. If the Enhanced Manual Routing option is enabled in the Main Properties property sheet, the FPGA Editor attempts to autoroute between the two selected routing resources. The Enhanced Manual Routing option is described in the FPGA Editor online help.

*   When you specify objects to route, consider the direction of current flow through the switches connecting the two objects. You can route from source to load or from load to source, but you cannot mix the two directions during a single Route command run.

*   When you select an unused net pin or pinwire for a connection, it is added to the net you are routing.

*   Manual routing only routes the specified connections. All other unrouted connections on the applicable net remain unrouted, unless the Enhanced Manual Routing option is enabled in the Main Properties property sheet.

- The Stub Trimming option displays only those portions of routing resources, for example, long lines and local lines, actually used by routes. If you disable Stub Trimming, the full routing resources taken up by the routes are displayed. For manual routing, disable Stub Trimming to determine which routing resources are available. Stub Trimming is described in the FPGA Editor online help.

- Partially routed nets that do not terminate at pins (antennas) are unrouted by the Autoroute –all command if you do not lock the net. Unlocked antennas are also removed if you run PAR on your design file.

- Any segment added to a net with manual routing is locked or unlocked, depending on the net's lock status.

- You may want to specify Automatic Hilite if you are manually routing a very dense design.

- You must select objects in the order in which you want them routed. When you select the objects, consider that connections are routed one by one from each selected object to the next. For example, if you select three objects in the order A, B, C, you are indicating that you want to route A to B, then route B to C.

Switch Boxes in Manual Routing

Switch boxes are located at the junctions of horizontal and vertical local lines and can connect local lines. The switch boxes have certain allowable matrices for pin connections. Allowable routes vary, depending on the location of the switch box on the chip. For example, switch boxes on the perimeter of the chip have fewer pins, so the routing matrices are different from those on switch boxes inside the perimeter.

To see the allowable paths for any pin in a given switch box, select the pin. The allowable connections from the selected pin to other pins within the switch box are displayed in yellow. Bi-directional pins are solid and uni-directional pins are hollow. Downhill paths are displayed as bright yellow and uphill paths are displayed as dull yellow. The switch box looks similar to the following figure.
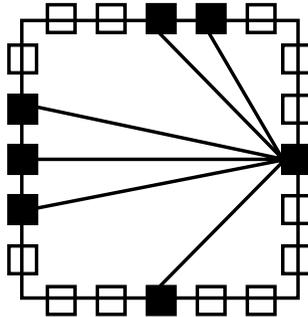
X8956

Figure 3-15 Pin Routing in a Switch Box

To deselect the pin and clear the paths from your screen, click in an empty part of the screen. Click the Shift key and the left mouse button to select multiple switch box pins.

To route a connection through a switch box, select the local lines leading to the pins you want to connect, as shown in the following figure.
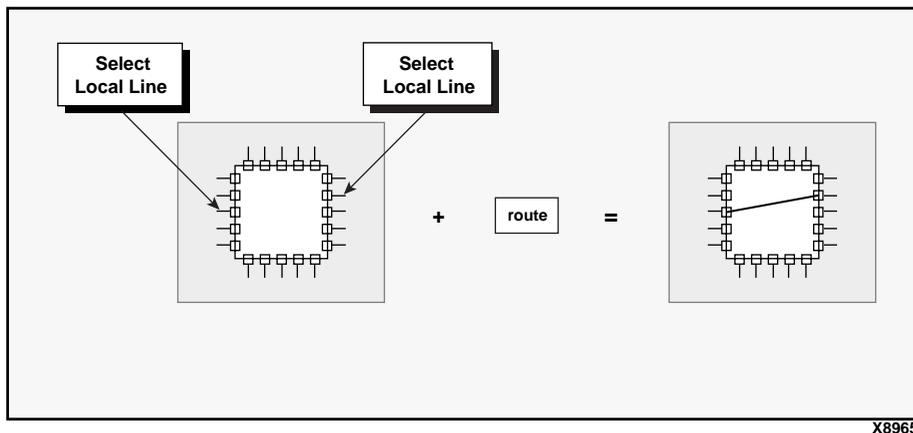
X8965

Figure 3-16 Routing Through a Switch Box

A "bank shot" is a way of indirectly connecting two switch box pins that cannot be connected directly. If you want to route a bank shot through a switch box, select in the correct order the local lines leading to all of the pins you want to connect, as shown in the following figure.

X8966
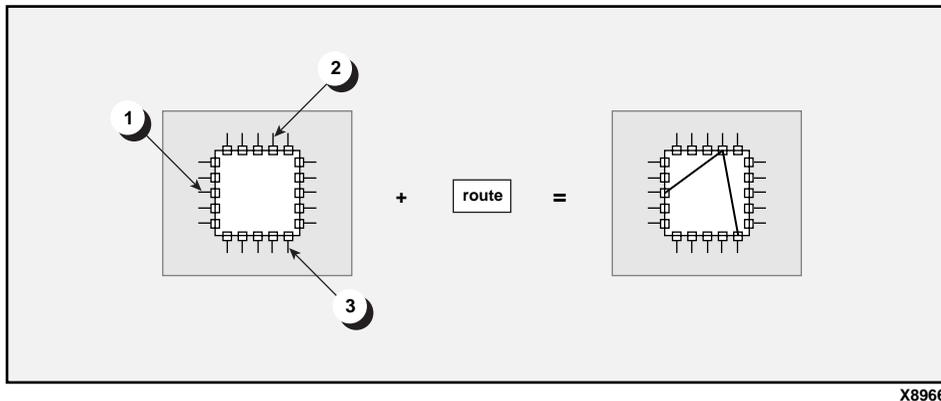
Figure 3-17 Routing a Bank Shot

Routing Through a Logic Block

A type of route called a route-through can pass through an occupied or unoccupied CLB site. A route-through provides routing resources that would otherwise be unavailable.

Use Control + Shift + left mouse button on a pin to highlight all available route-through paths for the pin at that site or component.

You can manually perform a route-through as shown in the following steps for a slice in a Virtex design.

1. During manual routing, select input pin G1, G2, G3, or G4 on the slice site (vacant or occupied) that you want to route through.

    You can also select the pinwire associated with the pin.
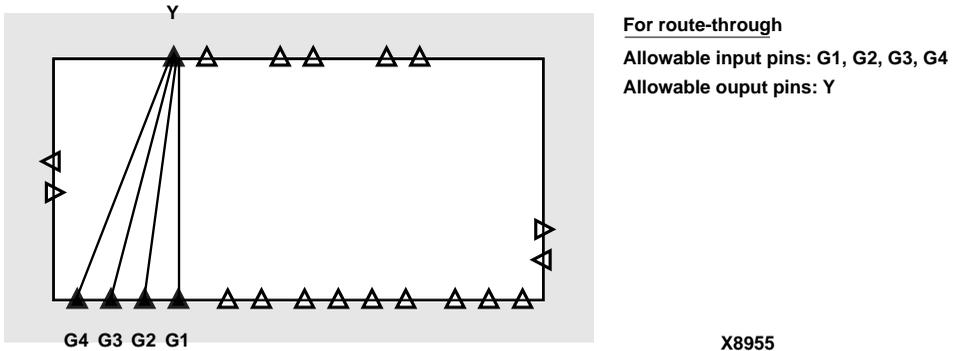
Figure 3-18 Slice Pins for Route-Throughs in Virtex Designs

2. Select output pin Y on the same site.

   You can also select the pinwire associated with the pin.

3. Select **Tools** → **Route** → **Manual Route**.

   The net is routed through the site or component.

   An error message appears if the route-through is not possible because of interfering logic in the slice or pre-existing signals on the route-through pins.

In the FPGA Editor, a route-through appears as a wide line connecting an input to an output pin. The line has the same width and color as other route line segments and is contained in the route graphics layer. If a component occupies the route-through site, the route-through line segment is displayed on top of the component.

Automatic Routing Option

The Automatic Routing option automatically routes any unrouted nets created as a result of an editing action, such as placing a component, creating a new net, or swapping components. If enabled, Automatic Routing is performed after each Add, Copy, Swap, Autoplace, and Place command.

The Automatic Routing option is included in the Main Properties property sheet, and the default setting is enabled. You can change the default setting by editing your fpga_editor.ini or fpga_editor_user.ini file. See the "Initializing the FPGA Editor" section of the "Customizing the FPGA Editor" chapter.

During an editing session, you can toggle the option between enabled and disabled with the following procedure.

1. Select **File → Main Properties** to display the Main Properties property sheet.

2. Select **Automatic Routing**.

3. Click **OK** or **Apply**.

Unrouting

Unrouting disconnects routed connections for specified objects. The logical connections remain after the unroute. The unrouted routes are replaced with ratsnest lines. You can unroute selected objects or unroute your entire design.

Note Locked nets cannot be unrouted.

You can unroute the following objects.

*   Nets

*   Net Pins

*   Route Segments

*   Macros

*   Used local lines, long lines, and pinwires

Unrouting Selected Objects

To unroute selected objects, follow this procedure.

1. Select any combination of components, nets, net pins, route segments, macros, and used wires in the Array window.

2. Select **Tools → Route → Unroute**.

    The objects are unrouted and routed connections are replaced by ratsnest lines.

Unrouting Entire Design

Use the following procedure to unroute your entire design.

1. Select **Tools → Route → Unroute All**.

2. Select **Yes** in the confirmation box that appears to unroute all nets.

    All routed connections in your design are unrouted except the locked signals and certain macro nets. See the "Routing and Unrouting Macros in Your Design" section of the "Working with Physical Macros" chapter.

Unrouting Notes

This section includes additional information on unrouting your design.

*   When you unroute a net pin, only the connection to the net pin is unrouted. All other routed connections on the net are retained.

*   When you unroute a net, the entire net is unrouted.

*   When you unroute a component, each net pin on the component is unrouted.

Constraints

Note For detailed information on attributes and constraints, see the *Libraries Guide.*

You can add certain constraints to or delete certain constraints from the .pcf file in the FPGA Editor. In the FPGA Editor, net, site, and component constraints are supported as property fields in the individual nets and components. Properties are set with the Setattr command and are read with the Getattr command. All Boolean constraints (block, locate, lock, offset, and prohibit) have values of On or Off; offset direction has a value of either In or Out; and offset order has a value of either Before or After. All other constraints have a numeric value and can also be set to Off to delete the constraint. All values are case-insensitive (for example, "On" and "on" are both accepted).

When you create a constraint in the FPGA Editor, the constraint is written to the PCF file whenever you save your design. When you use the FPGA Editor to delete a constraint and then save your design file, the line on which the constraint appears in the PCF file remains in the file but it is automatically commented out.

Some of the constraints supported in the FPGA Editor are listed in the following table.

Table 3-3 Constraints

Constraint	Accessed Through
block paths	Component Properties and Path Properties property sheet
define path	Created, edited, and removed with Add and Delete; viewed with Path Properties property sheet
location range	Component Properties Constraints page
locate macro	Macro Properties Constraints page
lock placement	Component Properties Constraints page
lock macro	Macro Properties Constraints page
lock routing of this net	Net Properties Constraints page
lock placement	Component Properties Constraints page
lock routing	Net Properties Constraints page
maxdelay allnets	Main Properties Constraints page
maxdelay allpaths	Main Properties Constraints page
maxdelay net	Net Properties Constraints page
maxdelay path	Path Properties property sheet
maxskew	Main Properties Constraints page
maxskew net	Net Properties Constraints page
offset comp	Component Properties Offset page
penalize tilde	Main Properties Constraints page
period	Main Properties Constraints page
period net	Net Properties Constraints page
prioritize net	Net Properties Constraints page
prohibit site	Site Properties property sheet

Locked Nets and Components

If a net is locked, you cannot unroute any portion of the net, including the entire net, a net segment, a pin, or a wire. To unroute

the net, you must first unlock it. You can add pins or routing to a locked net.

A net is displayed as locked in the FPGA Editor if the Lock Net [*net_name*] constraint is enabled in the PCF file. You can use the Net Properties property sheet to remove the lock constraint.

When a component is locked, one of the following constraints is set in the PCF file.

`lock comp` [*comp_name*]

`locate comp` [*comp_name*]

`lock macro` [*macro_name*]

`lock placement`

If a component is locked, you cannot unplace it, but you can unroute it. To unplace the component, you must first unlock it.

Using the ILA Command

This section describes how to use the Integrated Logic Analyzer (ILA) command in the Tools menu to view and change the nets connected to the capture units of the ILA core in your design.

