Floorplanner Guide

Introduction

Design Flow

Getting Started

Using the Floorplanner

Glossary

About This Manual

This manual describes the Xilinx Floorplanner, a graphically based tool that allows you to interactively and automatically place logic symbols from a hierarchical design into a Xilinx target FPGA.

Before using this manual, you should be familiar with the operations that are common to all Xilinx software tools: how to bring up the system, select a tool for use, specify operations, and manage design data. These topics are covered in the *Quick Start Guide*. Other publications you can consult for related information are the *Development System Reference Guide* and the *HDL Synthesis for FPGA's Design 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 Floorplanner interface, including basic operations, input and output files, and supported FPGA architectures.
- Chapter 2, "Design Flow" describes five distinct work flows in which you can use the Floorplanner.
- Chapter 3, "Getting Started" describes how to invoke the Floorplanner from the Design Manager or as a standalone tool, and provides more details about the interface.
- Chapter 4, "Using the Floorplanner" describes several important floorplanning procedures for implementing high-density designs in Xilinx devices.

- "Glossary" defines the terms used in this 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 contains 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

About This Manual

Manual Contents v
Additional Resources vi

Conventions

Typographical vii
Online Document viii

Chapter 1 Introduction

What is the Floorplanner? 1-1
Floorplanner Icon 1-3
Why Floorplan? 1-3
Floorplanning Prerequisites 1-3
Features of the Floorplanner 1-4
Supported Architectures 1-4
New Files 1-4
Input Files 1-5
Output Files 1-6

Chapter 2 Design Flow

Place and Route, then Floorplan 2-1
Floorplanning Prior to Place and Route 2-3
Iterative Floorplanning 2-4
Incremental Design Changes 2-5

Chapter 3 Getting Started

Running the Floorplanner 3-1
Design Manager Interface 3-2
Command Line Interface 3-4
Floorplanner Interface 3-5

Toolbar 3-5
Status Bar 3-5
Mouse 3-6
Keyboard 3-6
Dialog Boxes 3-7
The Floorplanner Windows 3-8
Primary Window 3-8
Design Hierarchy Window 3-10
Hierarchy Display 3-11
Selecting Logic 3-11
Expanding and Collapsing Hierarchical Groups 3-12
Hierarchical Group Annotation 3-13
Symbol Annotation 3-14
Design Nets Window 3-15
Floorplan Window 3-15
Resource Graphics 3-17
Placement Window 3-18
Closing the Current Design 3-18
Exiting the Floorplanner 3-19

Chapter 4 Using the Floorplanner

Opening a File 4-3
Saving a File 4-3
Using Colors 4-4
In the Design Hierarchy Window 4-4
In the Floorplan Window 4-4
Distinguishing Logic 4-5
Floorplanning Logic 4-6
Floorplanning Designs that Contain RPMs 4-8
Creating Groups 4-9
Manual Grouping 4-9
Automatic Grouping 4-12
Using Area Constraints 4-13
Flattening and Building the Hierarchy 4-13
How to Flatten the Hierarchy 4-13
How to Rebuild the Hierarchy 4-14
Walking Through the Design 4-14
Finding Logic 4-14
Finding Nets 4-15
Displaying the Ratsnest 4-15
Using the Ratsnest 4-15
Viewing Selected Nets in the Ratsnest 4-16

Analyzing PAR Placement 4-16
Analyzing PAR Placement for Timing Constraints 4-19
From the Design Manager 4-19
Using Find and Ratsnest to Find Critical Nets 4-21
Finding Logic Connected to Nets 4-23
Displaying Resources and Logic 4-23
Performing Detailed Manual Placement 4-25
Checking the Floorplan 4-26
Aligning Symbols 4-26
Working with Patterns 4-30
Creating a Pattern 4-31
Using a Pattern 4-31
How to Interleave Buses 4-32
Design Example 4-32
Iterative Floorplanning 4-41
Floorplanning Incremental Schematic Changes 4-42
Design Example 4-42
Making Small Modifications to Automatic Placement 4-45
Lock Down I/Os from Automatically Placed Design 4-46
Getting Started With an Unfamiliar Design 4-47
Creating Hierarchy at a Higher Level 4-47
Creating Subgroups 4-48
Floorplanning the New Hierarchy 4-48

Glossary

Chapter 1

Introduction

This chapter describes the graphical interface and the important features and capabilities of the Xilinx Floorplanner. It contains the following sections.

- "What is the Floorplanner?" is a general introduction to the Floorplanner tool.
- "Floorplanner Icon" shows the icon that appears in the Design Manager window.
- "Why Floorplan?" explains how the Floorplanner can help you improve the performance and density of your design.
- "Floorplanning Prerequisites" explains the necessary requirements for successful floorplanning.
- "Features of the Floorplanner" lists the features provided by the Floorplanner.
- "Supported Architectures" lists the Xilinx device families that Floorplanner supports.
- "New Files" describes the two new file types that Floorplanner uses.
- "Input Files" lists the files that Floorplanner uses as input.
- "Output Files" lists the files that Floorplanner generates.

What is the Floorplanner?

The Floorplanner is a graphical placement tool that gives you control over placing a design into a target FPGA using a "drag and drop" paradigm with the mouse pointer.

The Floorplanner displays a hierarchical representation of the design in the Design Hierarchy window using hierarchy structure lines and colors to distinguish the different hierarchical levels. The Floorplan window displays the floorplan of the target device into which you place logic from the hierarchy. The following figure shows the windows on the PC version.

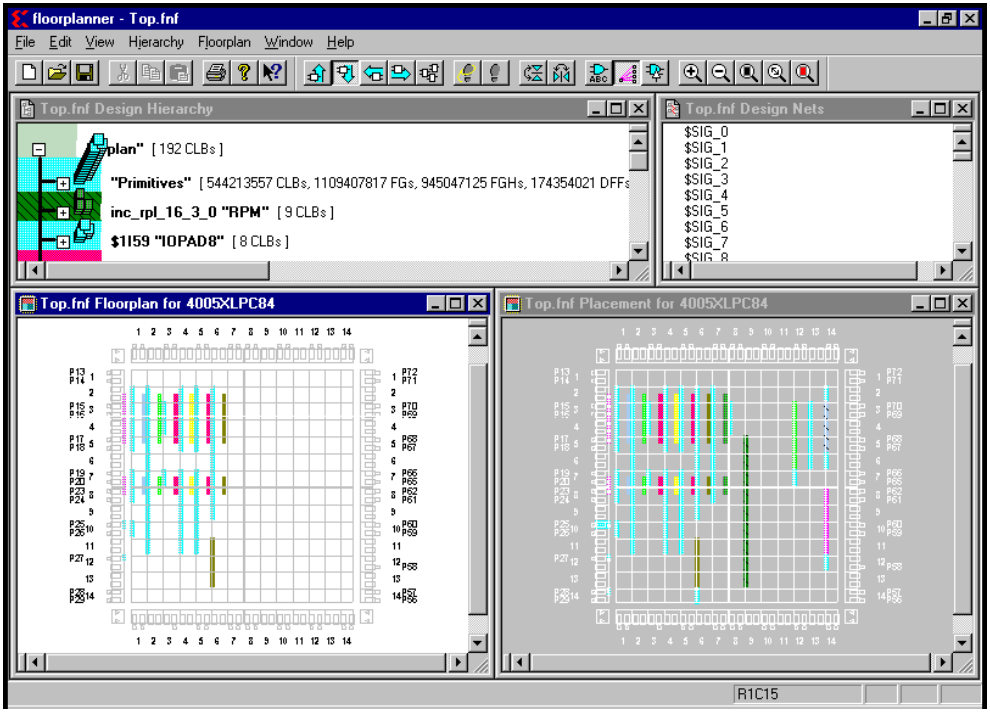


Figure 1-1 Floorplanner Window

Logic symbols represent each level of hierarchy in the Design Hierarchy window. You can modify that hierarchy in the Floorplanner without changing the original design.

You use the mouse to select the logic from the Design Hierarchy window and place it in the FPGA represented in the Floorplan window.

Alternatively, you can invoke the Floorplanner after running the automatic place and route tools to view and possibly improve the results of the automatic implementation.

Floorplanner Icon

The Floorplanner can be started by clicking the Floorplanner icon in the Design Manager screen. The following figure shows the icon.

Why Floorplan?

Floorplanning is an optional methodology to help you improve performance and density of a fully, automatically placed and routed design. Floorplanning is particularly useful on structured designs and data path logic. With the Floorplanner, you see where to place logic in the floorplan for optimal results, placing data paths exactly at the desired location on the die.

With the Floorplanner, you can floorplan your design prior to or after running PAR. In an iterative design flow, you floorplan and place and route, interactively. You can modify the logic placement in the Floorplan window as often as necessary to achieve your design goals. You can save the iterations of your floorplanned design to use later as a constraints file for MAP.

The 3.1i version of the Floorplanner allows you to begin floorplanning with just an NGD file generated in a previous flow. You can manually make IOB assignments which you can then write to a UCF file. The Floorplanner edits the UCF file by adding the newly created placement constraints. The placement constraints you create in the Floorplanner take precedence over existing constraints in the UCF.

Floorplanning Prerequisites

The Floorplanner is specifically intended to assist those users who require some degree of handcrafting for their designs. You must understand both the details of the device architectures and how floorplanning can be used to refine a design. Successful floorplanning is very much an iterative process and it can take time to develop a floorplan that outperforms an "automatically" processed design.

Because of the nature of the Floorplanner's interaction with the automatic MAP and PAR tools, several prerequisites are necessary in order to floorplan your design successfully.

- Detailed knowledge of the specifics of the target architecture and part
- Detailed knowledge of the specifics of the design being implemented
- A design that lends itself to floorplanning
- A willingness to iterate a floorplan to achieve the desired results
- Realistic performance and density goals

Features of the Floorplanner

The Floorplanner provides an easy-to-use graphical interface that offers the following features.

- Interacts at a high level of the design hierarchy, as well as with low-level elements such as I/Os, function generators, tristate buffers, flip-flops, and RAM/ROM
- Captures and imposes complex patterns, which is useful for repetitive logic structures such as interleaved buses
- Automatically distributes logic into columns or rows
- Uses dynamic rubberbanding to show the ratsnest connections
- Finds logic or nets by name or connectivity
- Permits design hierarchy rearrangement to simplify floorplanning
- Groups logic by connectivity or function
- Offers an alternative UCF flow that allows you to write out graphically created constraints to a UCF
- Allows binding and unbinding of RPMs
- Identifies placement problems in the Floorplan window
- Provides online help

Supported Architectures

The Floorplanner supports all Xilinx architectures in the Spartan/-II™, Virtex/-E/-II™, and XC4000™ device families.

New Files

There are three file types that Floorplanner uses.

- FNF

  The Floorplanner Netlist File (FNF) is the floorplanner's database. Its core function is to retain a record of all the (physical) constraints entered in the Floorplan window. If the FNF is generated using a placed NCD file, the placement information is also recorded in the FNF for use by the placement window. Additionally, the FNF file retains user-created groups from the Design Hierarchy window. When design modifications are made (modified source files (.ngd) or new placement (.ncd)), the existing FNF file can be updated with the new information using the **`File`** → **`Update`** command. This allows for design iterations without the loss of your previous work. Any design archive should include the FNF file to allow subsequent design or constraint modifications.

- MFP

  The Mapper Floorplan (MFP) file links the Floorplanner to the M1 flow by directing MAP behavior. It is the intervention into the map phase of the flow that permits you to floorplan at the BEL (Basic Element) level. The MFP is a subset (physical constraints only) of the FNF file. The FNF2MFP utility is automatically invoked when saving a floorplan, which produces an MFP file from the relevant information in the FNF file. The MFP file name always has the same root name as the FNF file, for example, design.fnf and design.mfp.

  Note The Floorplanner cannot read MFP files. You must save the FNF file in order to archive floorplan information.

- UCF

  The UCF is an ASCII file specifying constraints on the logical design. These constraints affect how the logical design is implemented in the target device. The Floorplanner allows you to read

in an existing UCF file, create constraints graphically in the floorplan, and write the constraints out to the UCF file. You perform these tasks using the UCF flow.

Input Files

The Floorplanner can read in the following input files.

- NCD

  This file is generated by either MAP or PAR. It is used by the Floorplanner during the **File** → **New** command to generate the physical design for the FNF file.

- NGD

  This file is generated by NGDBuild. It is used by the Floorplanner during the **File** → **New** command to correlate the physical design to the logical design when creating the FNF file.

- FNF

  A previous version of this file, which was generated by a **File** → **Save** in the Floorplanner, can optionally be used in the creation of a new FNF file. If used, it helps retain design constraints between floorplanning iterations.

- NGD

  The **File** → **New** command allows you to open an NGD source design file.

- UCF

  The 3.1i version of the Floorplanner allows you to read in a UCF file with the **File** → **Read Constraints** command.

Output Files

The Floorplanner generates the following output files.

- FNF

  The FNF file is the Floorplanner's database. It can be saved by using the **File** → **Save** command.

- MFP

This file is generated when the FNF file is saved in the Floorplanner. It is used as an input to MAP to transfer physical constraints from the Floorplanner back to the automatic implementation tools.

- UCF

  You can write out constraints created in a UCF flow to a UCF file by using the **`File`** → **`Write Constraints`** command.