Use the following procedure to attach a different net to a capture unit in your design.

Note For more information on the fields in the ILA dialog box, refer to the FPGA Editor online help.

1. Select the net in your design that you want to attach to the capture unit. Note the name of the net in the History toolbar.

2. Select **Tools** → **ILA** to display the ILA dialog box as shown in the following figure.
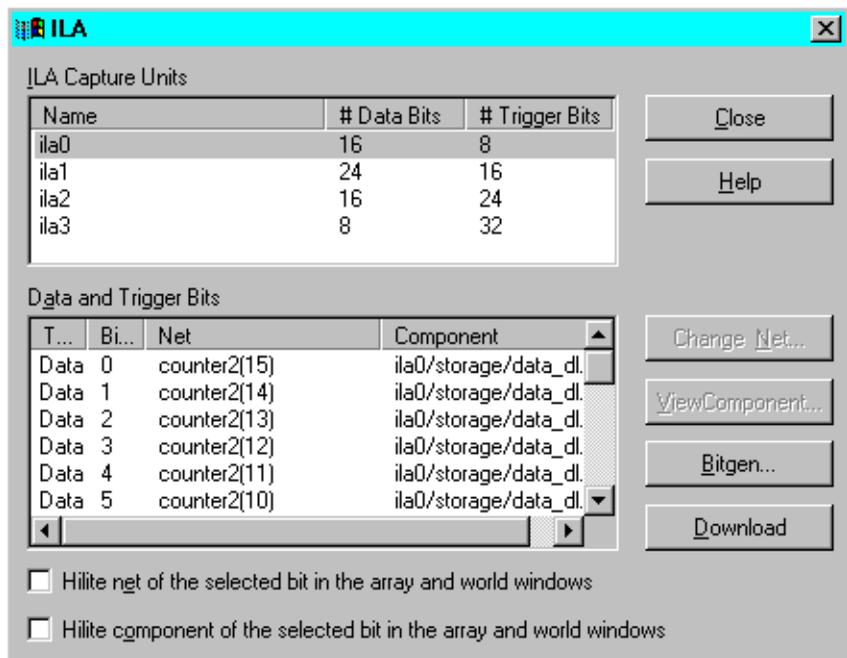
Figure 3-19 ILA Dialog Box

3.   Select the applicable capture unit from the list displayed in the
     dialog box. When you select the capture unit, the nets attached to
     its data and trigger bits are displayed in the Data and Trigger Bits
     section of the dialog box.

4.   Select a Data or Trigger Bit.

5.   Select one or both of the check boxes at the bottom of the dialog
     box to highlight the net or component of the selected bit in the
     Array and World windows.

6.   Find a data or trigger bit that is close to your selected net by
     observing the selection in the Array or World window.

7.   Select **Change Net** to display the ILA Net dialog box.
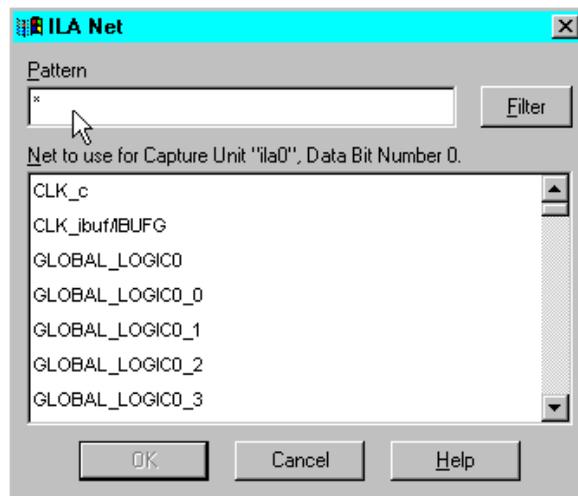
Figure 3-20 ILA Net Dialog Box

8. Enter the name of the net you want to connect to the capture unit in the Pattern field. You can use the Filter button or select from the list of available nets.

9. Click **OK** to unroute the net attached to the capture unit and attach your selected net.

10. To create a BIT file of your modified design, select **Bitgen** in the ILA dialog box. Figure 3-23 is displayed.

11. Click **OK** to create the bit file.

12. Select **Download** to start the Hardware Debugger tool so you can download your bit file to a test device. This tool is run in a separate process from the FPGA Editor.

Using Probes

This section describes how to add, save, and restore probes in your design. Probes are used to examine the signal states of the targeted FPGA device.

Adding a Probe

Use the following procedure to add a probe to your design.

1. Select **Tools** → **Probes** to display the Probes dialog box as shown in the following figure.
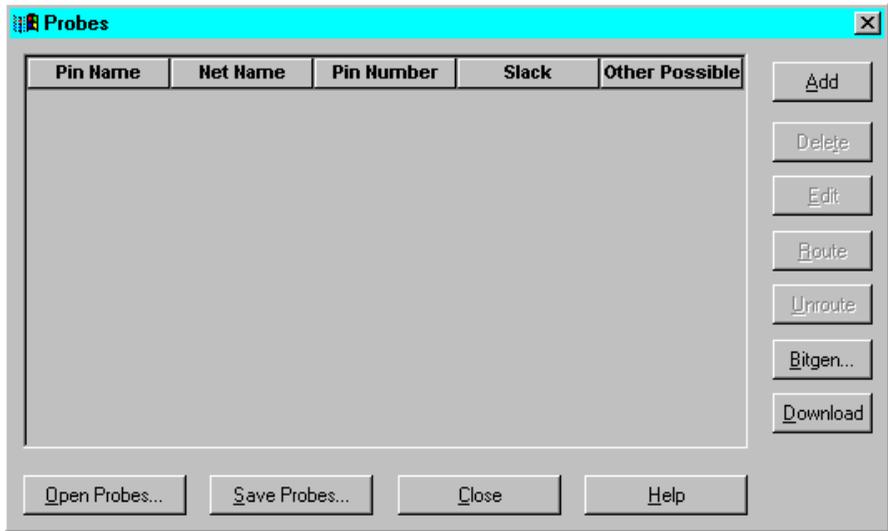
Figure 3-21 Probes Dialog Box

2. Select **Add** to display the Define Probe dialog box shown in the following figure.
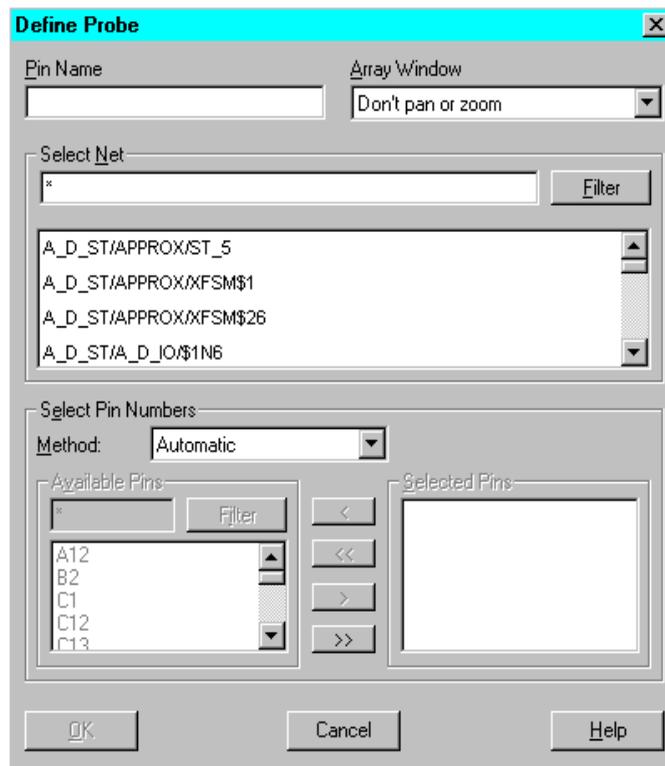
Figure 3-22 Define Probe Dialog Box

3.   Fill in the dialog box to define the probe. Click **Help** to get help on the fields.

> **Note** You can select probe pins automatically or manually in the Select Pin Numbers section of the Define Probe dialog box. The Manual option enables the additional options in this section to allow you to specify the pin numbers for the routed probe. The Automatic option disables the additional options. See the FPGA Editor online help for more information on this feature.

4.   Select **OK** to close the dialog box.

> **Note** To decrease the time it takes to route the probe, you can turn off the Delay Based Routing option in the Main Properties property sheet.

The probe is automatically routed, and the defined probe is displayed in the Probes dialog box.

5. Select **Bitgen** in the Probes dialog box to display the dialog box shown in the following figure.
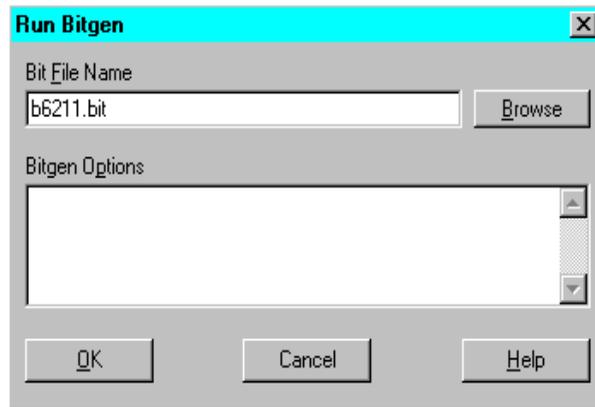
Figure 3-23 Run BitGen Dialog Box

6. The file name displayed in the Bit File Name field is the name of your probed design. If you change the name to *design_name*.bit, you are warned that this name represents your unprobed design. You can select **Yes** to save the bit file under this name or **No** to save the bit file under the name provided by the system.

7. Click **OK** to return to the Probes dialog box.

8. Select **Download** to start the Hardware Debugger tool so you can download your bit file to a test device. This tool is run in a separate process from the FPGA Editor.

Saving Probe Definition

Select **Save Probes** in the Probes dialog box to display the Save As dialog box. Enter a file name in the File Name field. Your probe definition is saved as a script file with a .scr extension (probes.scr is the default file name).

Opening a Probe Script File

Select **Open Probes** in the Probes dialog box to display the Open dialog box. Select the probe script file containing the commands that

define the probe. When the script is opened, the commands are performed, and the probe definition is added to the probes list.

Verifying Your Design

You can use the following FPGA Editor tools to verify your designs.

- Physical Design Rule Check (DRC)
- Delay Calculator

This section contains the following topics.

- "Physical Design Rule Check (DRC)"
- "Delay Calculator"
- "TRACE"

Physical Design Rule Check (DRC)

Physical Design Rule Check (DRC) is a series of tests to find logical and physical errors in your design. The DRC runs during these conditions.

- You select **Tools** → **DRC** → **Setup**
- You select **Tools** → **DRC** → **Run**
- You select the Run DRC toolbar button in the Block window
- During manual routing
- Whenever you add pins to a net

The DRC runs in the background. Results of the DRC are written into the History toolbar. DRC error messages indicate faulty or incomplete routing or component logic.

DRC runs can produce a large number of messages. Because the History toolbar accommodates a limited number of characters, you may not be able to scroll to messages at the beginning of a DRC run or to messages from previous runs. To view earlier messages, use a text editor to view the log file (*design_name*_fpga_editor.out) for your session. See the "Recovering a Terminated Session" section. When you exit the FPGA Editor, the log file is renamed *design_name*_fpga_editor_*YYMMDD_HHMMSS*.log, where Y is year, M is month, D is day, H is hour, M is minute, and S is second.

Note When you run the DRC on selected objects, the objects are deselected if the Automatic Deselect option in the Main Properties property sheet is enabled. If the option is disabled, the objects remain selected after the DRC.

The DRC tests follow these rules.

- If you select one or any combination of net names, routes, or net pins, the Net Check is performed.

- If you select one or more components, the Block Check is performed.

- If you select objects that combine the first and second types, both Net Check and Block Check are performed.

Running a DRC

To run a DRC, follow these steps.

1. Select the design objects for the DRC run.

   Select any combination of components, net pins, route segments, or nets. To check all design objects, do not select any objects.

2. Select **Tools** → **DRC** → **Setup** to display the DRC dialog box, as shown in the following figure.
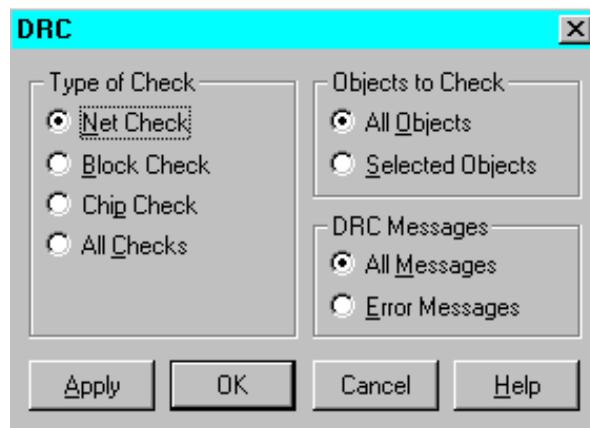
Figure 3-24 DRC Dialog Box

3. Select an option in the Type of Check field (you can only select one).

4. Select an option in the Objects to Check field to perform the DRC on all objects in your design or only on selected objects.

5. Select an option in the DRC Messages field to display all DRC messages, errors and warnings or only error messages.

6. Click **Apply** or **OK**

    The DRC tests are performed on the specified objects.

Delay Calculator

Delay is the time that it takes to propagate a signal from a driver pin to a load pin. If not otherwise indicated, delay values are given in nanoseconds (ns). The Delay Calculator tool calculates and displays the delay associated with load and driver pins in a given net or path.

Calculating Net Delay

You can either find the delay for all pins in the net or for specific pins.

To find the delay for all pins in a net, follow these steps.

1. Select a net name from the List window.

2. Select **Delay** in the User toolbar.

    If the net is fully or partially routed, a list of pins appears in the History toolbar, along with their associated delays. If the net is unrouted, each pin is listed as "unrouted."

3. Select **Attrib** in the User toolbar to display the Net Properties property sheet.

4. Select the Pins page to display the delays for associated pins.

    If a blank space appears next to a pin name, either in the pin list or in the History toolbar, the pin is the net driver that has no delay or the pin is unrouted. A tilde (~) appearing with a delay indicates that the value shown is approximate.

    Note When you display delays for all of the pins in a net, the net is deselected if the Automatic Deselect option in the Main Properties property sheet is enabled.

    To display the delays for selected pins in a net, use the mouse to select the specific pins. If the net is fully or partially routed, the

delays for the selected pins are automatically displayed in the History toolbar. If the net is unrouted, no delays are displayed.

Calculating Path Delay

To display the delay between two pins in a path follow these steps.

1. Select the two pins with the mouse.

2. Select **Delay** in the User toolbar.

    The path delay between the two pins is displayed in the History toolbar. If there are multiple paths between the two pins, the path with the maximum delay appears. If Automatic Hilite is enabled, the path between the pins is highlighted.

TRACE

You can run TRACE (Timing Reporter and Circuit Evaluator) in the FPGA Editor to visualize timing errors and to make modifications without going back and forth between the FPGA Editor and TRACE.

Functions performed by TRACE in the FPGA Editor are the same as those outside of the FPGA Editor with a few exceptions, as noted in this section.

TRACE functionality within the FPGA Editor includes the following.

*   Specify error or verbose report types

*   Limit the number of errors reported per constraint in the timing error and verbose reports. The summary report is not supported in the FPGA Editor

Input to TRACE is your currently loaded design file and the constraints file. Output is sent to the History toolbar, unless you specify otherwise, and to the OUT (log) file instead of to a timing report (TWR) file.

Running TRACE from the Tools Menu

To run TRACE from the Tools menu, follow these steps.

1. Select **Tools** → **Trace** → **Setup and Run** to display the dialog box shown in the following figure.
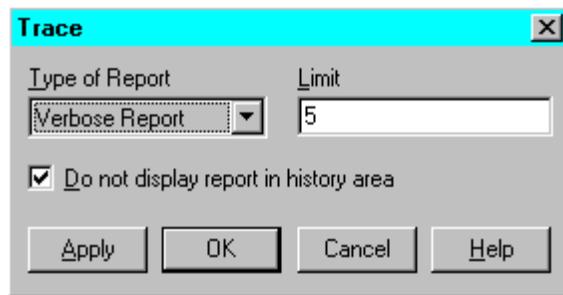
Figure 3-25 Trace Dialog Box

2. Fill in the dialog box and click **OK** or **Apply**

    TRACE runs timing analysis, and the Trace Summary dialog box
    appears as shown in the following figure. This dialog box lists
    information on any constraints that failed to meet timing.
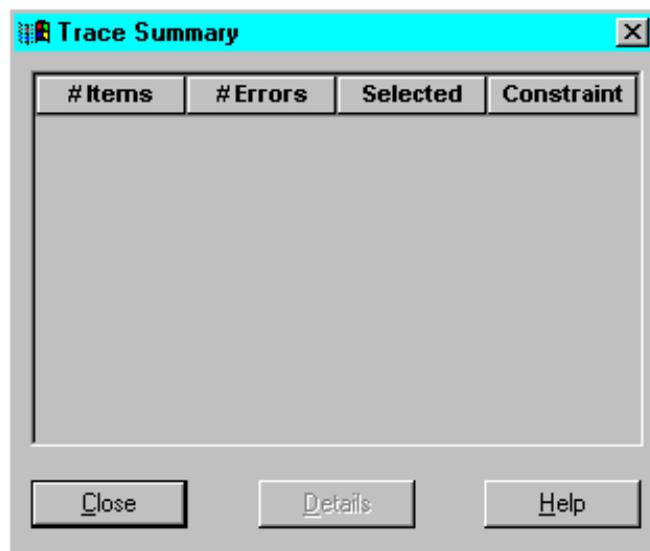
Figure 3-26 Trace Summary Dialog Box

3. For additional timing information on the constraints listed in the
    Trace Summary dialog box, click on the listed constraint and
    select **Details**. The TRACE Errors dialog box appears as shown
    in the following figure.
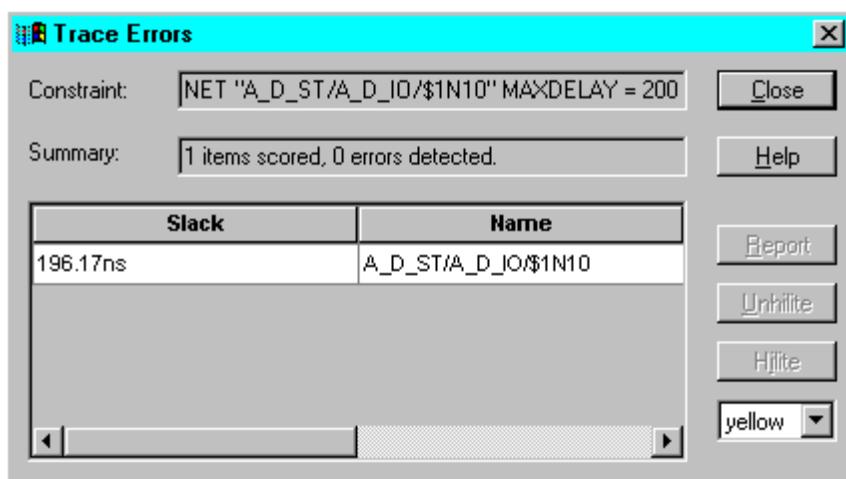
Figure 3-27 TRACE Errors Dialog Box

This dialog box contains the following fields.

Field	Description
Constraint	Displays the constraint (this field cannot be modified)
Summary	Displays a one line summary of the constraint (this field cannot be modified)
Slack/Name	Slack specifies the amount of margin between actual delay and specified constraint. Negative values indicate timing violations. Name specifies the start and end point for the path.
Report	Outputs the detailed TRACE report for the selected path to the history area and to the OUT (log) file
Unhilite	Unhighlights the selected path in the FPGA Editor window
Hilite	Highlights the selected path in the FPGA Editor window

Working with Physical Macros

A physical macro is a logical function created from components of a specific device family. Physical macros are stored in macro library files with a .nmc extension. In addition to components and nets, a macro can also contain placement and routing information. It can be unplaced, partially placed, fully placed, unrouted, partially routed, or fully routed.

You can create a new macro library file in the FPGA Editor or you can save an existing design as a macro. After creating a macro, you can instantiate it in your design. When you instantiate a macro, the contents of the macro library file are copied into your design file, and the link to the macro library file no longer exists. If you edit the macro library file, any new changes are not reflected in the instantiated macro in your design file. Refer to the following figure for an illustration of instantiating a macro.
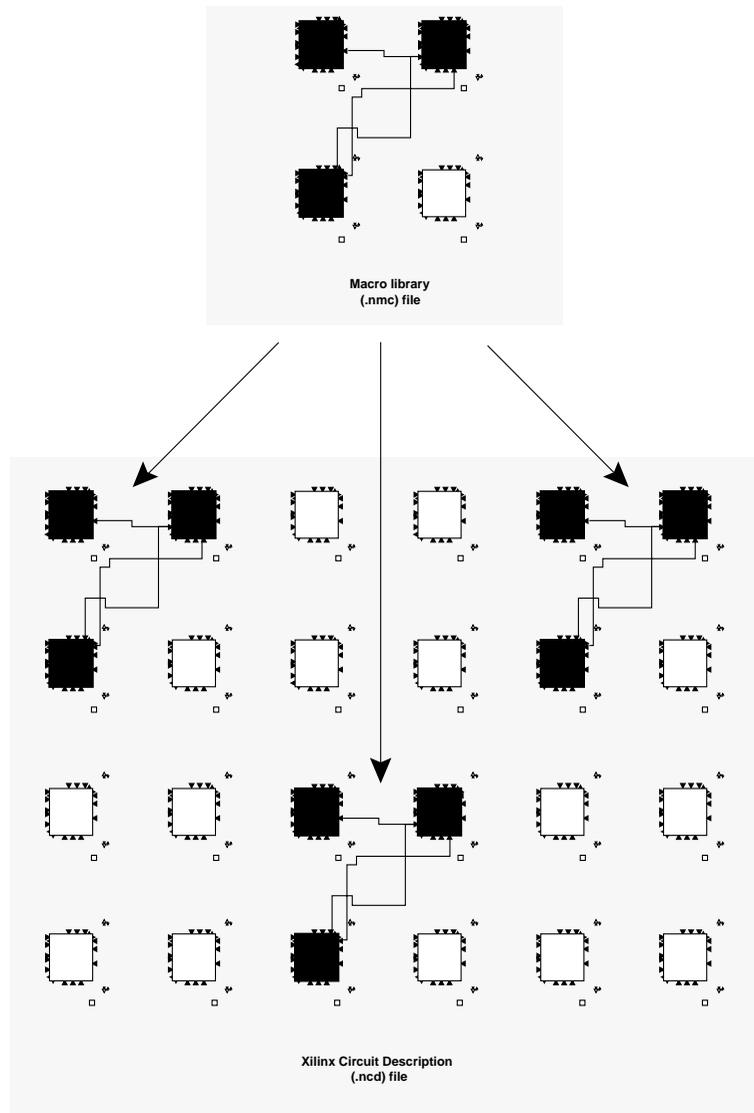
X7597

Figure 4-1 Instantiating a Macro in Your Design

You can also instantiate a macro into a schematic drawing by entering a block in the schematic, configuring the block appropriately, and placing a reference to the .nmc file in the schematic. You can instan-

tiate a macro in an HDL (Hardware Description Language) design by following the recommended instantiation method of the language.

This chapter describes creating, editing, and using macros in the FPGA Editor. It contains the following sections.