Chapter 2

Design Flow

This chapter describes the four different design flows that you can use with the Floorplanner to implement your design in a Xilinx FPGA. Accompanying each design flow is a comprehensive flow chart that indicates the programs you use, the input files required, and output files that are generated. The four design flows are described in the following sections.

- "Place and Route, then Floorplan"
- "Floorplanning Prior to Place and Route"
- "Iterative Floorplanning"
- "Incremental Design Changes"

Xilinx strongly recommends that you read the *HDL Synthesis and Simulation Design Guide* before attempting to floorplan your HDL designs. This document explains HDL-specific design issues and understanding them will make floorplanning your HDL designs easier and more effective.

The design flows in this chapter present a general picture of where the Floorplanner fits in the Xilinx design flow; in some instances the descriptions of the design flows are more relevant to designers using schematic capture tools than to designers using HDL.

Place and Route, then Floorplan

The first design flow describes how to Floorplan your design after placing and routing your design. This is the preferred methodology because it allows you to view both the physical constraints for the design and the results of the automatic placement.

You enter your design using either a schematic capture tool or HDL. Next, run MAP and PAR to place and route the design in a target

FPGA device. To view and improve performance of the automatic implementation, create a new Floorplan Netlist File within the Floorplanner from the placed and routed NCD file. Next, use the Floorplanner to constrain critical paths or adjust the automatic placement. Finally, run MAP and PAR with the newly generated MFP file to obtain the results of the floorplanned design. Refer to the design flow in the following figure.

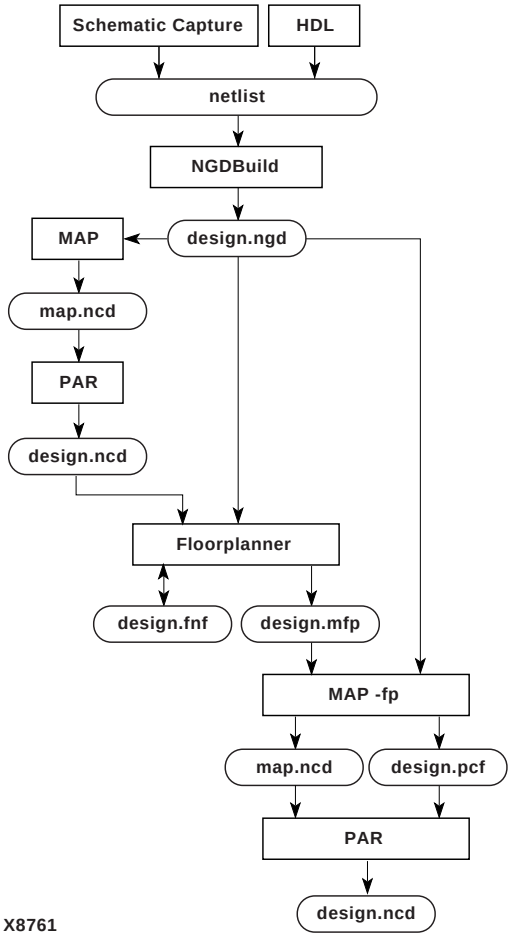


Figure 2-1 PAR Before Floorplanning Design Flow

Floorplanning Prior to Place and Route

The second design flow is to floorplan your design before using PAR to place and route it. In this flow, you enter your design using either a schematic capture tool or HDL. Run MAP on the design to create a physical design file (NCD). Use the Floorplanner to define placement constraints by manually placing selected logic into the resources of the target device. Next, run MAP and PAR to fit the design into the target FPGA using the Floorplan constraints. Refer to the design flow in the following figure.

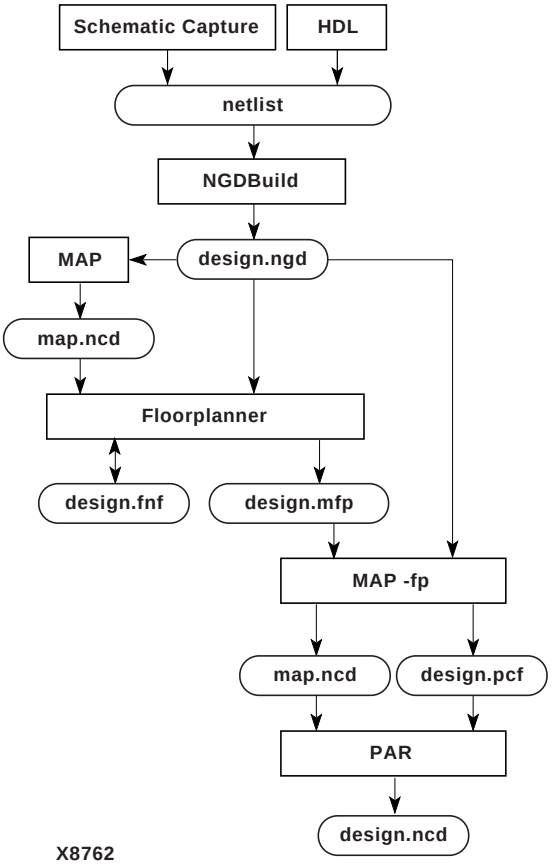


Figure 2-2 Floorplan First Design Flow

Iterative Floorplanning

In the third design flow, iterative floorplanning, you enter the design using a schematic capture tool or HDL. Next, use the Floorplanner to constrain portions of the design. Then, run MAP and PAR to map, place, and route the design into the target FPGA.

Based on the results of the automatic place and route tools, you can modify the currently floorplanned logic or select another portion of the design to constrain. Run MAP and PAR again with the new floorplanner constraints.

In addition, you can make small modifications to the placement done by PAR. After copying the placement over to the Floorplan window, you can make small changes to the Floorplan and save the new fully-constrained FNF file. The **`Floorplan` → `Constrain All From Placement`** command fully constrains the placement of the design and can be used to fix small performance problems, such as a few design elements that are not optimally placed.

Repeat this Floorplanner-to-MAP and PAR loop until you have achieved your performance goals for the design. Refer to the design flow in the following figure.

Note that the same NGD file is used throughout the design flow.

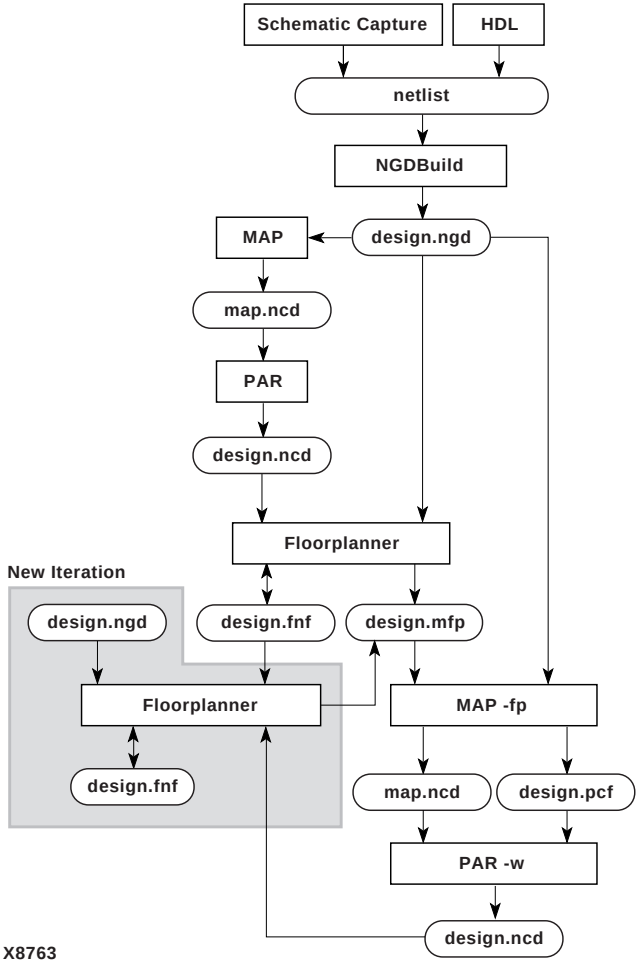


Figure 2-3 Iterative Floorplanning Design Flow

Incremental Design Changes

In the fourth design flow, you make changes to the schematic of a design that has been previously implemented in an FPGA, with or without floorplanning. You must re-implement the design into the target device while making only minimal changes to the previous implementation. These changes could be one or more of the following.

- Adding logic
- Removing logic
- Changing existing logic

If you used the Floorplanner to floorplan the original design, use the Floorplanner now to correlate the logic in that design with the new changes, and adjust the constraints information accordingly. Next, use the Floorplanner's output MFP file to constrain the design during the mapping and placement phases of the implementation.

For HDL users, incremental design change is more complex with HDL designs because the synthesis tools change symbol names whenever the compilation method changes. When applying a previous revision of your Floorplan to the newly synthesized revision, it may be necessary to constrain some or all of the previously floorplanned elements.

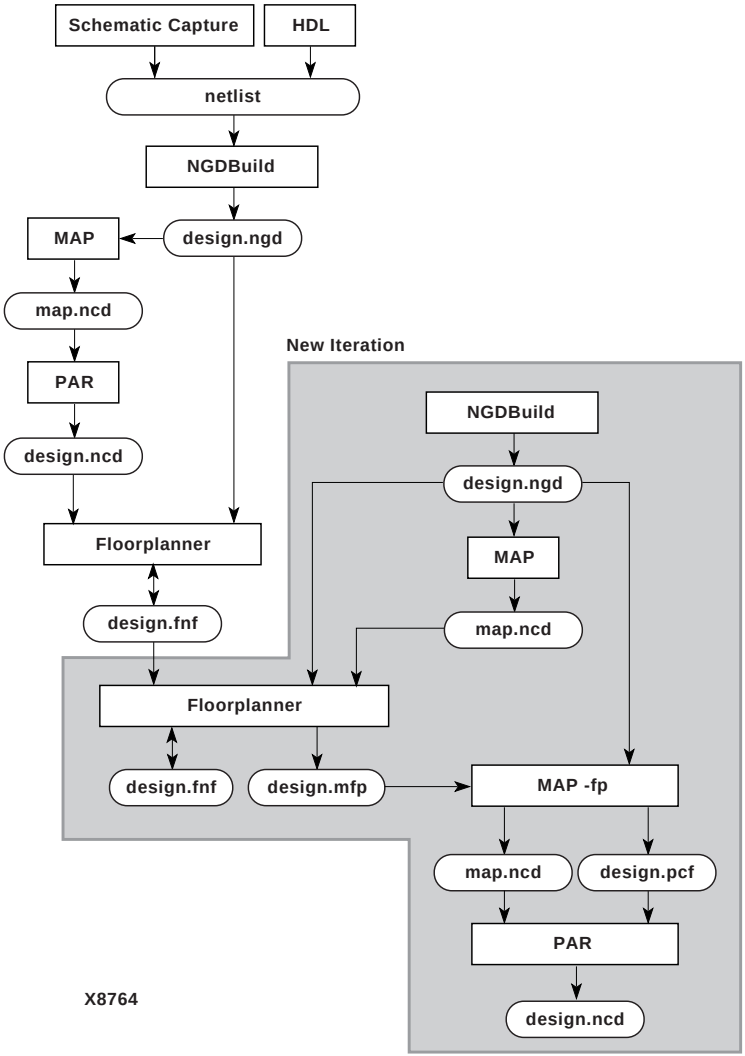


Figure 2-4 Incremental Design Change Design Flow

Creating UCF Constraints from IOB Placement

In the fifth design flow, you add constraints to the UCF file through the Floorplanner and iteratively implement your design to achieve optimal placement.

To begin with, you need only the NGD file generated in a previous flow. In the Floorplanner, you manually make IOB assignments which are automatically written into the UCF file. The Floorplanner edits the UCF file by adding the newly created placement constraints. The placement constraints you create in the Floorplanner take precedence over existing constraints in the UCF.

Next, go through the steps of implementing your design by running NGDBuild, MAP, and PAR. The following flowchart illustrates how you refine the UCF file through iterative runs.

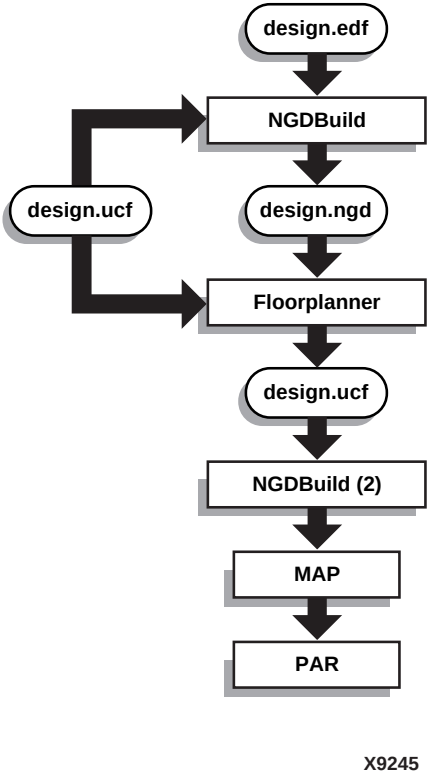


Figure 2-5 Creating UCF Constraints for IOB Placement Flow

Chapter 3

Getting Started

This chapter describes how to start and exit the Floorplanner. It also explains the basic elements and operations of the Floorplanner graphical user interface (GUI).

This chapter contains the following sections.