- "Macro Terminology"

- "Creating a New Macro File"

- "Opening an Existing Macro"

- "Saving your Design as a Macro File"

- "Saving a Macro Library File"

- "Saving a Macro Library File as a Design File"

- "Adding Macros to Your Design"

- "Operating on Macro Components and Nets in Your Design"

- "Selecting Macros in Your Design"

- "Deleting Macros from Your Design"

- "Viewing and Changing Macro Properties in Your Design"

- "Moving Macros in Your Design"

- "Unbinding Macros in Your Design"

- "Placing and Unplacing Macros in Your Design"

- "Routing and Unrouting Macros in Your Design"

- "External Macro Pins in Your Design"

- "Editing Your Macro File"

- "DRC Checks in Your Macro File"

Macro Terminology

The definitions in this section are specific to macros. See the following figure for an illustration of some of these terms.
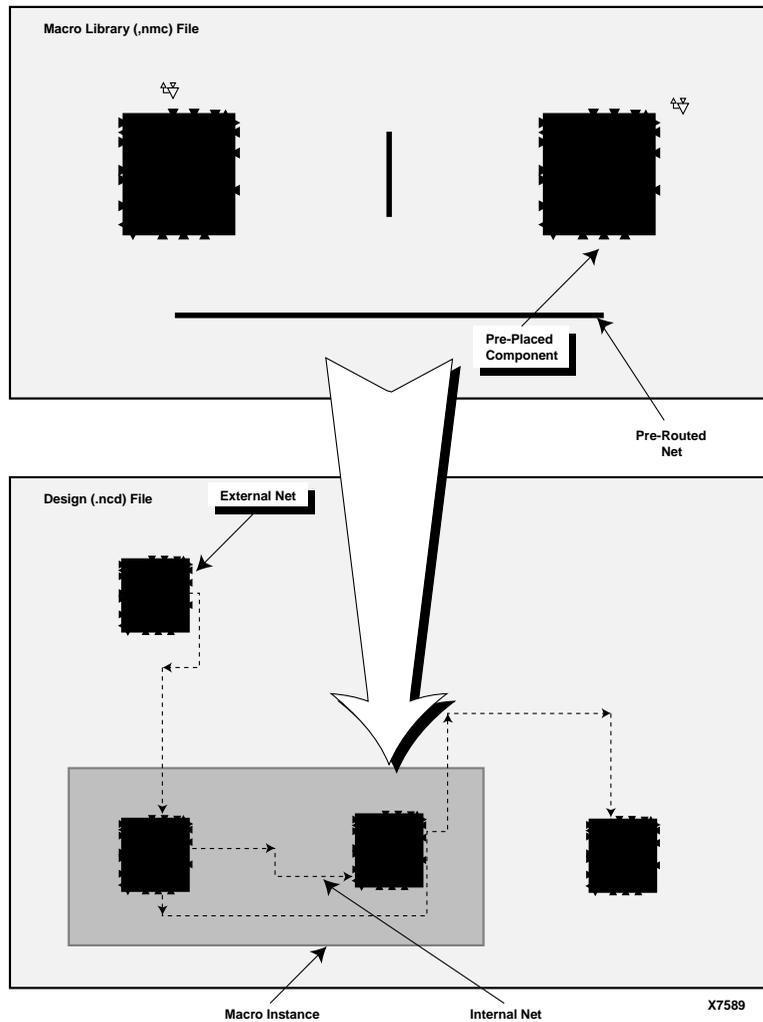
Figure 4-2 Macro Terminology

Macro Library File

A file containing the definition of a macro. Macro library files have a .nmc extension.

Macro Instance

A copy of a macro library file inserted in your design file. When you add a macro instance to your design, you "instantiate" the macro. Your design can contain multiple instances of the same library file, each with a unique instance name. Because the library file is copied into the design file when you instantiate a macro, if you then edit the library file, the changes are not reflected in the macro instantiated in your design file.

Pre-placed Component

A component that is placed in the macro library file.

Pre-routed Net

A net that is completely routed in the macro library file.

Note When you instantiate a pre-routed macro in your design, the design can take much longer time to place and route, depending on the routing complexity. Xilinx recommends allowing PAR to route the macro with your design.

Reference Component

A component in the macro library file used as a reference when a macro instance is placed, moved, or copied. Placement and routing of all other pre-placed macro components are determined relative to this component. If at least one of the macro's components is pre-placed, the macro will have a reference component. If none of the macro components are pre-placed, the macro will not have a reference component.

External Pin

A macro pin used to connect the components in an instantiated macro to other components in your design (outside of the macro).

Internal Net

A net in the macro library file that does not have a connection to any of the macro's external pins.

External Net

A net outside of a macro instance connected to one of the macro's external pins. Part of an external net can lie within the macro if the macro library file contains a net that is connected to an external pin.

Unbind

Use this command to disassociate a macro's components and nets from the macro. When you unbind a macro, the macro name is removed from your design's database. All of the components and nets formerly in the macro are then treated as separate components and nets.

Creating a New Macro File

Note See the "From the Command Line" section of the "Getting Started" chapter for the command line options for creating a new macro.

To create a new macro in the FPGA Editor, follow these steps.

1. Select **File** → **New** or click on the New toolbar button.

2. Select the **Macro** option.

3. Enter the macro name in the Macro File field. If you do not enter the .nmc extension, it is automatically added when the file is created.

    You can open a new macro under a directory other than the current working directory. Type in the path name of the target directory in the Macro File field, or click **Browse** to select your target directory and specify your macro file name.

4. Click **Select Part** to display the Part Selector dialog box, shown in Figure 3-2 of the "Using the FPGA Editor" chapter.

Note Although you must specify a part number and speed, a macro can be instantiated in any design file of the same family. For example, if you create a macro library file and specify a 4008EPG191 package at a speed of 4, the macro can still be instantiated in a design file for a 4010EPQ208 package at a speed of 3, or any other Xilinx XC4000E family design file.

5. Select a Family, Device, Package, and Speed Grade for your macro.

    Note You can only select a part number from a part library you have installed on your system.

6. Click **OK**. The Part field in the New dialog box is filled in with your selections.

7. Click **OK** to close the New dialog box.

    An empty (unprogrammed) macro is loaded into the FPGA Editor window with the part number and speed as specified.

Opening an Existing Macro

Note See the "From the Command Line" section of the "Getting Started" chapter for the command line options for opening an existing macro.

To open an existing macro file while you are in the FPGA Editor window follow these steps.

1. Select **File** → **Open** or click on the Open toolbar button.

    The Open dialog box appears, as shown in Figure 3-3 of the "Using the FPGA Editor" chapter.

2. Select the **Macro** option.

3. Enter the name of the .nmc file you want to open in the Macro File field, or click **Browse** to specify the macro file name and directory.

4. Select an Edit Mode for the macro from the pull-down list box.

    Select Read Only, No Logic Changes, or Read Write.

5. Click **OK**.

    The macro file you specified is loaded in the FPGA Editor window.

Saving your Design as a Macro File

You can create a macro library file by saving your design as a macro.

Note Constraints are not retained when you save your design file as a macro library file. A constraints file is not generated, even if your design had constraints defined.

Use the following steps to save your design as a macro library file.

1. Select **File** → **Save As** to display the Save As dialog box, shown in Figure 3-4 of the "Using the FPGA Editor" chapter.

2. Select the **Macro** option.

3. Enter the macro name in the Macro File field with a .nmc extension.

    You can save your macro in a directory other than the current working directory. Type in the path name of the target directory in the Macro File field, or click **Browse** to select your target directory and specify your macro file name.

4. Click **OK**.

    The dialog box closes, and the macro file is saved with the name you specified. The file name in the Title Bar at the top of the FPGA Editor window changes to indicate that this new file is currently displayed in the window. When you later save or exit the file, it is saved to the new file name.

Saving a Macro Library File

When you save a macro library file, you cannot save a constraints file along with the NMC file, even if constraints were created when the macro library file was edited. All constraints are lost when you save the NMC file.

With the Same Name

To save your current macro library file as a macro file with the same name, select **File → Save**.

With a Different Name

To save your macro library file as a macro library file with a different name, use this procedure.

1. Select **File → Save As** to display the Save As dialog box, shown in Figure 3-4 of the "Using the FPGA Editor" chapter.

2. Select the **Macro** option.

3. Enter the macro name in the Macro File field with a .nmc extension.

   You can save your macro in a directory other than the current working directory. Type in the path name of the target directory in the Macro File field, or click **Browse** to select your target directory and specify your macro file name.

4. Click **OK**.

   The dialog box closes, and the macro file is saved with the name you specified. The file name in the Title Bar at the top of the FPGA Editor window changes to indicate that this new file is currently displayed in the window. When you later save or exit the file, it is saved to the new file name.

Saving a Macro Library File as a Design File

To save your macro library file as a design file, use this procedure.

1. Select **File → Save As** to display the Save As dialog box, shown in Figure 3-4 of the "Using the FPGA Editor" chapter.

2. Select the **Design** option.

3. Enter the design name in the Design File field. If you do not enter the .ncd extension, it is automatically added to the file name.

   You can save your design in a directory other than the current working directory. Type in the path name of the target directory in the Design File field, or click **Browse** to select your target directory and specify your design file name.

4. If applicable, enter the constraints file name for the design in the Physical Constraints File field, or click **Browse** to select a directory and specify a file name.

5. Click **OK**.

   The dialog box closes, and the design file is saved with the name you specified. Any constraints in the macro library file are written to the specified constraints file. The file name in the Title Bar at the top of the FPGA Editor window changes to indicate that this new file is currently displayed in the window. When you later save or exit the file, it is saved to the new file name.

Adding Macros to Your Design

When you add a macro, you are instantiating a macro or adding an instance of a library macro file to your design. A design can contain multiple instances of the same macro library file, however, each instance has a unique name. A macro can only be added if the design file and the macro library file are the same family. For example, if your target device is an XC4000E, you can only add an XC4000E macro library file.

The first time you instantiate a macro library file, a copy of the library file is placed into your design file. Any subsequent instantiations of the same library file into your design file refers to this local copy, and are not affected by changes in the external library file that was used for the original instantiation.

To add a macro to your design, follow this procedure.

1. If you want to place the macro when you add it, select a vacant site for the macro.

   The site must match the reference component specified in the macro library file. For example, if the macro library file specifies an I/O component as the reference component, the selected site must be an I/O component site.

   If there is no reference component in the macro library file, the selected site is ignored and the macro is unplaced when it is instantiated. If you do not select a site, the macro is unplaced when it is instantiated.

2. Select **Edit** → **Add Macro** to display the Add Macro dialog box.
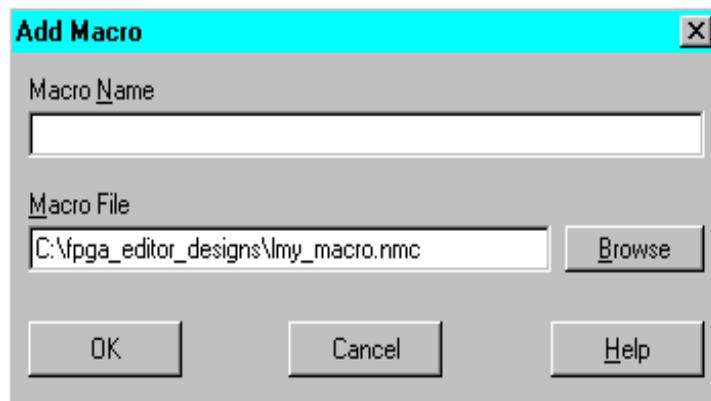
Figure 4-3 Add Macro Dialog Box

3. Fill in the dialog box fields and click **OK**. The fields in this dialog box are described in detail in the FPGA Editor online help.

    An instance of the macro library file is added to your design. If you selected a site, the macro's reference component is placed at the specified site, and any other placed components in the macro library file are placed relative to the reference component.

    When you close the Add Macro dialog box, a Macro Properties property sheet may appear for the newly created macro if the Automatic Post option is enabled in the Main Properties property sheet. You can edit the property sheet to modify macro properties or click **Cancel** to close the property sheet.

Notes on Adding Macros

This section includes additional information on adding macros to your design.

*   If you do not specify a macro name when you add a macro, the name assigned to the new macro is in the format: *$MACROINST_number*, where *MACROINST* is the name of the macro library file that defined the macro when it was added. *Number* is a number assigned to that instance of the macro library file. Numbering starts at 0, and increases by one for each new added macro. For example, if you add multiple instances of a macro library file named "adder" the first instance is $adder_0, the second is $adder_1, and so on.

- Any characters special to the FPGA Editor command interpreter must be preceded by a backslash (\) escape character when used in a macro name. Special characters are quotation marks (" or ') * ; ? # - (leading dash). The restriction only applies to commands entered at the command line or those in an FPGA Editor command script file. The special characters can be entered in the Macro Properties property sheet without the escape character. If you use another vendor's tool set in conjunction with the Xilinx tools, they may have other naming restrictions.

- When a macro is added, the names of all components, nets, and external pins within the macro has this format: *macro_name/ object_name*, for example, $adder_3/$comp_0 or $decoder_2/ $extpin_0.

- All components, nets, and macros have their own name space and must have unique names within this name space. For example, you can have a component and a macro named "FRED," but you cannot have two macros named "FRED" or two nets named "FRED." The FPGA Editor prevents you from entering a duplicate name in a name space. Also, the name cannot be empty.

- If the original (external) macro library file is changed or deleted, there is no effect on any instances of this library file in your design.

- If you change the external macro library file and want to update an existing design to use this revised library file, you can do one of the following.

    ♦ Remap your design

    ♦ Delete all instances of the revised macro library file, and then add new instances of the revised macro library file in place of the old instances.

Operating on Macro Components and Nets in Your Design

Normally you place, route, unplace, and unroute a macro instance as a whole object instead of performing these operations on the individual components and nets that comprise the macro. If a macro is instantiated from a completely placed and routed library file, these

operations are always performed on the entire macro. However, because it is possible to define a macro that is not completely placed and routed, you must place and route those individual components and nets that do not have predefined placement and routing information. The following rules govern a macro's internal components, nets, and pins.

- Components within a macro instance cannot be added, deleted, reconfigured, or renamed.

- Pre-placed components that are placed in the macro definition file cannot be manually or automatically placed, unplaced, or swapped in the macro instance, except by placing, unplacing, or swapping the entire macro instance.

- Components that are not pre-placed can be placed, unplaced, and swapped in the macro instance.

- Internal nets in a macro instance cannot be deleted or added. Pins cannot be added to or deleted from internal nets.

- Pre-routed nets that are routed in the macro library file cannot be manually or automatically routed or unrouted in the macro instance, except by routing the entire macro instance.

- Macro nets that are not pre-routed can be routed and unrouted.

- Unused pins on macro components cannot be used to create non-macro nets. Only external pins can be connected to external nets.

Selecting Macros in Your Design

To select a macro in your design, display a list of the appropriate macros (all, placed, or unplaced) in the List window, and then select the desired macro name. If you select the name of a placed macro, the components and nets making up the macro change color in the Array window.

Deleting Macros from Your Design

When you delete a macro, all of the macro's external pins are unrouted, all components and nets that are part of the macro are removed, and the macro name is deleted from the database.

To delete a macro, follow this procedure.

1. Display a list of macro names in the List window.

2. Select the macros to delete.

3. Select **Edit** → **Cut**

    The macros are eliminated from the database.

Viewing and Changing Macro Properties in Your Design

To view macro properties, follow this procedure.

1. In the List window, display a list of macro names.

2. Select a macro from this list.

3. Select **Edit** → **Properties of Selected Items** to display the Macro Properties property sheet shown in the following figure. This property sheet contains the General and Physical Constraints pages.
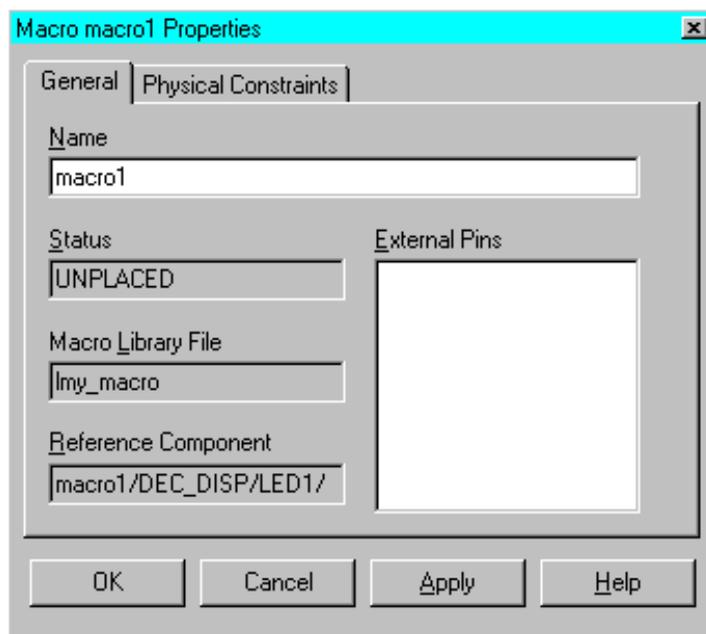
Figure 4-4 Macro Properties Property Sheet

General Page

Refer to the following table for a description of the General page.

Property	Description
Name	Name of the macro. If you change the macro name, all of the names of the components, nets, and external pins that are part of the macro are updated automatically to reflect this change.
Status	Indicates whether the macro is unplaced or placed. A macro is only shown as placed if all the components in the macro are placed.
Macro Library File	Name of the macro library file that defined this macro instance.
Reference Component	Indicates the component that is designated as the reference component. A macro without pre-placed components does not have a reference component; a macro with at least one pre-placed component has one.
External Pins	Lists the pins that are defined as external pins.

Physical Constraints Page

This page displays the constraints that are applied to the macro. Refer to the following table for a description of the Physical Constraints page.

Property	Description
Lock Placement	If enabled, indicates a lock macro or global lock placement constraint is applied. If a macro is locked, all macro components that were pre-placed are locked. The macro cannot be unplaced.
Requirement	Specifies whether the value in the Location Range field is an absolute requirement (hard) or not (soft) for the placement tools.
Effort	Sets up a priority for the autoplacer to resolve constraint conflicts.

Property	Description
Location Range	Places the macro's reference component at a specified site, places all of the macro's pre-placed components (that is, all components that were placed in the macro's library file) in sites relative to the reference component, and locks all of these placed components at their sites.
Block Paths	Use this timing constraint to block the enumeration of all timing paths that go through this macro
TSid	Assigns a timing period or frequency to a timing specification

Moving Macros in Your Design

This section describes two methods for moving macros in your design.

Moving Macros with the Swap Command

To move a macro with placed components to an unused group of sites, use the Swap command. When you move a placed macro by swapping, you save a step compared to unplacing the macro then placing it at the new location.

To move a macro to an unused group of sites, follow this procedure.

1. Select the macro to move.

2. Press the Control key and the left mouse button to select an unused site for the macro's reference component.

    The reference component and the selected site must be the same type of block, that is, I/O block and I/O block, or logic block and logic block.

3. Select **Edit** → **Swap**.

    The reference component moves to the unused site, and the other components in the macro move to maintain their positions relative to the reference component.

    If the Automatic Routing option is enabled, the macro's external pins are routed after the macro is placed. Also, any unrouted macro nets are automatically routed.

Moving Macros using Drag and Drop

You can use the drag and drop method to move *placed* macros from one location to another within the Array window. You can also drag and drop *unplaced* macros from the List window to an unused site in the Array window.

To move a macro with placed components to an unused group of sites using the drag and drop method, select the macro with the right mouse button and drag it to an unused site for the reference component. To move an unplaced macro from the List window, select the macro with the left mouse button and drag it to an unused site for the reference component. The reference component and the selected site must be the same type of block, that is, I/O block and I/O block, or logic block and logic block. The macro's reference component moves to the unused site, and the other components in the macro move to maintain their positions relative to the reference component. If the Automatic Routing option is enabled, the macro's external pins are routed after the macro is placed. Also, any unrouted macro nets are automatically routed.

Unbinding Macros in Your Design

When you unbind a macro, the components and nets are disassociated from the macro. The macro name is deleted from the design's database. After unbinding, all of the components and nets are treated as separate components and nets. Also, component names and net names are changed, replacing the forward slash (/) that indicates the macro's library file with an underscore (_).

To unbind a macro, follow these steps.

1. Display a list of macro names in the List window.

2. Select the macros to unbind.

3. Select **Edit** → **Unbind Macro**.

    For each of the selected macros, the included components and nets no longer are associated with the macro, and the macro name disappears from the List window.

Placing and Unplacing Macros in Your Design

Macros can be placed in the Array window either automatically or manually. When you use automatic placement (AutoPlace), the system selects an appropriate site or sites for each macro you select. When you use manual placement, you select both a macro and a site in which to place the macro's reference component.

When a macro is manually placed, only the pre-placed components in the macro are placed (those components that were placed in the macro library file defining this macro instance). Any unplaced components in the macro's library file must be placed in a separate operation. When a macro is automatically placed, the system tries to place all of the macro components, whether they are pre-placed or not.

Macros can also be unplaced. Unplacing a macro unroutes each of the macro's external pins, removes the macro from the Array window, and adds the macro to the list of the design's unplaced macros.

AutoPlacing Macros

Use the following steps to place macros automatically with the Auto Place command.

1. Display a list of unplaced macros in the List window.

2. Select the macros to automatically place.

3. Select **Tools** $\rightarrow$ **Place** $\rightarrow$ **Auto Place**.

    The selected macros are automatically placed in vacant sites in the Array window. For each macro, any connections that were unrouted in the macro's library file are routed when the macro is placed.

    If the Automatic Routing option is enabled, each macro's external pins are routed after the macro is placed. Also, any unrouted macro nets are automatically routed.

Manually Placing Macros

Use the following steps to manually place an unplaced macro.

1. Display a list of unplaced macros in the List window.

2. Select a name from the list of unplaced macros.

3. Press the Control key and the left mouse button and select a vacant site in the Array window in which to place the macro reference component.

    The reference component and the selected site must be the same type of block (I/O component and I/O component or logic component and logic component).

4. Select **Tools → Place → Manual Place**.

    The reference component is placed in the selected site, and all of the macro's pre-placed components (components placed in the macro's library file) are placed in sites relative to the reference component. Macro components that are not pre-placed must be placed in a separate operation.

    Any connections that were routed in the macro's library file are routed when the macro is placed. If the Automatic Routing option is enabled, the macro's external pins are routed after the macro is placed. Also, any unrouted macro nets are automatically routed.

Unplacing

The Unplace command removes selected macros from their current sites and puts them into the unplaced macros list. The list of unplaced macros can be viewed in the List window. Unplaced macros can be re-placed. Before unplacing a macro, the system unroutes each external pin on the macro.

Note Locked macros cannot be unplaced.

To unplace selected macros, follow this procedure.

1. Select the macros to unplace in the List window.

2. Select **Tools → Place → Unplace**.

    All of the selected macros are unplaced.

Routing and Unrouting Macros in Your Design

Routing and unrouting macros is similar to routing and unrouting components. These operations are described in the "Routing and Unrouting" section of the "Using the FPGA Editor" chapter.

- To select a macro for automatic routing, display a list of placed macros in the List window, and then select the name of the desired macro.

- When a macro is placed, any pre-routed connections are automatically routed after the macro is placed. You cannot unroute these connections in your design.

- When a macro is placed without the Automatic Routing option enabled, any internal connections that are unrouted in the macro's library file are unrouted after the macro is placed. You must perform a separate route operation to route these connections.

- When a macro is automatically routed, each external pin connected to a net is routed. Also, any nets that are not pre-routed are automatically routed.

External Macro Pins in Your Design

External macro pins are similar to other pins in your design. You can add them to nets in your design, route and unroute them (if external to the macro), and find them with the Find command. However, you cannot add or delete external pins in your design, and you cannot change the name of an external pin.

Editing Your Macro File

This section describes how to edit a macro library file. You can perform the following editing operations.

- Add, delete, list, and view properties of external pins

- Designate a macro reference component

- Add a macro to the library file

- Place and route macros

- Perform a macro DRC

The FPGA Editor Edit menu includes the following macro commands.

Table 4-1 Edit Menu Macro Commands

Command	Description
Add Macro	Use this command to add a macro to your design
Add Macro External Pin	Use this command to define a selected component pin as an external macro pin
Set Macro Reference Comp	Use this command to designate a selected component in a macro file as the macro's reference component
Unbind Macro	Use this command to separate a selected macro into its individual components

Adding External Pins

External pins connect the instantiated macro to other components in your design. The external pins are defined in the macro library file, and the pins cannot be modified in an instantiated macro. External pins are displayed in the same color as the macro layer in the Array window.

To add external pins to a macro library file, follow these steps.

Note If you are adding a number of external pins, you may want to create a User toolbar button to make this operation easier.

1. Open a macro file in the Array window.

2. Select the pins that you want to define as external pins.

3. Select `Edit` → `Add Macro External Pin`.

    The selected pins are now external pins and the color changes to the color used for the macro layer.

When you add external pins, Figure 4-7 (External Pin Property Sheet) is displayed for each newly created external pin if the Automatic Post option is enabled in the Main Properties property sheet. Each property sheet displays the name of the new pin and additional pin information. You can edit the property sheet to modify external pin properties, or you can click `Cancel` to close the property sheet.

When you add an external pin, the name assigned to the new pin is in the format *$extpin_number*, where *number* is a number assigned to each new pin. Numbering starts at **0**, and increases by one for each new pin. Optionally, you can change the assigned pin name in the External Pin Properties property sheet.

Macro to External Net Recommendations

For macro nets connected to external nets, use the recommendations in this section to declare and name external pins, and to set their Type attributes.

- If a macro net is an output to the external circuitry (that is, one or more pins on the net are drivers for the external net), and the macro net has both input and output pins, declare one output driver pin as external. If there are multiple output pins on the net, it does not matter which one you select.