- "Running the Floorplanner" describes how to start the Floorplanner.
- "Design Manager Interface" explains how to use the Design Manager to run the Floorplanner.
- "Command Line Interface" shows the syntax to use when running NGDBuild, MAP, and PAR in a command window as part of the floorplanning process.
- "Floorplanner Interface" describes the GUI.
- "The Floorplanner Windows" describes the four sub-windows inside the Floorplanner primary window.
- "Closing the Current Design" describes how to close the current design without exiting the Floorplanner.
- "Exiting the Floorplanner" explains how to save your current design and exit the Floorplanner.

Running the Floorplanner

You can run the Floorplanner on a PC running Windows NT® or on an HP-based or Solaris®-based workstation under Design Manager. You can input either an HDL-based design or a schematic-based design.

The following figure shows the Design Manager screen from which you can launch the Floorplanner.

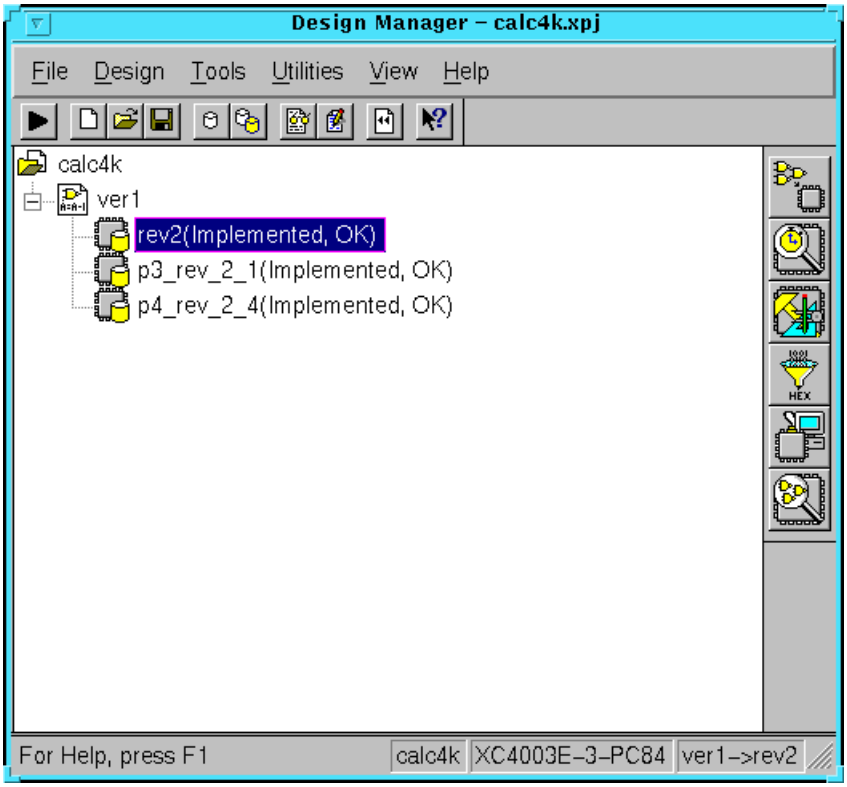


Figure 3-1 Design Manager Window

You can invoke the Floorplanner from the Design Manager in either of two ways.

- Select the **`Tools`** → **`Floorplanner`** command.
- Click the Floorplanner toolbox button. In the above figure, this is the third button from the top in the toolbar on the right.

  Refer to the *Design Manager/Flow Engine Guide* for more details about invoking the Floorplanner from the Design Manager.

Design Manager Interface

When you use the Floorplanner, an MFP file is generated that contains mapping information. You can instruct the Design Manager to use this file as a guide for mapping an implementation revision.

To guide a design with Floorplan files, do the following steps. For more information, refer to the *Design Manager/Flow Engine Guide*.

1. Select **Design** → **Set Floorplan File(s)** from the Design Manager.

   The Set Floorplan File(s) dialog box appears, as shown below.

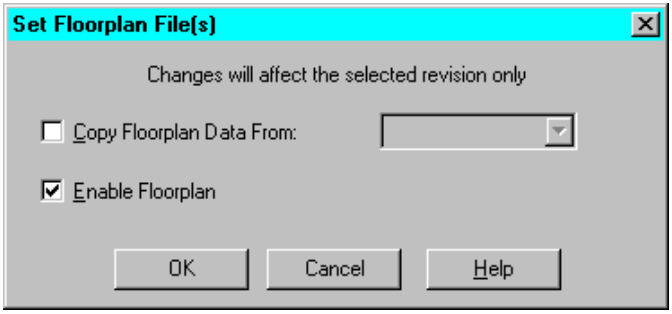


Figure 3-2 Set Floorplan File(s) Dialog Box

2. Select the Copy Floorplan Data From option to copy a floorplan file into the selected implementation revision. If the Enable Floorplan option is selected, the software uses this file to implement the design.
3. Deselect the Enable Floorplan option if you want to keep your floorplan intact, but do not want to guide your design in this revision. By default, the Enable Floorplan option is selected and the software uses the specified floorplan file. Note that if you deselect this option and implement your design, you are prompted to confirm that you want to implement without using a floorplan file.
4. Select a Floorplan guide design from the Floorplan Design drop-down list.
   - Select an existing implementation revision.
   - Select **None** if you do not want to guide the design.
   - Select **Custom** to guide from any mapped file in your file system, including designs not generated from within the Design Manager. This option invokes the Custom dialog box, shown in the following figure. You can specify an FNF file for the Floorplanning File field and an MFP file for the Floorplanned Guide File field.

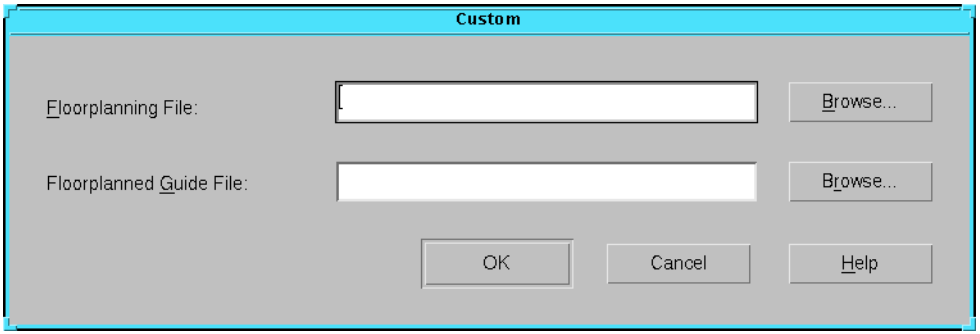


Figure 3-3 Custom Dialog Box (Floorplan Files)

5. The Flow Engine uses the selected file to guide the implementation.

Command Line Interface

As shown in Figure 2-1 of the "Design Flow" chapter, the Floorplanner interacts with the core flow by first reading in an NCD file generated by MAP or PAR and an NGD file generated by NGDBuild. Next, it writes out a *design*.fnf file, which stores all the Floorplanning information, and a *design*.mfp file, which is used during MAP to apply the floorplanning constraints. Following is a sample of the command lines that would be used in a typical floorplanning session.

```
ngdbuild -p part_name design_name

map -p part_name -o map.ncd design_name.ngd
design_name.pcf

par map.ncd part_name.ncd part_name.pcf
```

Use **File** → **New** in the Floorplanner to create a floorplan and **File** → **Save** to create *design_name*.fnf and *design_name*.mfp.

```
map –fp design_name.mfp –p part_name -o map.ncd
design_name.ngd design_name.pcf
```

Note The **–fp** option above tells MAP to use the Floorplanner constraints in the MFP file.

```
par map.ncd design_name.ncd design_name.pcf
```

Floorplanner Interface

The Floorplanner GUI consists of the primary Floorplanner window and four sub-windows labelled Floorplan, Placement, Design Hierarchy, and Design Nets. The primary window also contains pull-down menus, dialog boxes, a toolbar, and a status bar.

The Floorplanner GUI uses pull-down menus that contain all of the necessary commands to floorplan your design. The menus contain many commands that open dialog boxes, from which you can select various options and parameters for that command. Other commands act immediately on the selected logic.

Window operations, such as opening, closing, sizing, and moving are consistent with the window environment of your particular platform.

Toolbar

The toolbar is a feature on the Floorplanner that gives you push-button access to many tasks. You can zoom in and out of the Floorplan window and enable the display of resource graphics, labels, and ratsnest lines. It also gives you pushbutton access to changing the distribution direction for placing logic symbols.

You can use the toolbar buttons shown in the following figure instead of the pull-down menus to perform some of the basic operations in the Floorplan window. For a complete description of the toolbar buttons, refer to the Floorplanner online help.

Note The toolbar also contains buttons for a number of standard system functions, such as opening a file and printing. These buttons are not shown below.

Figure 3-4 Floorplanner Toolbar

Status Bar

The status bar is at the bottom of the primary window. In this area, the Floorplanner displays various resource information. To the far right it displays current row and column coordinates when the mouse pointer is in the Floorplan window.

The status bar also provides information about the toolbar buttons. When the toolbar is enabled and you move the mouse pointer over a toolbar button, the Floorplanner displays the name of that button and its function.

Mouse

The mouse is integral to many operations. Use it to select and place logic, access commands from the menus, and perform various window operations. Selecting logic in the Floorplanner windows is a "drag-and-drop" operation using the mouse pointer. You select logic from the hierarchical design in the Design Hierarchy window by placing the pointer over the symbol icon or hierarchical group name and clicking the left mouse button. Then you drag the pointer to the Floorplan window (a ghost image of the selected logic moves with the pointer) and release the mouse button to drop the logic at the pointer's location in the window.

To make multiple logic selections in the Floorplanner, use the standard windows procedures.

- To make multiple consecutive selections, click the left mouse button when the pointer is over the first selection. Then move the pointer to the last item, press and hold down the Shift key, and click the left mouse button.
- To make multiple nonconsecutive selections, click the left mouse button when the pointer is over the first selection. Then hold down the Control (Ctrl) key and click the left mouse button over each additional selection. Clicking on a selection that is already highlighted deselects that choice.

Keyboard

The Floorplanner uses the keyboard function keys that are mapped to specific menu commands and Floorplanner functions for ease of use. The following table lists the keyboard shortcuts for the Floorplanner

and shows the related toolbar button, if any. You should exercise care when using the keyboard shortcuts.

Table 3-1 Floorplanner Keyboard Shortcuts

Function Key	Menu Command/Function	Toolbar Button
F1	**Help → Help Topics**	None
F2	**View → Options**	None
F3	**Hierarchy → Group**	None
F4	**Edit → Colors**	None
F5	**View → Refresh**	None
F6	**View → Zoom to Selected**	
F7	**View → Zoom In**	
F8	**View → Zoom Out**	
F9	**View → Zoom to Box**	
F10	Change focus to the menu bar	None
F11	**View → Zoom Full View**	
Del	**Floorplan → Remove**	None
Esc	Cancel current operation	None

Dialog Boxes

The Floorplanner has many commands that, when invoked, open dialog boxes that contain default settings for command execution. These types of commands have an ellipsis (...) after the command name in the menu.

The dialog boxes are composed of the following elements.

- Edit boxes, in which you can type information such as a different path name
- List boxes, which list design information such as net names
- Buttons, which allow you browse information and easily change the way a command functions

The "Using the Floorplanner" chapter provides a detailed description of the Floorplanner dialog boxes in the command descriptions.

The Floorplanner Windows

The primary Floorplanner window contains four sub-windows: Design Hierarchy, Design Nets, Floorplan, and Placement. Descriptions of these windows follow.

Primary Window

When you invoke the Floorplanner, the primary window, shown in the following figure, is the first window to display on your monitor.

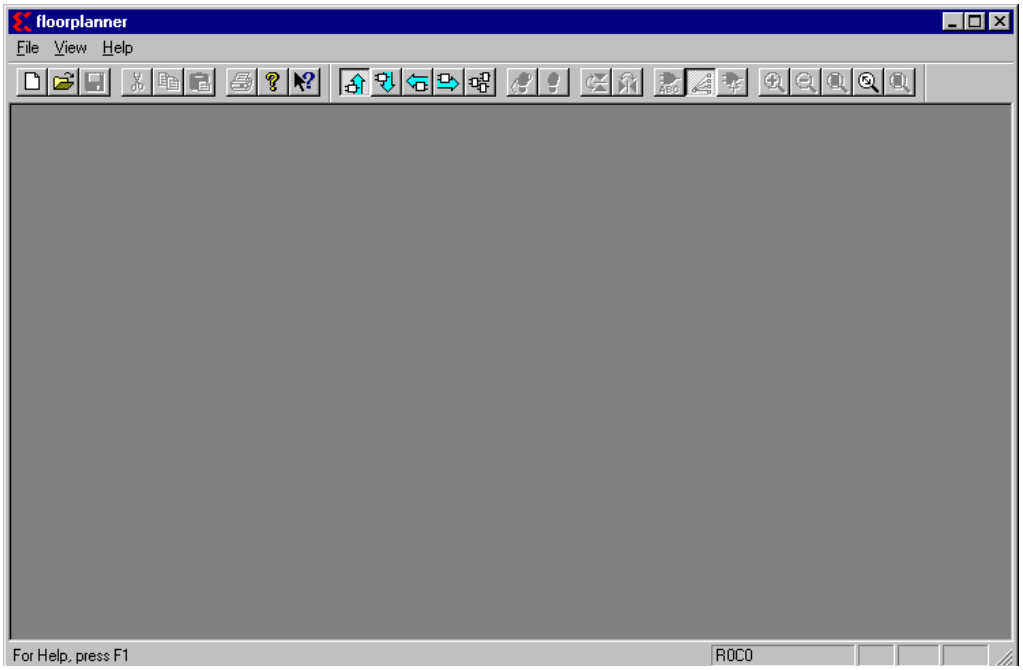


Figure 3-5 Floorplanner Primary Window

To begin floorplanning, select **File** → **New** to create a Floorplan file.

When you load a Floorplan file using the **File** → **New** or **File** → **Open** command, the Design Hierarchy, Design Nets, Floorplan, and Placement windows are opened, as shown in the following figure.

Note The Placement window only appears if PAR was run on the NCD file used to create the Floorplan file (FNF).

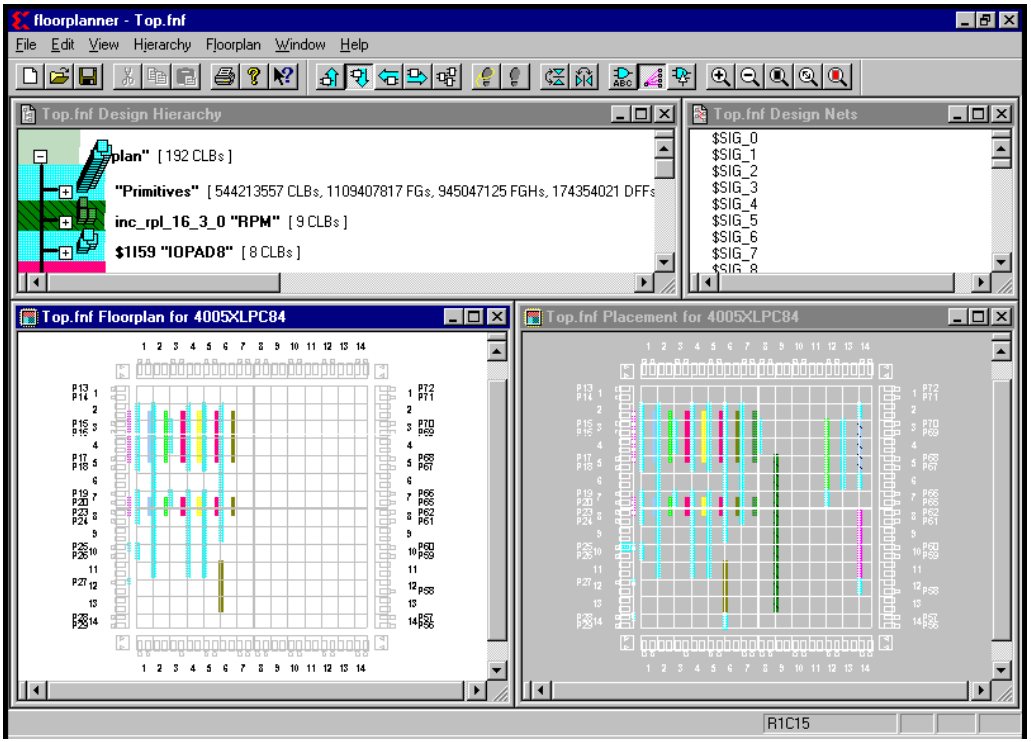


Figure 3-6 Floorplanner Windows

Design Hierarchy Window

The Floorplanner generates a hierarchical representation from the NGD and NCD input files. The Design Hierarchy window, shown in the following figure, displays a fully expandable and annotated hierarchy. The header line indicates the name of the design that is currently loaded.

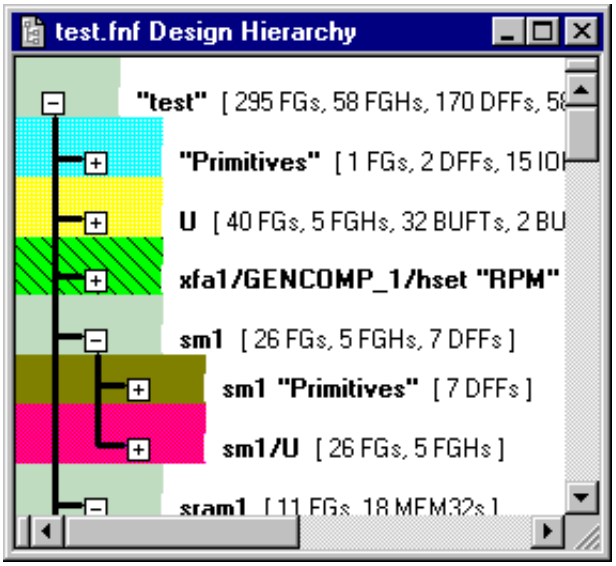


Figure 3-7 Design Hierarchy Window

Hierarchy Display

The Floorplanner uses colors in the hierarchy display to distinguish the levels in the hierarchy, which are annotated with the instance name from the design file. Hierarchy structure lines are black lines that traverse the colored areas and show the hierarchy of each hierarchical group. Each hierarchical group has a gray box with a minus sign, "–", or a plus sign, "+". The "–" indicates that the hierarchical group is expanded to show the next lower level of hierarchy. The "+" indicates that the hierarchical group is collapsed, and that lower levels of hierarchy exist for that hierarchical group.

Selecting Logic

When you place the mouse pointer over a hierarchical group instance name or its logic symbol icon (in the Design Hierarchy window) and click the left mouse button, you select that logic. The Floorplanner displays selected logic in the Design Hierarchy window in reverse video. (If Flashing is enabled, the selected logic in the Floorplan window flashes.) When you select a hierarchical group, you also select all the sub-hierarchy in that hierarchical group. When you select logic at some lower level, the Floorplanner draws a rectangular

box around all associated higher levels of hierarchy. The following figure shows an example. The $7I586/REG hierarchical group is the selected logic, as indicated by the reverse video. The hierarchical groups hpmvsel and $7I586/REG are the higher-level associated logic, as indicated by the box around those hierarchical group names.

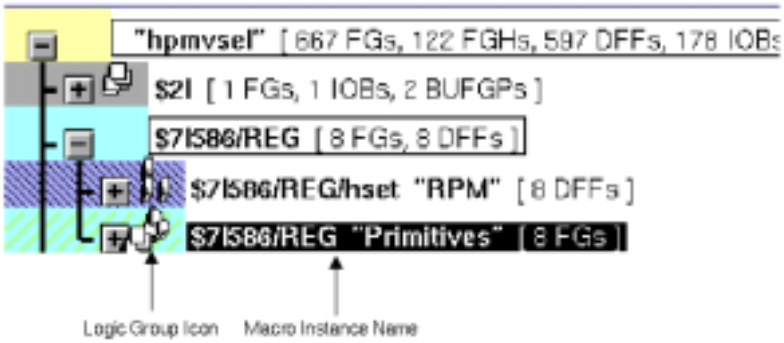


Figure 3-8 Selected Logic in the Design Hierarchy Window

Expanding and Collapsing Hierarchical Groups

Click the left mouse button on the Expand/Collapse button when it displays the "+" sign to expand a hierarchical group and display the next level of hierarchy. The logic elements that comprise each hierarchical group appear as an icon between the Expand/Collapse button and the instance name. The following figure shows a sample hierarchy.

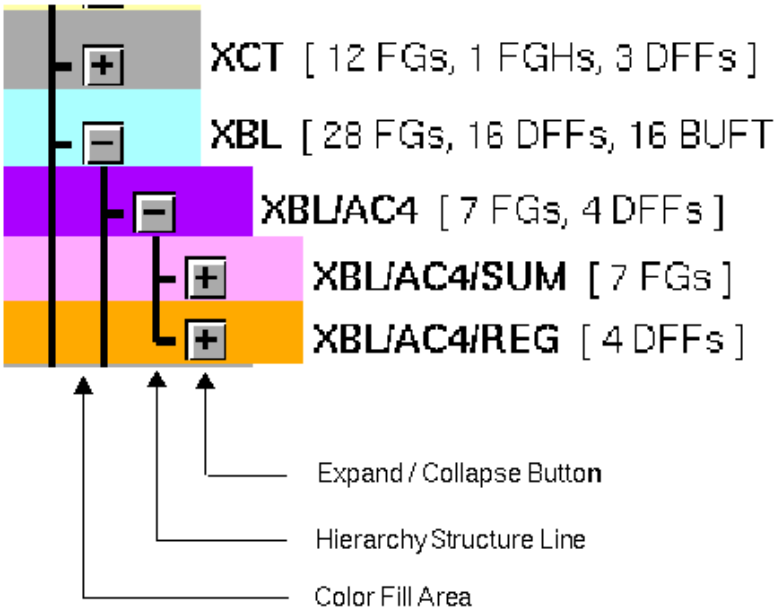


Figure 3-9 Hierarchical Group Display in the Design Hierarchy Window

Hierarchical Group Annotation

Each hierarchical group in the design hierarchy contains the instance name and symbol counts and, optionally, the group or source name. The symbol count is the number of FPGA resources required to accommodate that hierarchical group in the floorplan. Groups that you create with either the Group or Group By commands contain a non-hierarchical name and symbol count.

In the case of the XC4000 family, the FPGA resources are categorized by type, such as FG (function generator), BUFTs (tristate buffers), DFFs (registers), IOBs (input/output pads), RAM/ROM, and related logic. The following figure shows the important parts of a hierarchical group.

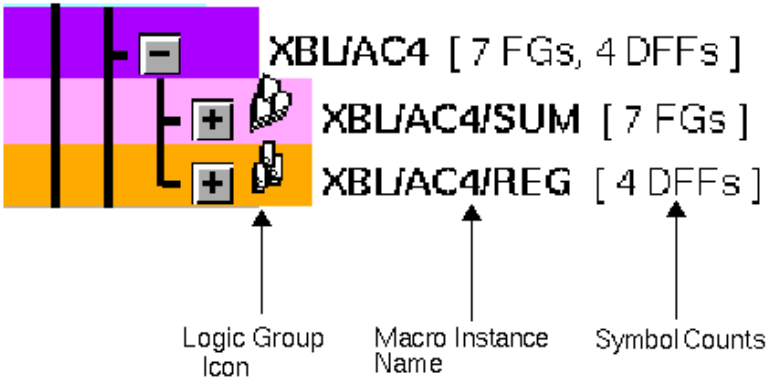


Figure 3-10 Hierarchical Group Annotation

Symbol Annotation

The symbol line represents the lowest hierarchy of any hierarchical group. It represents a single resource requirement of the design. The symbol line contains a logic icon, the symbol name, symbol type, and the list of pin:net pairs. The following figure shows one of the symbol instances of the expanded hierarchical group, XBL / AC4 / SUM (shown in the above figure). It labels the important parts of the symbol line.

Figure 3-11 Symbol Annotation

In this figure, the logic icon that is shown represents a single 4-input function generator (FG) named $FG_XBL / AC4 / S3; its output connects to net XBL / AC4 / S3; its inputs connect to the nets IA3, XBL / AC4 / SUM / C2, and XBL / Q43.

The symbol instance name is the corresponding symbol in the source netlist. For function generators or CLBs, it is a name that the mapping

software (MAP) provides. For other symbols, it is the name in the schematic that either you or the schematic entry tool chooses.

The symbol type refers to the type of resource that the symbol requires. Examples of symbol types are DFF (D-type flip-flop), IOB (Input / Output buffer), CLB (configurable logic block), FG (function generator), and BUFT (tristate buffer).

Design Nets Window

The Design Nets window lists the nets that connect the logic in the design. This window is shown in the following figure.

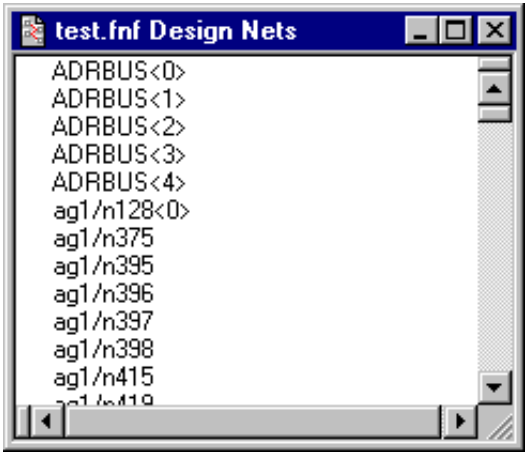


Figure 3-12 Design Nets Window

You can list either all nets in the design or just the nets that are currently displayed in the Floorplan window. To switch from one list to the other, click the right mouse button to bring up a menu and then click the **`List Visible Nets`** or **`List All Nets`** command in that menu.

When only the visible nets are listed in the window, the heading of the window changes to Visible Nets.

Floorplan Window

The Floorplan window displays the die for a selected part type, such as XC4005EPC84. This window, shown in the following figure, is a

scrollable, scalable view of a resource map of the device that is specified in the design.