- If a macro net is an input to the external circuitry (that is, one or more pins on the net act as loads for the external net), and the net has both input and output pins, declare one input pin as external. If there are multiple input pins on the net, it does not matter which one you select.

- If you use the macro library file in a schematic, you must give the external pins in the macro library file the same names as the pins on the symbol used to instantiate the macro in the schematic.

  Note If your schematic symbol contains a multi-bit pin and your schematic is read into the Xilinx tools as an XNF file, you must name the corresponding macro external pins in the format: *pin_name number* (for example, DATA<0> or A<2>). When you name these external pins in the macro, you must insert the angle brackets (< and >) to denote the bit numbers for these pins.

- To ensure timing analysis works correctly when the macro is instantiated, you may have to set each external pin's Type property when you add the external pin to the library file. For most cases, the default Type set by the system is adequate. External pin attributes are described in the "Viewing and Changing Macro Properties in Your Design" section.

The following examples illustrate these recommendations.

Example 1

The following figure shows a macro that connects an external clock pin to multiple clock input pins. Only one K pin needs to be designated as the external clock pin to the macro.
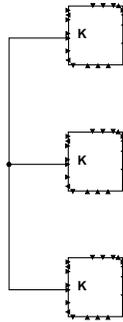
X7590

Figure 4-5 External Pins Example 1

Example 2

The following figure shows a macro that functions as a latch. External pins appear as filled-in triangles. There are two input external pins and one output external pin for the macro. The output external pin (Y) also feeds back to an input pin (the F1 pin) within the macro, but only the Y pin is used as an external pin. Y is a driver to the external circuitry, and should be designated as an external pin.

LOGICAL FUNCTION

PHYSICAL IMPLEMENTATION (MACRO)
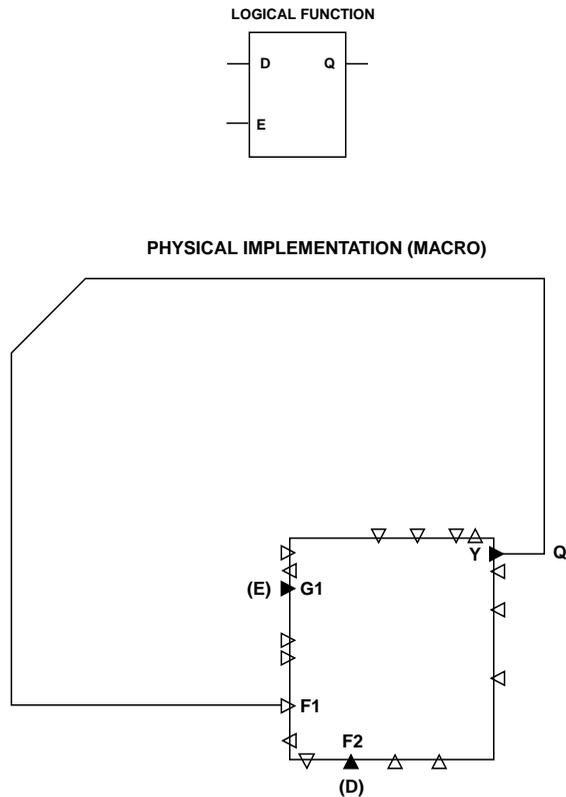
X7591

Figure 4-6 External Pins Example 2

Viewing and Changing External Pin Properties

To view or change external pin properties while editing a macro, follow these steps.

1. Select an external pin.

2. Select **Edit** → **Properties of Selected Items** to display the External Properties property sheet, as shown in the following figure.
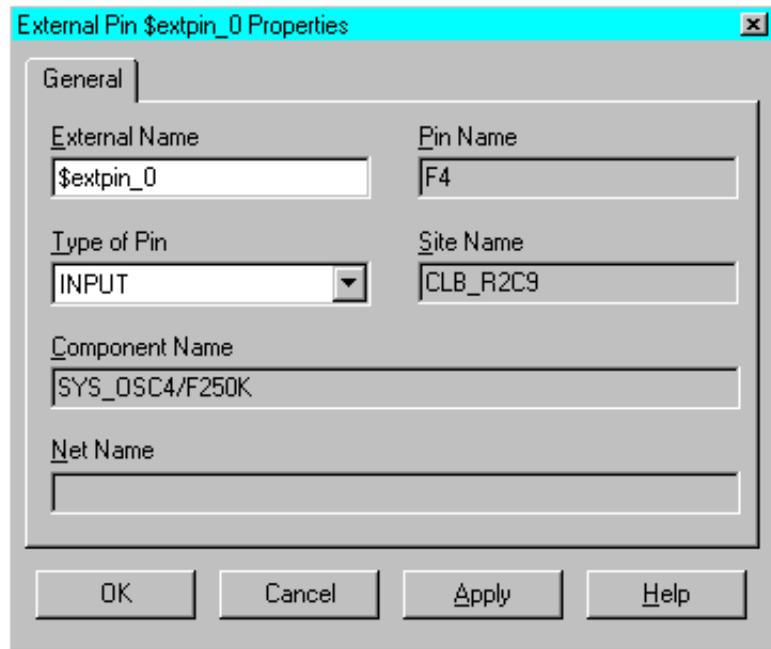
Figure 4-7 External Pin Property Sheet

Refer to the following table for a description of the properties in the External Pin Property sheet.

Property	Description
External Name	Name of this pin when the macro library file is instantiated in your design
Pin Name	Name of component or site pin
Type of Pin	Specifies the function of the pin within the macro. The pin type determines how paths passing through the pin are defined for timing analysis. Pin types are described in the following tables.
Site Name	Site in which the component containing the pin is placed
Component Name	Name of the component containing the pin
Net Name	Name of the net connected to the pin

For an input pin, the pin types are the following.

Type	Description
INPUT	Pin is an input to combinational logic in the macro; it is not clocked
CLOCK	Pin is used as a clock signal within the macro
SRDIRECT	Pin is used as an asynchronous Set or Reset signal within the macro
ENABLE	Pin is used as an asynchronous control line (for example, a tristate buffer enable) within the macro
SYNCINPUT	Pin is an input that is clocked within the macro (for example, a register input)
SYNCENABLE	Pin is used as a control line synchronous to a clock within the macro
SETDIRECT	Pin is used as an asynchronous Set signal within the macro.
RSTDIRECT	Pin is used as an asynchronous Reset signal within the macro.

For an output pin, the pin types are the following.

Type	Description
OUTPUT	Pin is an output from combinational logic within the macro; it is not clocked within the macro
SYNCOUTPUT	Pin is an output from clocked logic within the macro. The pin is synchronous with respect to the macro clock.
DIRECTIN	Pin should be treated as a direct input pin, that is, a pin with no internal delay. This pin type *should not* be specified for an external macro pin.

Deleting External Pins

When you delete an external pin, the pin is no longer defined as external. If the external pin is connected to a net, it remains connected after you delete it as an external pin.

To delete external pins, follow this procedure.

1. Select the pins that you want to remove as external pins.

2. Select **Edit** → **Cut**.

   The selected pins are no longer external pins. If you display a list of external pins, these pins do not appear in the list.

Listing External Pins

To list all currently defined external pins, enter the **getattr main extpins** command in the Command Line toolbar. A list of all external pins currently defined for the macro appears in the History toolbar and is written to the log file.

Designating a Reference Component

A reference component is a component in the macro library file used as the origin when a macro instance is placed, moved, or copied. Placement and routing of all other macro components are determined relative to this component.

If a macro library file has at least one pre-placed component, the macro will have a reference component. A macro can have only one reference component. You can specify the reference component when you create a macro library file, or you can have the system assign one automatically. If you do not designate a reference component, the component in the lower-left corner is selected by default when the macro file is saved.

When you place, move, or copy a macro instance, select a site, and then execute the appropriate command, the macro's reference component is placed in the site you designated. All other placed components in the macro are located in sites relative to the reference component.

To designate a reference component, follow this procedure.

1. Select the component you want as the reference component.

2. Select **Edit** → **Set Macro Reference Comp**.

   To determine which component is the reference component, enter **getattr main refcomp** in the Command Line toolbar. The name of the reference component appears in the History toolbar.

You can define a macro with unplaced components and unrouted nets. In this case, there is no reference component, and you cannot place an instance of this library file except by placing each macro component individually.

Adding a Macro to a Macro File

You can add a macro instance to a macro file. Use the procedure described in the "Adding Macros to Your Design" section. When you add a macro instance to a macro file, the macro instance is unbound; the components and nets in the instance are added as separate components and nets, not as macro elements. After instantiation, there is no indication that these components or nets were added as a macro.

Macro File Placement and Routing

You can place components and route the connecting nets in your macro file. When you later instantiate the macro library file in your design, the placement of the components relative to each other is maintained in the instantiated macro. Also, routing in the instantiated macro follows the routing specified in the library file. Placement and routing operations in a macro file are identical to these same operations in a design file.

When you instantiate a pre-routed macro into your design, the design can take a substantially longer time to place and route, depending on the route complexity and the number of connections in the macro nets. Xilinx recommends that you do not add any unnecessary routing to your library file.

Do not define a macro with CLB K pins routed together using long lines. This configuration may be impossible to route when the macro is instantiated into your design.

DRC Checks in Your Macro File

There are a number of physical DRC checks for testing your macro file that can be run in the FPGA Editor. These tests also run automatically whenever you save your macro library file.

To run the macro DRC check, follow these steps.

1. Make sure nothing is selected in the Array window or in the List window. If necessary, select **Clear** in the User toolbar to deselect any objects.

2. Select **Tools** → **DRC** → **Run**.

   The macro DRC is performed (along with other DRC tests). DRC results appear in the History toolbar.

Chapter 5

Customizing the FPGA Editor

You can change many of the FPGA Editor default settings to customize the interface to suit your needs. This chapter includes the following sections.

- "Initializing the FPGA Editor"

- "Customizing the User Toolbar"

- "Customizing Mouse Buttons"

- "Customizing Colors"

- "Customizing Fonts (PC only)"

- "Customizing with Command Aliases"

- "Customizing with Command Scripts"

Initializing the FPGA Editor

The fpga_editor.ini initialization file is installed in $XILINX/data when the FPGA Editor is installed on your system. This file is a text file containing a series of commands that are performed when the FPGA Editor is invoked. You can use any text editor to edit this file. Any of the FPGA Editor command line commands described in the online help can be added to the fpga_editor.ini file.

If you want to customize the fpga_editor.ini file to suit your needs, copy it to your home directory, edit it, and rename it fpga_editor_user.ini. When the FPGA Editor is launched, it reads the fpga_editor.ini file before the fpga_editor_user.ini file. You can also copy the fpga_editor.ini file to your working directory and edit it for your own use.

When the FPGA Editor is invoked, it automatically looks for the fpga_editor.ini file in the following order.

- Current working directory

- Your home directory

- $XILINX/data (location of default fpga_editor.ini file)

Customizing the User Toolbar

You can customize the User toolbar to include frequently used commands. You can either edit the fpga_editor.ini (or fpga_editor_user.ini) file before you start the FPGA Editor, or make changes to the User toolbar during an editing session.

Before Starting the FPGA Editor

When the FPGA Editor is invoked, it reads the fpga_editor.ini file. One function of the fpga_editor.ini file is to set the User toolbar commands that are displayed when the FPGA Editor window opens. You can change the default configuration by editing the fpga_editor.ini or fpga_editor_user.ini file.

During an Editing Session

You can also add or delete commands to the User toolbar during an editing session with the Button and Unbutton commands, respectively. Any command changes made during an editing session are not saved for later sessions. The Button and Unbutton commands are described in the FPGA Editor online help.

Customizing Mouse Buttons

You can program the mouse buttons to run commands from the command line. To customize the mouse buttons, use the Alias command in the fpga_editor.ini or fpga_editor_user.ini file, as shown in the following section of the fpga_editor.ini file. For more information on the Alias command, see the FPGA Editor online help.

```
alias [LeftClick] "unselect –all;
pick –order pin:etc"
   alias [Control+LeftClick] "pick –order pin:etc"
```

Mouse button commands used in the fpga_editor.ini file are shown in the following table. If you *unalias* a programmed mouse button, the FPGA Editor restores the default command to that button.

Mouse Button	Command Mapping
Left Click	unselect –all; pick –order *order_spec*
Control + Left Click	pick –order *order_spec*
Shift + Left Click	pick –k –order *order_spec*
Control + Shift + Left Click	pick –order *order_spec*
Control + Right Click	zoom in @cursor
Shift + Control + Right Click	zoom out @cursor
Left Double Click	post block

Customizing Colors

Areas in the FPGA Editor window (such as the User toolbar or the History toolbar) are automatically assigned colors when the FPGA Editor window opens. These colors cannot be modified after an FPGA Editor session begins.