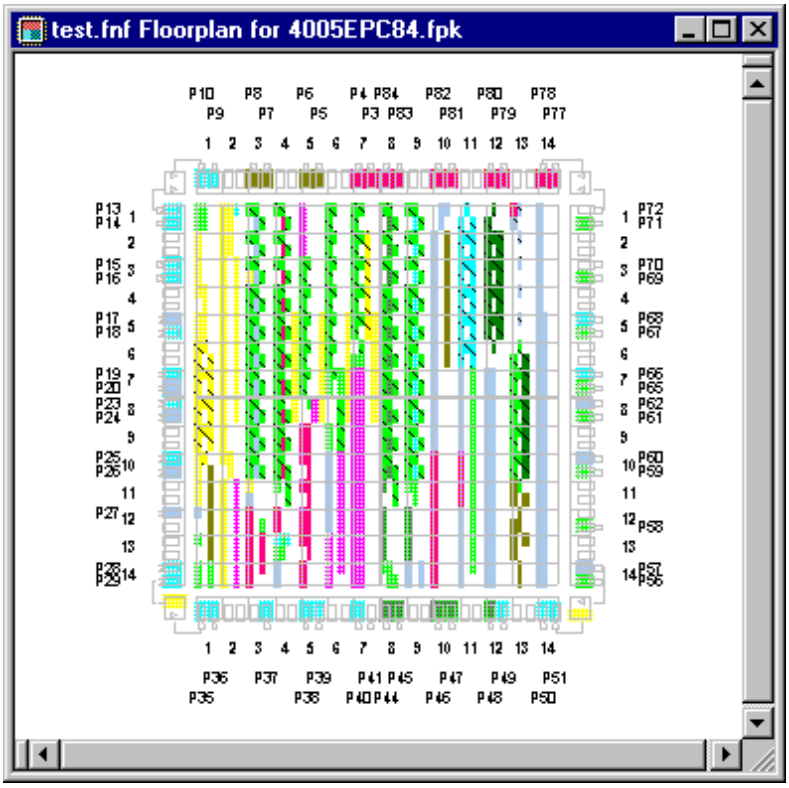


Figure 3-13 Floorplan Window

You floorplan by dragging selected logic from the Design Hierarchy window and dropping it into this window.

When a new Floorplan file (FNF) is created for a design and a previous FNF does not exist, the Floorplan window displays any physical design constraints that are in the initial design netlist, the UCF file, or the NCF file. If a previous Floorplan file does exist and is specified in the creation of the FNF file, the initial Floorplan constraints are generated from the previous FNF file. All netlist, UCF, and NCF constraints are ignored, unless they exist in the previous FNF.

Resource Graphics

Each device family architecture has specific resources on the die. You can display these resources using the Resources panel of the **View** → **Options** command. For example the resource graphics for the XC4000 family include I/O pads, function generators, registers, RAM/ROM, and BUFTs in the CLBs. With this feature, you control the view of the logic and available device resources. The fewer resources you display in the window, the faster the screen refreshes.

In the XC4000 family devices, flip-flops display as rectangles, function generators as trapezoids, and BUFTs as triangles. The following figure shows an example of the resource graphics available in a quadrant of the Floorplan window.

Note The global buffers have a pair of dedicated I/O pads that can also connect to other logic. The lines in the floorplan die show which I/O pad is dedicated to that buffer.

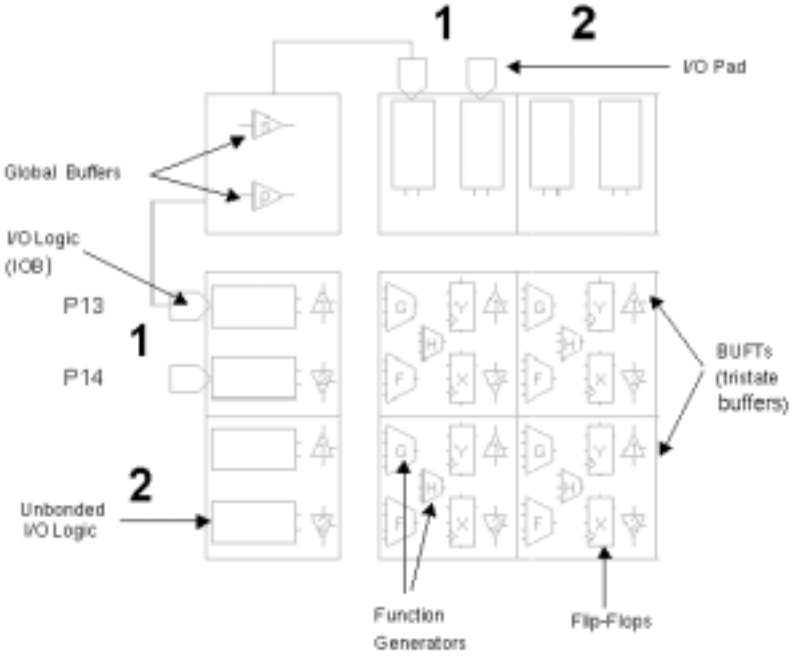


Figure 3-14 Resource Graphics for an XC4000 Device

Placement Window

The Placement window displays the design after it has been placed and routed. This window is shown in the following figure. This is a very useful function to help evaluate the properties of an automatically generated placement.

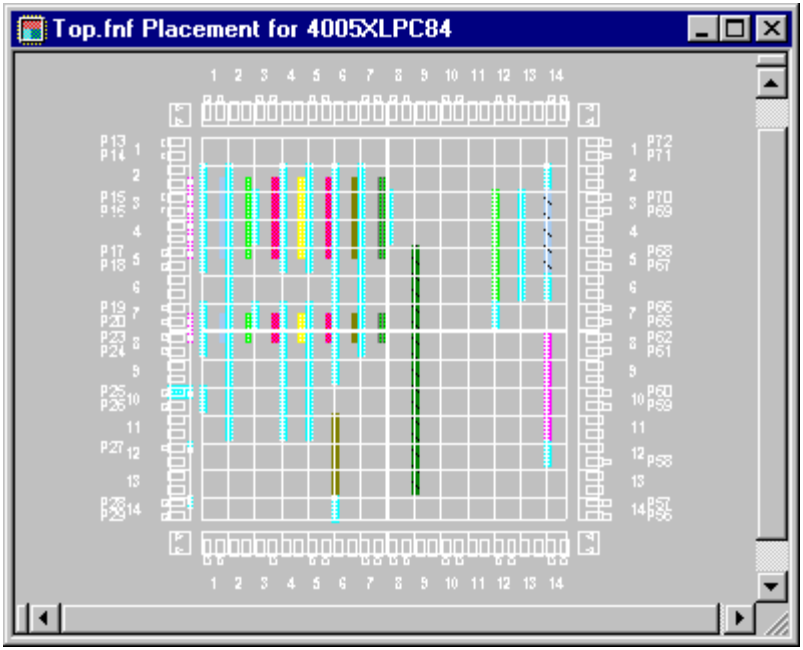


Figure 3-15 Placement Window

The elements in the Placement Window cannot be moved or modified without loading in new placement information from a placed NCD file using the **File** → **New** or **File** → **Update** commands. To translate placement information into Floorplan constraints, use the **Floorplan** → **Constrain From Placement** or **Floorplan** → **Replace All With Placement** commands.

Closing the Current Design

To close the current design without exiting the Floorplanner, select the **File** → **Close** command. If you have not made any changes to the current design, the windows that contain the design close while the Floorplanner window remains open.

If you have made any changes to the design in the Floorplanner, a dialog box prompts you to save those changes. Click the appropriate button.

- If you click **Yes**, the Floorplanner writes out an FNF file that is a snapshot of the floorplanned logic and the Floorplanner environment. In addition, the Floorplanner runs a utility to create an MFP file containing all physically constrained elements in the FNF file. The current design then closes.
- If you click **No**, the current design closes without saving the changes to the design.
- If you click **Cancel**, the dialog box closes and you can continue with the current floorplanning session.

Exiting the Floorplanner

To exit from the current floorplanning session, select **File** → **Exit**. If you have not made any changes to the design, the Floorplanner window closes.

If you have floorplanned any logic, a dialog box displays and prompts you to save the changes to the current floorplan. Click the appropriate button.

- If you click **Yes**, the Floorplanner writes out an FNF file that is a snapshot of the floorplanned logic and the Floorplanner environment. In addition, the Floorplanner runs a utility to create an MFP file containing all physically constrained elements in the FNF file. The Floorplanner window closes.
- If you click **No**, the Floorplanner window closes without saving the changes to the current design.
- If you click **Cancel**, the dialog box closes and you can continue with the current floorplanning session.

Chapter 4

Using the Floorplanner

This chapter provides step-by-step instructions for performing the important floorplanning tasks.

These procedures are presented in sequential order; however, not every task that can be performed is documented in this chapter. Floorplanning a design may require you to repeat some procedures several times. Consult the design flows in the "Design Flow" chapter to aid your floorplanning efforts.

For information on starting and exiting the Floorplanner, see the "Getting Started" chapter.

This chapter contains the following sections.

- "Opening a File" describes how to load a design file into the Floorplanner.
- "Saving a File" explains how to save your floorplanned design.
- "Using Colors" explains how to assign different colors to the hierarchical groups display.
- "Floorplanning Logic" describes how to move logic symbols from the Design Hierarchy window to the Floorplan window.
- "Floorplanning Designs that Contain RPMs" explains how to floorplan designs that contain Relationally Placed Macros (RPMs).
- "Creating Groups" explains how to rearrange and regroup the design hierarchy.
- "Using Area Constraints" describes how to create area constraints.
- "Using the UCF Flow"describes how to use the UCF flow to write constraints to a UCF.

- "Flattening and Building the Hierarchy" describes how to flatten and rebuild the design hierarchy.
- "Walking Through the Design" describes how to select successive logic by connectivity rather than by name.
- "Analyzing PAR Placement" describes how to analyze the placement results that PAR generates.
- "Analyzing PAR Placement for Timing Constraints" describes how to analyze the placement of floorplanned logic by PAR with respect to Timing Constraints.
- "Finding Logic Connected to Nets" explains how to find logic that is connected to nets in the floorplanned design.
- "Displaying Resources and Logic" describes how to display the logic resources that are available on the FPGA.
- "Performing Detailed Manual Placement" explains how to manually place logic into the Floorplanner window.
- "Checking the Floorplan" describes how to check the floorplanned logic for placement problems.
- "Aligning Symbols" describes how to align symbols in the Floorplan window to reduce unnecessary routing.
- "Working with Patterns" describes how to work with patterns when you place selected logic in the Floorplan window.
- "How to Interleave Buses" describes how to interleave buses in the Floorplan design.
- "Iterative Floorplanning" explains how to floorplan your design iteratively.
- "Floorplanning Incremental Schematic Changes" describes how to make incremental changes to a design that has been previously implemented in an FPGA.
- "Making Small Modifications to Automatic Placement" describes how to make small changes to an automatically placed design in order to fix packing or placement problems.
- "Lock Down I/Os from Automatically Placed Design" describes how to select the I/Os from an automatically placed and routed design and lock them down in the Floorplan window.

- "Getting Started With an Unfamiliar Design" describes how to familiarize yourself with the connectivity of someone else's design.

Opening a File

To load a design file in the Floorplanner, follow these steps.

1. Select **File** → **New** or **File** → **Open**.

   The **File** → **New** command opens the New Floorplan dialog box. You can choose either a standard or UCF flow. You enter the filename of an appropriate NCD, NGD or FNF file.

   The **File** → **Open** command opens the Open File dialog box where you specify which FNF or NGD file to load. In this dialog box, you can also change directories if the desired file is in a directory other than the current directory.

2. Browse until you find the desired design file in the list.

3. Double-click on the design file name, or enter the filename you want. Click **Okay** or **Open** to load the file and open the floorplan window.

   The Floorplanner reads the file, loads the correct device (part type), opens the Design Hierarchy window with a hierarchical design, and opens the Floorplan window with the correct FPGA die.

Saving a File

To save your floorplanned design, select **File** → **Save**.

The Save command creates a file with the same name as the current design and the extension .fnf. The information stored in this file includes the design's hierarchy organization, floorplanned logic, and net, logic, and color assignments.

This file represents a snapshot of the current state of the floorplan that you can use later.

In addition, whenever the FNF file is saved, the floorplanner automatically creates an MFP file. The MFP file contains all the physical constraints for the design that appear in the Floorplan window, and is used as an input to MAP.

Using Colors

The Floorplanner automatically assigns unique colors to hierarchical groups when it reads a new design. Hierarchical nodes that have one lower level of hierarchy are set to the color of that lower level node. Hierarchical groups with more than one lower level of hierarchy, as well as individual symbols, are not assigned colors. Individual symbols without assigned colors inherit the color of the lowest level of associated hierarchy.

You can change the colors of any hierarchical group using the **Edit** → **Colors** command. If you assign a color to a symbol, that symbol is always shown in that color in both windows. If you assign a color to a hierarchical group, the colorless symbols under that node will appear in that group's color.

Note You can remove color assignments using the **Auto Assign** button in the Colors dialog box.

The following sections describe using colors in the Design Hierarchy window and in the Floorplan window. The last section describes how to use colors to distinguish between floorplanned logic and place and routed logic.

In the Design Hierarchy Window

1. Select the hierarchical groups and symbols for which you want to change colors.
2. Select **Edit** → **Colors**.

   This command opens the Edit Colors dialog box.
3. Click the button that displays the color you want to use.
4. Click **Apply** to change to the new color.

In the Floorplan Window

1. Select the logic by dragging out an area around the desired logic for which you want to change colors, or click the left mouse button on an individual symbol.
2. Select **Edit** → **Colors**.

   This command opens the Edit Colors dialog box.

3. Click the button that displays the color you want to use.
4. Make sure the **Apply to Symbols** radio button is selected.
5. Click **Apply** to change to the new color.

Distinguishing Logic

You can use the **Edit** → **Colors** command to distinguish placed and routed logic and floorplanned logic. You can change the color of the floorplanned logic to a color not used in the hierarchy. Then, when you view the placement window, the unique color distinguishes the placed and routed logic from floorplanned logic in the design.

1. Select all the logic in the design.
2. Select **Edit** → **Colors**.
3. Choose a new color for the selected logic from the palette in the Edit Colors dialog box.

   Note When colors are automatically assigned, the first two and the last two colors are not included. You can use one of those colors to make the logic distinguishable from the rest of the design.
4. Click **Apply**.
5. Select the floorplanned logic by dragging the mouse pointer over the entire Floorplan window.
6. Choose a different color from the palette in the Edit Colors dialog box for the selected floorplanned logic.
7. Click **Apply**.
8. If the NCD file that was used to create the Floorplanner file contained placement information, selecting **View** → **Placement** will show the entire placed design, with the Floorplanned logic in a different color than the non-floorplanned logic.

Note You can use the **Auto Assign** button in the Edit Colors dialog box to return to a normal display.

Floorplanning Logic

This procedure explains how to select, move, and manipulate logic symbols from the Design Hierarchy window to the Floorplan window.

1. Select the desired logic from the Design Hierarchy window. There are two ways to do this.

   - Using the mouse, place the pointer on the logic group icon of the desired hierarchical group and click the left mouse button.

Figure 4-1 Select an Icon Stack from the Design Hierarchy Window

   - Select **Edit** → **Find** to find and select the desired logic. In the Find dialog box, you can type in the instance name, choose a specific type of logic, such as I/O Pads for IOBs, Flip-Flops for DFF, or type of connection.

     Click **Find**.

     If the search criteria that you applied is correct, an arrow will point to the applicable instance or hierarchical group in the Design Hierarchy window.

2. Use the mouse and click on the found logic or hierarchical group icon.

3. If you want to move an individual piece of logic, expand the hierarchical group and click on the desired logic icon.

   The logic icons change to a ghost image as you move the mouse pointer.

   When moving more than one icon at a time, you must use one of the four directional arrow toolbar buttons to determine the distribution direction. The default direction is from top to bottom.

4. Move the mouse pointer from the Design Hierarchy window to the desired location on the FPGA in the Floorplan window.

   A ghost image of the selected logic icons showing the allocation direction moves with the pointer.

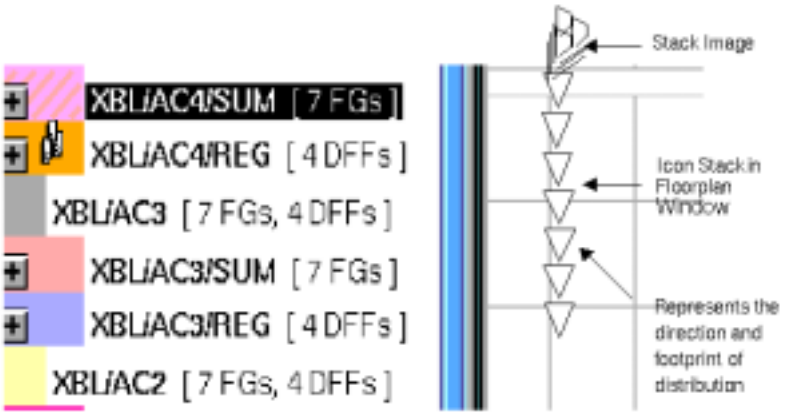


Figure 4-2 Icons in Transit from the Design Hierarchy Window to the Floorplan Window

5. At the desired location in the Floorplanner window, click the left mouse button to place the logic. The Floorplanner places the logic according to the distribution direction you have chosen.

   If you have Ratsnest turned on, you will see black lines indicating logic connectivity to the group you have just selected and placed.

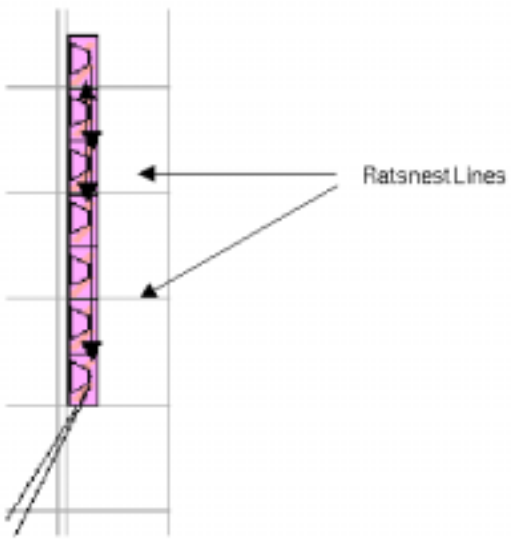


Figure 4-3 Logic Icons Placed Using Distribute Mode; Ratsnest Turned On

Floorplanning Designs that Contain RPMs

This procedure explains how to floorplan designs that contain Relationally Placed Macros (RPMs). RPMs are optimized macros that have the relative positions of the logic locked down. For this reason, you can only place RPMs as a whole unit into the Floorplan window.

You must have enough resources in the Floorplan window to accommodate the entire RPM, because it must be placed in its entirety. For example, if an RPM is four CLBs in height and the largest vacancy in the die is 3 CLBs high, the RPM will not fit.

Perform the following steps to place an RPM in the Floorplan window.

1. Select the desired RPM from the Design Hierarchy window.
2. Place the selected RPM into the Floorplan window in an area that can accommodate the entire RPM.

Note A good floorplanning practice is to floorplan both the RPM and the logic it is driving in a specific set of locations.

RPM Binding

The Floorplanner now allows you to create an RPM graphically. You can select logic symbols and bind them into an RPM. You use the **Pattern → Bind** and **Pattern → Unbind** commands to create and undo RPMs.

Note You cannot bind and unbind RPMs in a UCF flow.

- Select the desired logic symbols from the Design Hierarchy window.
- Select **Hierarchy → Group** to group the logic symbols.
- Place the group by dragging the group elements to the Floorplan window. The logic remains stacked after you lay it out.
- Select the **Pattern → Bind** command to create the new RPM.

Note At this point, you can drag the newly created RPM from the Floorplan window to the Design Hierarchy window. When you run PAR, the RPM gets placed. If you leave the RPM in the Floorplan window, it is locked in place as seen in the Floorplan window. The RPM remains bound unless you choose to unbind it.

To unbind an existing or new RPM, follow these steps.

- Select the RPM you wish to unbind in the FloorPlan window.
- Select the **Pattern → Unbind** command to unbind the RPM into individual logic symbols.

Caution

Unbinding RPMs that contain carry logic is risky. You may delete constraints that MAP and PAR will later require to implement your design.

Creating Groups

The hierarchical representation of your design is a result of mapping your original design; it might not be optimized for your floorplanning preferences. To make floorplanning easier, you can rearrange and regroup the design hierarchy.

Note The function or connectivity of logic elements may be a reason for you to place them together as a group.

Manual Grouping

To create a group manually, follow these steps.

1. In the Design Hierarchy window, select the logic that you want to group.
2. Select the first piece of logic with the left mouse button and subsequent logic with the middle mouse button. Or, select subsequent logic by holding down the control key and pressing the left mouse button.
3. Select **Hierarchy** → **Group**, or press the F3 key.

   The Floorplanner creates a new group and assigns an arbitrary name to the group "GRP0". The text line looks as follows.

   `GRP0 "Grouped by: User"` [*symbol count*]

   [*symbol count*] is the number of logic elements

   The Floorplanner labels subsequent new user-created groups "GRP1", "GRP2", and so on.
4. Use the **Edit** → **Properties** command to give the new group a better name.

   The newly created group occupies a position in the lowest level of hierarchy that is common to all logic that comprises the group.

The following figures illustrate how to create a new group in the design hierarchy.

The first figure shows the four D-type flip-flops that appear in reverse video. These are the logic symbols that have been selected to form a new group.

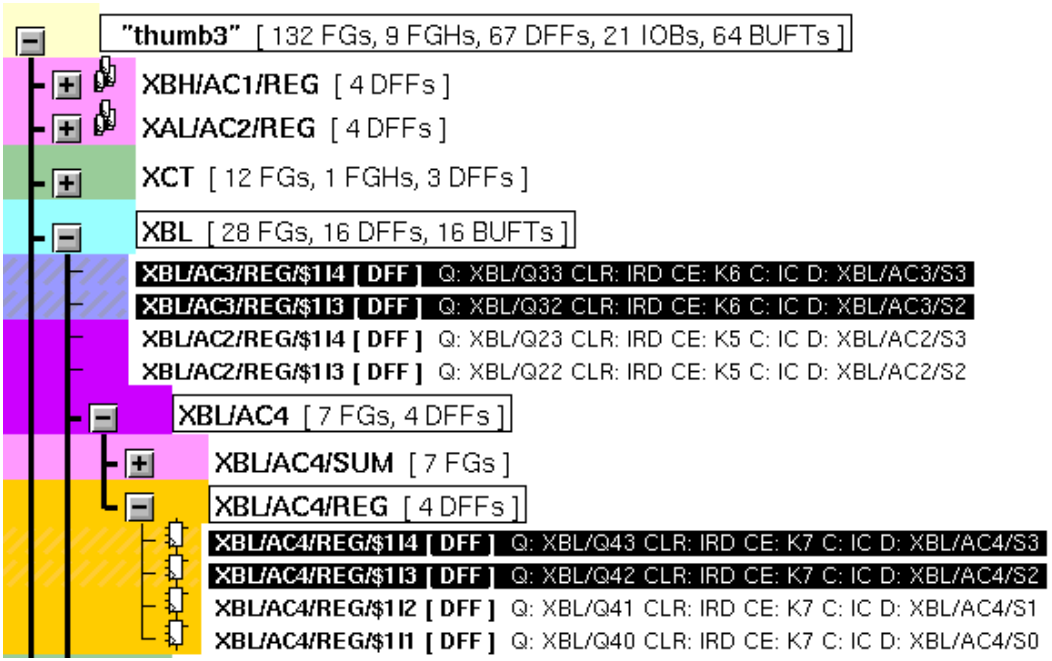


Figure 4-4 Select Logic for Grouping

The next figure shows the new group. The new group is named GRP0 (highlighted in reverse video). It is the first new group to be created in the design. Note that the four flip-flops are no longer part of XBL and AC4, respectively, as indicated by the new symbol counts.

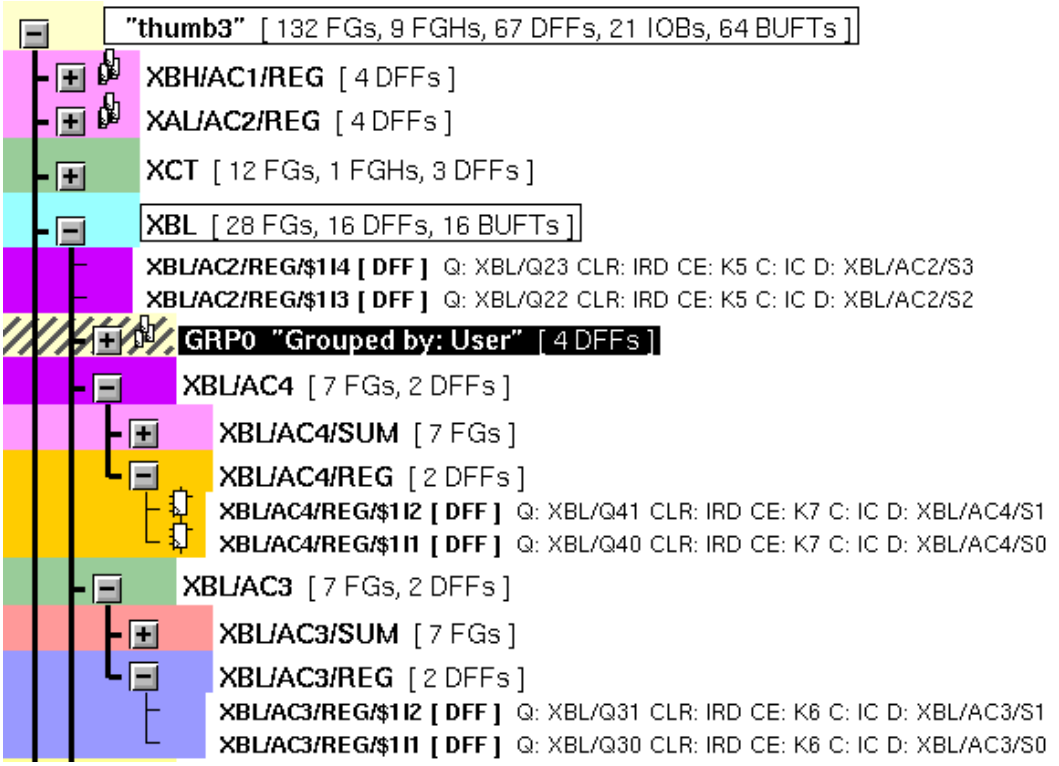


Figure 4-5 The Floorplanner Creates New Group

The Floorplanner places the new group in the hierarchy near the group XBL, the lowest level of hierarchy that is common to the four flip-flops chosen for the group. When you expand the new group, you see that the four flip-flops are now in the new group.

The following figure shows the new group expanded. You can place this new group as a unit on the floorplan die.