Object layers in the Array window (for example, components, long lines, or selected objects) are also assigned colors when the FPGA Editor window opens. You can edit the fpga_editor.ini file to customize the layer colors that appear when the FPGA Editor is invoked. You can also use the Setattr command during an editing session to change the default colors. Any color changes made during an editing session are not saved for later sessions. You can use the Devcolor command to specify different colors for routing resources based on the names of the resources. These three procedures are described in the "Changing Object Colors on Workstation and PC"section. Also, refer to the FPGA Editor online help for a complete description of the Setattr and Devcolor commands.

Changing Window Colors on the PC

On the PC, the FPGA Editor windows use the standard system colors. You can change these colors by selecting the Display icon in the

Control Panel. The default colors for some of the areas in the FPGA Editor window are shown in the following table.

PC Window Area	Color
Background of Array window	Black
Background of World window	Black
Background of List window	White
Background of Menu bar	Standard System Colors
Background of History toolbar	Standard System Colors
Background of User toolbar	Standard System Colors
Background of Command Line toolbar	Standard System Colors
Background of Status bar	Standard System Colors

Changing Object Colors on Workstation and PC

On a workstation or a PC, you can control the colors of layers of objects (such as sites, components, switchboxes) in the Array window. You can change the colors assigned when the FPGA Editor window opens, and also change colors during an editing session. When you change layer colors during an editing session, the changed colors are only valid for the current session.

When the FPGA Editor window opens, the system assigns default colors to object layers. To change these default colors, you must edit the fpga_editor.ini file. To change colors during an editing session, use the Command Line toolbar. Both of these procedures are described in this section.

Note You can also modify a layer color by using a Layer Properties dialog box as described in the "Viewing and Changing Properties" section of the "Using the FPGA Editor" chapter.

Editing the FGPA_editor.ini File

Use the following steps to edit the fpga_editor.ini file.

1. Open the fpga_editor.ini file with a text editor and go to the section of the file where layer colors are defined.

2. Enter the following command for each layer color you want to modify.

`setattr layer` *layer_name* **`color`** *color*

The *layer_name* variable specifies the name of a particular layer. Examples of types of layers are sites, components, switch_boxes, pinwires, local_lines, long_lines, pips, routes, ratsnests, hilite, select, and text. Different layers are available for different device architectures.

The *color* variable specifies the color you want to use for a particular layer. Possible colors are black, blue, green, cyan, red, magenta, yellow, white, grey, dark blue, dark green, dark cyan, dark red, dark magenta, or olive.

3. Save your changes and exit the file.

Using the Command Line Toolbar

At any time during an FPGA Editor session, you can change the color assigned to a specified layer by entering the Setattr command in the Command Line toolbar.

`setattr layer` *layer_name* **`color`** *color*

See the "Editing the FGPA_editor.ini File" section for definitions of the layer name and color variables.

Using the Devcolor Command

The Devcolor command allows you to specify different colors for routing resources based on the names of the resources. The syntax for this command is as follows.

`devcolor wire` *name_pattern color*

The following examples use asterisks as wild card characters in the routing resource names.

devcolor wire H*HEX* blue

devcolor wire H*DOUBLE* dark_blue

devcolor wire V*HEX* green

devcolor wire V*DOUBLE* dark_green

Customizing Fonts (PC only)

On the PC, the FPGA Editor windows use the standard system fonts. To change the fonts used in dialog boxes, menu bars, and other areas of the FPGA Editor window, use the Display icon in the Control Panel.

Customizing with Command Aliases

Use the Alias command to assign a name to an FPGA Editor command stream. You can then enter the alias name to perform the assigned commands. This function is useful for abbreviating or renaming existing commands. Refer to the FPGA Editor online help for more information on the Alias command. Default aliases for the FPGA Editor are in the fpga_editor.ini file.

Defining Keyboard Shortcuts

Note The keyboard shortcuts for the FPGA Editor pull-down menu commands are hard coded and cannot be used with the Alias command. For example, you cannot re-program the Control + O shortcut for the Open command to perform any other command.

For frequently used FPGA Editor commands, you can create keyboard shortcuts by programming one or more keys to execute any of the command line commands described in the FPGA Editor online help. After the shortcut keys are defined, you can invoke a command by pressing the appropriate key when the Array window has the keyboard focus instead of typing the command in the Command Line toolbar.

On PCs, you must click the left mouse button in a window to activate the commands applicable to that window. For example, if your active window is the List window, and you move the cursor to the Array window, and type a keyboard shortcut without first clicking in the Array window, the shortcut will not work.

The default aliases for the keys are contained in the fpga_editor.ini file. You can set up your own aliases in the fpga_editor_user.ini file.

To view a list of the active keyboard shortcuts in the History toolbar, enter the following in the Command Line toolbar.

```
alias
```

The following two lines are a small section of this list.

```
"0x111" ([UpArrow]) = "pan up 75"
"0x113" ([LeftArrow]) = "pan left 75"
```

The hexadecimal number represents the value for the key enclosed in brackets. The information following the equal sign is the FPGA Editor command represented by the key. For example, the first alias listed uses the UpArrow key to pan up 75% of one window height.

Some of the shortcuts that you can customize are listed in the following table.

Table 5-1 Default Shortcuts

Key Name	Action
Left arrow (←)	Pan left 75 % of one window width
Right arrow (→)	Pan right 75 % of one window width
Up arrow (↑)	Pan up 75 % of one window height
Down arrow (↓)	Pan down 75 % of one window height
Shift + ←	Pan to left edge of device
Shift + →	Pan to right edge of device
Shift + ↑	Pan to upper edge of device
Shift + ↓	Pan to lower edge of device
Space bar	Center the window about the cursor
A, a	Post the Command Line toolbar
B, b	Toggle the view of switchboxes
C, c	Toggle the view of components
D, d	Toggle the view of pinwires
G, g	Toggle the view of local lines
L, l	Toggle the view of long lines
N, n	Toggle the view of ratsnest lines
O, o	Zoom all the way out and turn off dense graphical items for speed
P, p	Toggle the view of pips
Q, q	Display information about the object currently under the cursor

Table 5-1 Default Shortcuts

Key Name	Action
R, r	Toggle the view of routes
S, s	Toggle the view of sites
T, t	Toggle the view of text
U, u	Display uphill path back to the source pin of the portion of the net directly under the cursor
W, w	Zoom to a good working level and turn on appropriate items for editing
X, x	Posts the Command Line toolbar
Z, z	Toggle the zoom
Control + e	Change the edit mode to Read-Write to allow changes to your design

The following table lists the hard-coded shortcuts that cannot be customized.

Table 5-2 Hard-Coded Shortcuts

Key Name	Action
F1	Help
F2	Go to command line
Control + N	File → New
Control + F6	Go to next window
Shift + Control + F6	Go to previous window
Control + C	Copy
Control + O	File → Open
Control + P	File → Print
Control + S	File → Save
Control + V	Paste text to the command line
Control + X	Cut text from the command line

You can define keyboard shortcuts either on the command line during an editing session, or in the initialization file. Shortcuts defined during an editing session can only be used for that session.

Shortcuts defined in the .ini file can be used for all subsequent sessions.

Customizing with Command Scripts

A script is a series of commands that you can record and play back.

Recording a Script

Use the following procedure to record a script.

1. If the script is going to be used on objects that will change each time you run the script, you must first select the objects you want the script to process.

2. Select the following to display the Script Record dialog box.

    Tools → Scripts → Begin Recording.
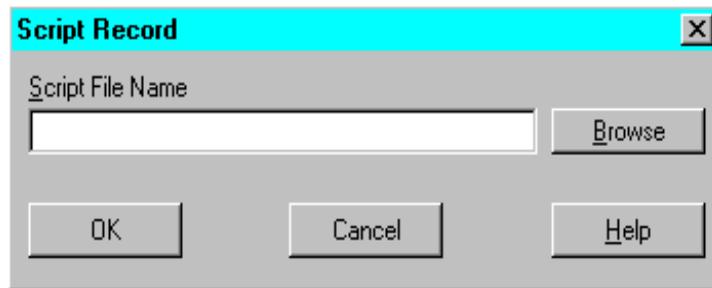
Figure 5-1 Script Record Dialog Box

3. Enter a name in the Script File Name field, or click **Browse** and use the Save As dialog box to specify the name and directory. Script files use a .scr extension. If you do not include an extension, the system adds this extension to your file name.

4. Select **OK**.

5. Enter the series of commands that you want to record on the command line, from the pull-down menus, and so on.

    Note The Script Record dialog box does not allow nested scripts. If you attempt to record a script within a script using the Script Record dialog box, you will get an error message. However, you can manually edit a script file to use nested scripts.

6. To finish recording the script, select the following.

    Tools → Scripts → End Recording.

Playing Back a Script

Use the following procedure to play back a script.

1. If the script commands are performed on selected objects, you must first select these objects.

2. Select the following to display the Script Playback dialog box.

    Tools → Scripts → Playback
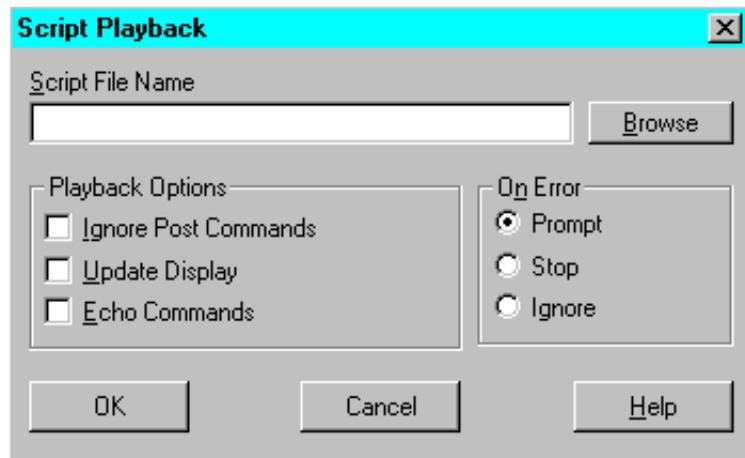
Figure 5-2 Script Playback Dialog Box

3. Enter the script file name in the Script File Name field, or click **Browse** and use the Open dialog box to select the file.

4. Select the Playback Options to specify the following.

    ♦   Ignore Post Commands

        Dialog boxes resulting from commands in the script are not posted as the script plays back. If you do not select this option, dialog boxes are posted and remain open after playback is finished. In either case, the commands invoked by the dialog boxes are performed.

♦ Update Display

This option updates the display in the FPGA Editor window every time a command that redraws the window is run by the script. If you do not select this option, the display is updated only after script playback is finished.

♦ Echo Commands

The commands run by the script appear in the History toolbar during playback. If you do not select this option, the only commands that appear in the History toolbar are the commands indicating when the script playback starts and ends. This option is useful because you can scroll through the History toolbar to view the commands.

5. Select **OK**.

The commands in the script are performed on the selected objects.

Appendix A

FPGA Editor Files

This appendix provides a listing of the files used by the FPGA Editor.

Name	Type	Produced By	Description
fpga_editor.ini	ASCII	Xilinx software in $XILINX/data	Script that determines what FPGA Editor commands are performed when FPGA Editor starts up
fpga_editor_user.ini	ASCII	Xilinx software	A supplement to the *fpga_editor.ini* file; used for modifying or extending the *fpga_editor.ini* file (located in user's home directory)
.log	ASCII	FPGA Editor	FPGA Editor command log file; keeps a record of all FPGA Editor commands executed and output generated
.ncd	Data	Mappers, Translators, PAR, FPGA Editor	A flat physical design database correlated to the physical side of the .ngd in order to provide coupling back to your original design
.nmc	Binary	FPGA Editor	Xilinx physical macro library file; contains a physical macro definition that can be instantiated into an FPGA Editor design

Name	Type	Produced By	Description
.out	ASCII	FPGA Editor	The system keeps track of all commands performed on your design in the temporary command log file, *design_name*_fpga_editor.out. When you end an FPGA Editor session, this file is renamed *design_name*_fpga_editor_YYMMDD_HHMMSS.log, where Y is year, M is month, D is day, H is hour, M is minute, and S is second. If a session is unexpectedly terminated, the *design_name*_fpga_editor.out file remains in your current design directory, and the next time you start the FPGA Editor, the *design_name*_fpga_editor.out file is renamed *design_name*_fpga_editor.rcv.
.pcf	ASCII	Mapper, Translator, FPGA Editor	A file containing constraints specified during design entry and constraints added by the user
.rcv	ASCII	FPGA Editor	FPGA Editor recovery file (see description of .out file)
.scr	ASCII	FPGA Editor	Command script file

Window Manager Resources

On a UNIX workstation, you can customize the appearance of the FPGA Editor by modifying the X window system configuration file, .Xdefaults. In addition, you may be able to customize your window manager to simplify access to the FPGA Editor and the graphic shells. This is done by modifying the window manager-specific resource files.