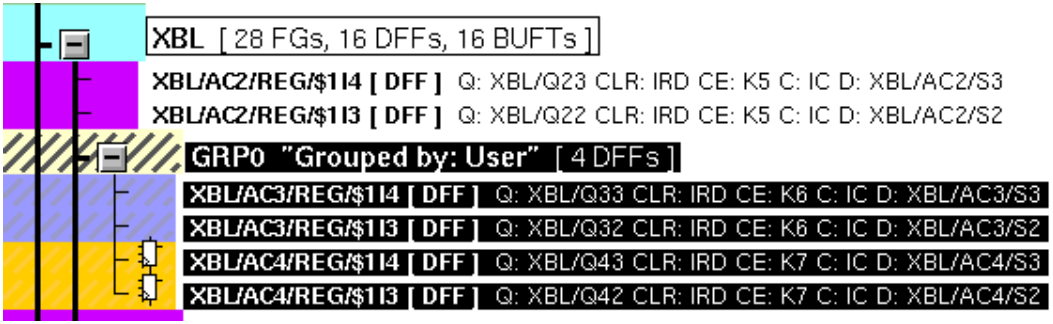


Figure 4-6 GRP0 Expanded to Show Logic Elements

Automatic Grouping

It is often more convenient to group logic by common type or signal names. Creating such groups can make it easier to floorplan.

For example, you might want to make a group of all the BUFTs that have a common output enable. In this case, do the following steps.

1. Select **Edit** → **Find**.

   This command opens the Find dialog box.

2. Select a BUFT in the Design Hierarchy window that contains the enable signal of interest by clicking the mouse on the appropriate line (on the BUFT's instance name).

3. In the Connections list box, select Common Enable (BUFTs, DFFs, IO).

4. Click **Find**.

5. Click **Select Found**.

   The Floorplanner searches throughout the design hierarchy and highlights each BUFT that meets the criteria you selected.

   Note The Floorplanner automatically expands groups that have sub–hierarchy that meet the find criteria.

6. Select **Hierarchy** → **Group**, or press the F3 key.

   The Floorplanner creates a new group and assigns an arbitrary name to the group "GRP0" (if this is the first user-created group). The text line looks as follows.

`GRP0 "Grouped by: User"` [*symbol count*]

[*symbol count*] is the number of logic elements.

Note You can also use the **`Hierarchy` → `Group By`** command to create these types of groups.

Using Area Constraints

Area constraints are a way of restricting where PAR can place a particular piece of logic. By reducing PAR's search area for placing logic, PAR's performance may be improved.

To create an area constraint, do the following steps.

1. Select a hierarchical group in the Design Hierarchy window.
2. Select **`Floorplan` → `Assign Area Constraint`** or click on the toolbar button.
3. In the Floorplan window, use the mouse to drag out a rectangular box where you want the area constraint to be located. The area constraint will cover all the tiles that are inside the drag box.

Area constraints may overlap each other. Select **`Floorplan` → `Bring Area To Front`** or **`Floorplan` → `Push Area To Back`** to move a selected area constraint in front of or behind another.

Using the UCF Flow

You can use the Floorplanner to create a User Constraints File (UCF). This feature allows you to floorplan at an earlier stage of the design process, before or just after design entry. The UCF flow causes the Floorplanner to write constraints to the UCF instead of producing an MFP file as output.

When using the UCF flow, remember the following points.

- Avoid constraining your design in the source design schematic or HDL. The Floorplanner can add, remove, and edit constraints to an existing UCF, but cannot modify constraints elsewhere in the design source (including those in the NCF).
- Do not attempt RPM binding. The UCF does not support RPM unbinding. Therefore, the **`Bind`** and **`Unbind`** commands in the Pattern menu are disabled.

- Remove any previously generated MFP files before implementing your design. If the design directory includes an MFP file from previous floorplanning (using the standard flow), the Design Manger may detect it and invoke MAP to use the previously generated MFP file. This causes any UCF constraints to be ignored.

To start a UCF flow, follow these steps.

- Select the **File** → **New** command to display the New Floorplan dialog box.
- Select UCF Flow instead of the default Standard Flow. The two radio buttons, UCF Flow and Standard Flow are at the top of the dialog box.
- Type an FNF or NGD filename. If you type an NCD filename, make sure you enter the filenames for either an FNF or NGD file as well.

  The Floorplan window opens with the label **UCF Flow**. Some menu commands, such as **Bind** and **Unbind** are disabled for the UCF flow; others such as **Change Device** and **Add Block** in the Edit menu apply specifically to the UCF flow.

To save a UCF Floorplan, follow these steps.

- Select **File** → **Save** or **File** → **Save As**. The Save Constraints As dialog box displays.
- Click **Okay** to save the newly created or edited UCF in the current directory.

Creating UCF Constraints for IOB Placement

The Floorplanner now offers a graphical way to make and edit IO pin assignments in the UCF. When you write these constraints to a UCF, the Floorplanner applies them to the design along with any existing constraints in the design source files.

To graphically create new UCF constraints for IOB placement, follow these steps.

- Start a UCF Flow by selecting **File** → **New** to display the New Floorplan dialog box.

  Note Alternatively, you can select **File** → **Use UCF Flow**.

- Select UCF Flow instead of the default Standard Flow. The two radio buttons, UCF Flow and Standard Flow, are at the top of the dialog box.
- Type an FNF or NGD filename. If you type and NCD filename, make sure you enter the filename for either an FNF or an NGD file as well.

  Note You can also type an NCD filename, but it is not required.
- Select the desired IOB logic symbols in the Design Hierarchy window and drag them to the Floorplan window.
- Place the symbols along the periphery of the Floorplan window, thus creating IOB constraints.

  Note The Floorplanner performs only minimal rule checking while placing IOB symbols. It does not check to ensure that IO pins driving the global clock buffers are in the right locations. Refer to the device datasheets while deciding on pin locations.
- Select **`File`** → **`Write Constraints`** to write the newly created IOB constraints to the UCF.

Flattening and Building the Hierarchy

You may have floorplanned the design and saved it. Then you realize that you must make a change to the design at the schematic level that deletes some logic. When you read in the FNF file, the Floorplanner reads in the old hierarchy. Due to the changes made at the schematic level, the old hierarchy may represent divisions that are no longer valid.

How to Flatten the Hierarchy

You must select a hierarchical node to enable the Flatten Groups command.

This procedure explains how to flatten the hierarchy in the Design Hierarchy window.

1. Select the part of the design hierarchy that you want to flatten. You can select just the level of hierarchy that contains the logic that was changed in the schematic.

   To flatten the entire hierarchy, select the top-level hierarchy.

2. Select **Hierarchy** → **Flatten Groups**.

   The Floorplanner removes all lower level hierarchy from the selected hierarchical node and moves all of the associated symbols up in the hierarchy.

How to Rebuild the Hierarchy

To rebuild the hierarchy using symbol instance names, select **Hierarchy** → **Rebuild**. The Floorplanner builds a hierarchy tree and places the selected logic symbols in their proper positions in that tree.

Note The Rebuild command works on all logic, not just selected logic.

Walking Through the Design

This procedure explains how to select successive logic by connectivity rather than by name. This process helps you see which floorplanned logic elements drive other logic in the design.

Finding Logic

1. Determine the logic that is of interest.
2. Select **Edit** → **Find**.

   This command opens the Find dialog box.
3. In the Connections list box, select either Loading Selected Logic or Driving Selected Logic.
4. Click **Find**.
5. Click **Select Found**.

   The Floorplanner searches the design for the appropriate symbols. The Floorplanner highlights the first “found” symbol in the Design Hierarchy window with an arrowhead at the left of the symbol line text. Matching symbols in the Floorplan window flash (blink) if you have enabled Flashing.

Finding Nets

1. Determine the nets that you want to find.
2. Select **Edit** → **Find**.

This command opens the Find dialog box.

3. In the Type list box, select Nets.
4. Use the Name dialog box, or Connections list box to find the desired net(s).
5. Click **Find**.
6. Click **Select Found**.

   For logic that is placed in the Floorplan or Placement windows, the Floorplanner searches the design and highlights the appropriate nets by drawing them as red lines. Note that only currently selected nets appear in red; previously found nets, which were deselected, are drawn with black lines.

Displaying the Ratsnest

The Floorplanner shows connectivity between logic symbols in the Floorplan window with the aid of the ratsnest display. A ratsnest is a display of lines between logic blocks that indicate connections between the inputs and the outputs of floorplanned logic.

By default, the Floorplanner automatically displays ratsnest for selected floorplanned logic and selected nets. Whenever you place selected logic in the Floorplan window, the ratsnest displays the connectivity between the logic symbols.

Using the Ratsnest

By default, the ratsnest is on. As you place selected logic in the Floorplan window, the ratsnest lines show connectivity. You can see which logic is connected to a given net by selecting nets in the Ratsnest dialog box and choosing the appropriate connection.

1. Select some floorplanned logic. Use the mouse to draw an area around the desired logic or click on an individual piece of logic.

   The Floorplanner draws the ratsnest to the associated logic.

2. In the Design Nets window, use the right mouse button to click **List Visible Nets**. The names of the nets connected to the selected logic appear in the window.
3. Select a net in the Design Nets window and look at the Floorplan window.

The Floorplanner draws a ratsnest of the selected net in red.

Viewing Selected Nets in the Ratsnest

Use the Design Nets window to display specific nets in the Floorplan window. Click on a net in the Design Nets window. The Floorplanner window displays the net in red lines. To display consecutive nets in the Design Nets window, hold down the Shift key while selecting the desired nets. To display non-consecutive nets, hold down the Control key while making your selections.

Analyzing PAR Placement

This procedure describes how to analyze the placement results that PAR generates. You must first run PAR to generate an NCD file with placement information.

1. Select **`File`** → **`Update`** and read in the NCD and NGD files.
2. Select **`View`** → **`Placement`** in order to display the Placement window.

   Note If the Placement window is empty, this indicates that the NCD file used to generate the Floorplanner netlist did not contain placement information. Make sure that you have run PAR and selected **`File`** → **`Update`**, specifying the placed NCD file.

3. Examine the Placement window for the following potential problems.

   - BUFTs with common output enables that are not vertically aligned or cross over longlines midpoints
   - RAM that is misaligned so as to prevent control signals from being routed on longlines
   - RPMs, BUFTs, and other groups source and load nets are close together
   - Structured logic elements that cross longline splitters into another quadrant

   Refer to the following figure. Note the positions of the dark–colored BUFTs on the left. The placement is inefficient because two different longlines are required for the output enable signal.

These BUFTs are misaligned because they are in two columns, which require different longlines, and are split between two different quadrants, crossing longline midpoints. Now, look at the light–colored BUFTs on the right. These BUFTs represent proper alignment in the FPGA because the common output enable signals connect to the same vertical longline.

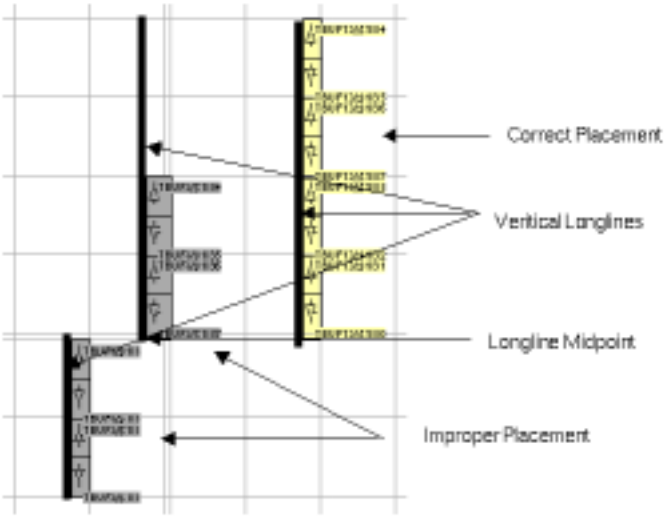


Figure 4-7 BUFT Placement

The following figure illustrates flip-flop to BUFT alignment in the FPGA. The left half of the example shows ineffective placement; see how the ratsnest lines intersect. The right half of the example shows the proper alignment of the flip-flops and their associated BUFTs as indicated by the parallel ratsnest.

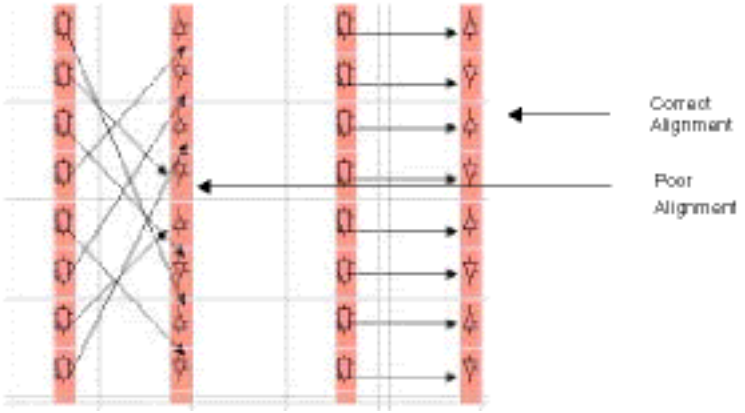


Figure 4-8 Flip-flop/BUFT Alignment Example

Note For clarity, the flip-flops and BUFTs occupy two different tiles. Normally, you would place this logic in the same tile to minimize the length of the ratsnest.

The following figure shows an example of groups placed so that the source and load nets are in proximity. In many cases, where there is sufficient routing resources, you want to place source and load nets close together.