Glossary

Architecture

Architecture is the common logic structure of a family of programmable integrated circuits. The same architecture can be realized in different manufacturing processes. Examples of Xilinx architectures are the XC4000, Virtex, and XC5200 devices.

Array Window

The Array window displays a graphical representation of the FPGA device. The device components and the interconnections (both logical and routed) between these components are displayed in this window. When you edit the internal logic of a programmable component such as a logic block, a schematic of the interior of the component is displayed in the Block window.

Auto Place

The automatic placement (Auto Place) is software that selects sites intelligently based on routability. You can automatically place selected components in your design.

Auto Route

Auto Route automatically routes the objects you specify.

Bel

A Basic Element, for example, an individual flip-flop or LUT.

Block

A block is a group of one or more logic functions. A block is a schematic or symbol sheet. There are four types of blocks.

- A Composite block indicates that the design is hierarchical.

- A Module block is a symbol with no underlying schematic.

- A Pin block represents a schematic pin.

- An Annotate block is a symbol without electrical connectivity that is used only for documentation and graphics.

Block Window

This window is displayed when you double click the left mouse button on a logic block.
The Block window is used to edit logic blocks. You can use only one Block window at a time for editing; however, you can have additional Block windows open for viewing.

Carry Logic

An architecture feature of the Xilinx Virtex, XC4000, and XC5200 families. Carry logic is designed to speed-up and reduce the area of counters, adders, incrementers, decrementers, comparators, and subtractors. It is a special interconnect that speeds up the carry path of adders and counters from one CLB to another. This dedicated carry line runs along each column of CLBs as well as the top and bottom CLBs. FPGA Express can synthesize carry logic directly.

CLB

The Configurable Logic Block (CLB). Constitutes the basic FPGA cell. It includes two 16-bit function generators (F or G), one 8-bit function generator (H), two registers (flip-flops or latches), and reprogrammable routing controls (multiplexers). CLBs are used to implement macros and other designed functions. They provide the physical support for an implemented and downloaded design. CLBs have inputs on each side, and this versatility makes them flexible for the mapping and partitioning of logic.

Command Line Toolbar

The Command Line toolbar is located below the History toolbar. Use this toolbar to enter commands from the keyboard.

Component

A component is an instantiation or symbol reference from a library of logic elements that can be placed on a schematic.

Constraints

Constraints are specifications for the implementation process. There are several categories of constraints: routing, timing, area, mapping, and placement constraints. Using constraints, you can force the placement of logic (macros) in CLBs, the location of CLBs on the chip, and the maximum delay between flip-flops. PAR does not change the location of constrained logic. A constraints file specifies constraints (location and path delay) information in a textual form. An alternate method is to place constraints on a schematic.

Delay Calculator

This tool calculates and displays the delay associated with load pins and driver pins in a given net or path.

Design Rule Check (DRC)

Physical Design Rule Check (DRC) is a series of tests to discover logical and physical errors in the design. Physical DRC is applied to the FPGA Editor and BitGen. Results of the DRC are written into the History toolbar.

Device

A device is an integrated circuit or other solid-state circuit formed in semiconducting materials during manufacturing. Each Xilinx architecture family contains specific devices, such as xc4003e and xc5202. A complete Xilinx part number includes architecture (for example, xc4000ex), device (for example, xc4028ex), package (for example, pg299), and speed (for example, -3).

Edit Mode

Specifies what modifications can be made to your design. The Read Only option prevents any changes to the design. The Read Write option allows you to save changes to your design, including changes to the logical organization. Changes to the logical organization include creating or deleting nets or components, and reprogramming programmable components. The No Logic Changes option allows you

to make placement and routing changes, but not changes to the logical organization. You cannot add or delete nets and components, or reprogram programmable components.

External Pin

A macro pin used to connect the components in an instantiated macro to other components in your design (outside of the macro).

FPGA

Field Programmable Gate Array (FPGA), is a class of integrated circuits pioneered by Xilinx in which the logic function is defined by the customer using Xilinx development system software after the IC has been manufactured and delivered to the end user. Gate arrays are another type of IC whose logic is defined during the manufacturing process. Xilinx supplies RAM-based FPGA devices.
FPGA applications include fast counters, fast pipelined designs, register intensive designs, and battery powered multi-level logic.

FPGA Editor Main Window

The FPGA Editor main window is the background against which all other windows are displayed. To work in a particular section of the interface, click the left mouse button in that section. You can execute commands from the menus, toolbars, and command line. The client or work space area includes the Array window, List window, and World window.

Fpga_editor.ini File

Script that determines what FPGA Editor commands are performed when the FPGA Editor starts up. For more information on the fpga_editor.ini file, see the "Customizing the FPGA Editor" chapter in the FPGA Editor online manual (accessible from the FPGA Editor Help menu).

Fpga_editor_user.ini File

You can customize the fpga_editor.ini initialization file by creating an fpga_editor_user.ini file in your home directory. When the FPGA Editor is initialized, it reads the fpga_editor.ini file first and then the fpga_editor_user.ini file. For more information on the fpga_editor_user.ini file, see the "Customizing the FPGA Editor"

chapter in the FPGA Editor online manual (accessible from the FPGA Editor Help menu).

History Toolbar

The History toolbar is located below the Array window and displays commands and responses. All error messages, warnings, and command responses are written to the History toolbar. Information in the History toolbar is especially useful for deciphering unexpected command results.

Integrated Logic Analyzer (ILA)

The Integrated Logic Analyzer (ILA) is a parameterized soft core that you can embed in your designs. It is used in conjunction with the ChipScope software to provide real-time, on-chip debugging and design verification.

IOB (input/output block)

An IOB is a collection or grouping of basic elements that implement the input and output functions of an FPGA device.

Layer

An FPGA Editor layer contains all of one type of object (for example, all long lines in the device, or all components in the design database). Layers are graphically displayed in the editing area.

Layer Visibility Toolbar

The Layer Visibility toolbar allows you to specify which objects are displayed in the Array window. Select the layers you want displayed and deselect the layers you want hidden.

List Window

The List window displays a list of the components, nets, layers, paths, and macros in your design. Use the pull-down list box at the top of the window to specify the items you want displayed in the List window.

Local Line

Local lines usually span across multiple CLBs; typically they go between switch boxes. Local lines do not directly connect to site pins,

such as direct connects, and they do not span across the entire length of the device, such as long lines.

Locking

A lock constraint in the PCF file locks a component. A lock routing constraint specifies that the current routing cannot be changed or unrouted. A lock placement constraint specifies that placed components cannot be unplaced, moved, or deleted.

Log File

The log file is a command log file. This file records all FPGA Editor commands executed and output generated.

Long Line

A long line connects to a primary global net or to any secondary global net. Long lines carry signals across the length or width of the chip with minimal delay and negligible skew.

Look-up Table (LUT)

Look-up tables (LUTs) are used to implement function generators in CLBs. Four independent inputs are provided to each of two function generators (F1-F4 and G1-G4). These function generators can implement any arbitrarily defined Boolean function of four inputs. The H function generator can implement any Boolean function of four inputs.

Macro

A physical macro is a logical function that has been created from components of a specific device family. Physical macros are stored in files with the extension .nmc.

Macro Instance

A copy of a macro library file inserted in a design file. When you add a macro instance to a design you "instantiate" the macro. A design may contain multiple instances of the same library file, and each will receive a unique name. Since the library file is copied into the design file when you instantiate a macro, if you then change the library file the changes will not be reflected in the macro instantiated in the design file.

Macro Library File

A file containing the definition of a macro. Macro library files have an .nmc extension.

MAP

Mapping is the process of assigning a design's logic elements to the specific physical elements that actually implement logic functions in a device.

Menu Bar

The menu bar is located above the Array window. Most of the FPGA Editor commands are available in the pull-down menus of the FPGA Editor window after a design is loaded.

NCD File

An NCD file is the output design file from the MAP or PAR program and represents the physical design.

Net

1. A net is a logical connection between two or more symbol instance pins. After routing, the abstract concept of a net is transformed to a physical connection called a wire.

2. A net is an electronic connection between components or nets. It can also be a connection from a single component. It is the same as a wire or a signal.

Net Delay

Displays the delay for all pins in a net. You can either find the delay for all pins in the net or you find delays for specific pins.

NMC File

A NMC file contains a physical macro which can be created or viewed with the FPGA Editor.

Package

A package is the physical packaging of a chip, for example, PC84, VQ100, and HQ240. A complete Xilinx part number includes

architecture (for example, xc4000ex), device (for example, xc4028ex), package (for example, pg299), and speed (for example, -3).

Path

A path is a connected series of nets and logic elements. A path has a start point and an end point that are different depending on the type of path. The time taken for a signal to propagate through a path is referred to as the path delay.

PAR

Place and route tool. PAR is a program for mapping, placing, and routing an FPGA design. The process is called design implementation.

Physical Constraints File (PCF)

The Physical Constraints File (PCF) is an ASCII file that contains physical constraints created by the MAP program, as well as physical constraints entered by you. You can edit the PCF file in FPGA Editor.

Physical Macros

A physical macro is a logical function that has been created from components of a specific device family. Physical macros are stored in files with the extension .nmc.

Pin

A pin is an attachment point on a site or a component. Nets can be attached to pins.

Pinwire

Pinwires are wires that are directly tied to the pin of a site (CLB, IOB, and so on).

Placing

Placing is the process of assigning physical device cell locations to the logic in a design.

Probe

A probe is an IOB that you can add to a net to determine the value of the net. You can use probes to debug your design.

Programmable Interconnect Points (PIP)

Programmable interconnect points, or PIPs, provide the routing paths used to connect the inputs and outputs of IOBs and CLBs into logic networks. A PIP is a CMOS transistor switch that you can program to turn on or off.

Property

Properties are instructions placed on symbols or nets in an FPGA schematic to indicate their placement, implementation, naming, directionality, or other properties.

Ratsnest

A ratsnest consists of lines that are point-to-point connections between unrouted pins on a given net.

Reference Component

A component in the macro library file used as a reference when a macro instance is placed, moved, or copied. Placement and routing of all other pre-placed macro components are determined relative to this component.

Router

The router connects the pins on a net (in the logical design) using the available routing resources in the FPGA. The physically routed net or signal is called a wire.

Routing

The process of assigning logical nets to physical wire segments in the FPGA that interconnect logic cells.

Site

A site is a programmable logic element (used or unused) location within the device, sites are potential locations for components and are displayed in the editing area as outlines of components.

Speed

Speed is a function of net types, CLB density, switching matrices, and architecture. A complete Xilinx part number includes architecture (for

example, xc4000ex), device (for example, xc4028ex), package (for example, pg299), and speed (for example, -3).

Status Bar

The status bar appears at the bottom of the main window. When you select a menu command, a brief description of the command's function appears in the status bar.

Switch Box

A switch matrix is a collection of transistors located between CLB blocks that enables the connection of two interconnect lines. PAR uses the switch matrices and interconnects to connect CLB inputs and outputs. Switch matrices reduce some of the net delay. They have three possible directions: top, bottom, and left.

Timing

Timing is the process that calculates the delays associated with each of the routed nets in the design.

Title Bar

The title bar displays the program name and the name of the currently loaded design.

TRACE

TRACE (Timing Reporter and Circuit Evaluator) is a program you can run within FPGA Editor that provides static timing analysis of the physical design based on input timing constraints.

User Toolbar

The User toolbar provides a convenient way to perform frequently used commands. To use a command, select the appropriate command button with the left mouse button.

Wire

A wire is a physically routed net or a signal using the routing resources of the FPGA.

World Window

The World window shows the area of the device that is currently displayed in the Array window. As you pan and zoom the Array window, notice the corresponding changes in the size and position of the rectangle within the World window. Also, any objects selected in the Array window appear in the World window. You can drag the inner box with the mouse button to pan the display to the desired position. If you have multiple Array windows, the World window displays a rectangle for each Array window.