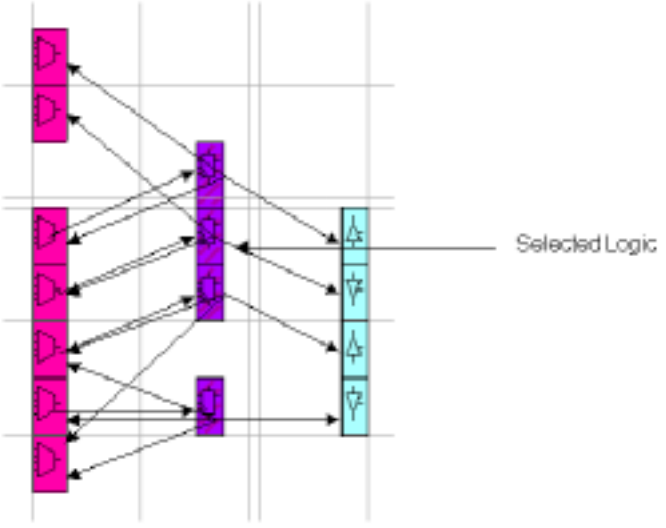


Figure 4-9 Viewing Placed Logic Symbols Using the Ratsnest

Note In the example, the middle group is the selected logic from which the ratsnest shows its source nets and load nets.

4. To see that PAR placed group symbols efficiently, select them from the Placement window or the Design Hierarchy window.
5. Step through the design, selecting critical groups, and using the Ratsnest to view the distance between selected groups and their sources and loads.

   The Floorplanner highlights the selected symbols in the Placement window.

Analyzing PAR Placement for Timing Constraints

This procedure explains how to analyze the placement of floorplanned logic by PAR with respect to Timing Constraints. If you are using the Design Manager, follow the instructions in that section.

You should analyze the logic placement in the following sequence.

1. Examine the PAR report for Timing Constraints that were not met.
2. Use Timing Analyzer to find those Timing Constraints that were not met.
3. Use the Floorplanner ratsnest to view the path.
4. If sub-optimal placement is causing the timing constraints to fail, floorplanning constraints can be used to improve the placement of logic to shorten time delays.
5. Rerun MAP and PAR with the new floorplanning constraints.

From the Design Manager

If you are running the Floorplanner from the Design Manager, follow these steps to analyze the placement with regard to your Timing Constraints.

1. Select a routed implementation revision from the Project View.
2. Open up the Report Browser in one of two ways.
   - Select **`Utilities`** → **`Report Browser`**.
   - Click the **`Report Browser`** toolbar button.

3. From the Report Browser, double-click on the Post Layout Timing Report icon.
4. Examine the report for any missed Timing Constraints.
5. If you find that there are missed Timing Constraints in your design, click the **Timing Analyzer** toolbox button in the Design Manager.

   This step opens the Timing Analyzer window that displays the report file.
6. Select **Analyze** → **Timing Constraints** in the Timing Analyzer window.
7. Select **File** → **Save** in the Timing Analyzer window to save the timing report.

   Following is a portion of a sample TRCE report.

```
Data path myaddr21 to syn294866 contains 4 levels of logic:

Path starting from Comp: IOB.PAD
To              Delay type        Delay(ns)   Physical Resource
                                              Logical Resource(s)
------------------------------------------- ------------------
IOB.I1          Tpid                1.089R    myaddr21
                                              myaddr21
                                              my_addr21
CLB.G3          net (fanout=3)      2.984R    my_addr21
CLB.Y           Tilo                1.190R    N279
                                              syn295958
CLB.G4          net (fanout=4)      2.984R    syn295958
CLB.X           Tiho                1.959R    syn294869
                                              syn323918
                                              syn294869
CLB.G1          net (fanout=2)      2.984R    syn294869
CLB.K           Tihck               1.640R    syn294866
                                              syn323905
                                              my_next_state<3>
                                              my_current_state<3>
------------------------------------------------
Total (5.878ns logic, 8.952ns route)14.830ns
```

Using Find and Ratsnest to Find Critical Nets

With the edited TRCE report, you can use the Floorplanner **Find** command to display the paths that need fixing.

1. Select **Edit** → **Find** in the Floorplanner window.
2. In the Find dialog box, set the Type field to Nets. Turn Auto Goto on and Auto Select off.
3. In the Find dialog box, type the name of the first net in the TRCE report into the Name field. In the above example, the first net (identified by the first occurrence of the word "net" in the Delay type column) is my_addr21.
4. Click **Find** in the Find dialog box.
5. In the Design Nets window an arrow appears next to the net specified in the Find dialog box. Select this net. The net is displayed in the Placement window.
6. Repeat steps 3 through 5 for each net listed in the TRCE report. In the Design Nets window hold down the Control button when you select each additional net. This ensures that the previous nets remain selected.

   When done, the Placement window shows the full path described in the TRCE report.

   Note You can also identify nets that have long delays by looking at the TRCE report and searching for the net by name.

   The Floorplanner displays the ratsnest in red. The length of the ratsnest does not correlate to a specific time delay. However, by moving logic blocks to shorten the ratsnest, you can improve on the delays.

   The following figure shows the ratsnest for a failed Timing Constraint.

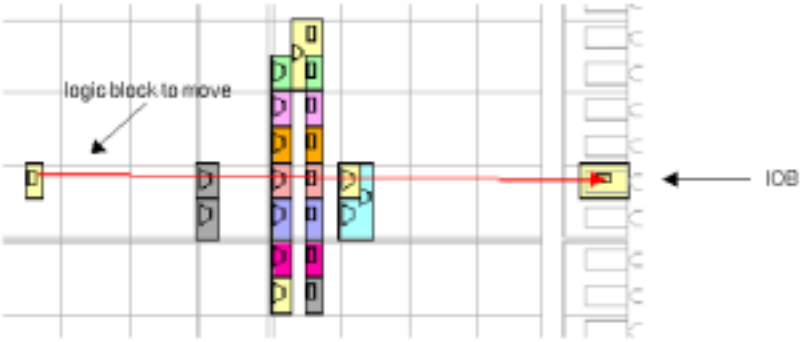


Figure 4-10 Ratsnest of the Failed Timing Constraint Path in the Floorplan Window

Now that you know where the routing delay is in the Floorplan window, you can manually move logic blocks to shorten the ratsnest and improve the routing delays to meet the Timing Constraint.

The following figure shows the example design. Compare this figure with the previous one and see that the logic block has been repositioned to be closer to the IOB.

Figure 4-11 Repositioned Logic in Floorplan Window

7. Save the Floorplanner file.
8. Rerun MAP and PAR with the new floorplan constraints to see whether the new placement meets the Timing Constraint in the design.

Finding Logic Connected to Nets

It is often difficult to find a specific piece of logic or the nets that connect them. The following procedure explains how to find logic that is connected to nets in the floorplanned design.

You choose the desired selection criteria such as name or connection. The Floorplanner places a black arrow to the left of the name of the found logic in the Design Hierarchy window. When the ratsnest is turned on, the Floorplanner draws black lines between logic in the Floorplan and Placement windows, indicating connectivity to the selected logic.

1. Select **Edit** → **Find** to open the Find dialog box.
2. Fill in the Search Criteria for the desired net. Set the Type field to Nets.
3. Click **Find**.
4. Click **SelectFound**.

   All nets that meet the search criteria are selected in the Design Nets window.
5. In the Find dialog box, fill in the Search Criteria. Set the Type field to the desired logic. In the Connections field, choose the connection type (either Loading Selected Nets or Driving Selected Nets) that you want from the list.

   For example, if you choose “Driving Selected Nets” the Floorplanner finds all the logic symbols that drive the input to the selected net as indicated by a black arrowhead to the left of the symbol name in the Design Nets window.
6. Click **Find**.
7. If any symbols are found, click **Select Found** to highlight them in the Design Hierarchy, Placement, and Floorplan windows. (The symbols are only highlighted in the Floorplan window if the logic has been floorplanned.)

Displaying Resources and Logic

You can display the logic resources that are available on the FPGA in the Floorplan and Placement windows. For example, the XC4000

family has these resources: F, G, and H function generators, Global Buffers, D-type flip-flops, Tristate Buffers, I/O, and RAM/ROM.

Note Screen refreshes take longer when you display all of the resource graphics in the Floorplan and Placement windows than when you just use the default settings.

By default the CLB resources are turned off to reduce the clutter in the Floorplan window. Only the Grid and I/O options are enabled.

1. Activate either the Floorplan or Placement window by clicking on the window title bar with the left mouse button.
2. Select **`View`** → **`Options`** or press the F2 key.

   This command opens the Options dialog box.
3. In the Resources panel, determine the type of resources that you want displayed in the selected (either Placement or Floorplan) window and click those boxes to enable the display.

   The following figure illustrates a partial display for an XC4000 device with all of the resources in the CLB enabled for display.

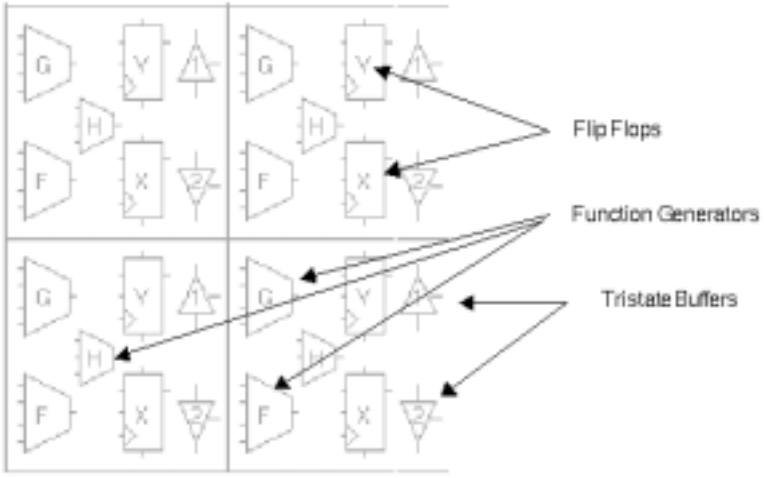


Figure 4-12 Display of Resources in the Floorplan Window

The following table shows how resources and placed logic affect which information the floorplan and placement windows display.

Table 4-1 Resource Graphics Display

Resources	Placed Logic	Display
Off	Off	Blank
On	Off	Shows available resources (not occupied by logic)
Off	On	Shows used resources
On	On	Shows both used and available resources

Performing Detailed Manual Placement

This procedure describes how to manually place logic into the Floorplan window, by selecting individual symbols from the design hierarchy and placing them in the Floorplan window.

Note You can only place an RPM as an entire group.

Perform the following steps to manually placed selected logic into the Floorplan window.

1. Expand a node in the Design Hierarchy window to show all of the symbols under that hierarchy. The following figure shows an example of an expanded hierarchical group.

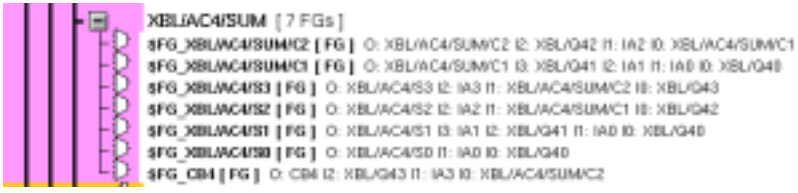


Figure 4-13 Expanded Hierarchical Group in the Design Hierarchy Window

2. Pick one symbol at a time and place it in the desired location in the Floorplan window. The following figure shows the cascaded Design Hierarchy and Floorplan windows.

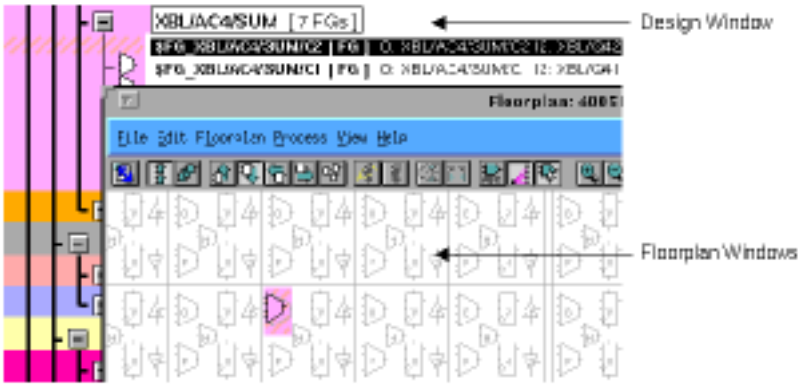


Figure 4-14 Placed Logic Symbol in the Floorplan Window

3. Repeat step 2 until you have placed all the symbols of the selected node in the Floorplan window.

Checking the Floorplan

The **`Floorplan`** → **`Check Floorplan`** command checks the floorplanned logic for placement problems. The Floorplanner opens the Check Floorplan Warnings dialog box and lists the warnings that describe any placement problems. If no problems are found, a message in the dialog box states that “0 Floorplan Errors Found.”.

1. You can find the logic that is associated with a warning by double–clicking on that warning in the dialog box.

   The Floorplanner zooms in to the problem logic, which appears highlighted.

2. You can also find the problem logic by single-clicking and clicking **`Find`** on the dialog box.

   If you have flashing enabled, the problem logic flashes in the Floorplan window.

Aligning Symbols

This procedure explains how to align symbols. When you are placing logic in the Floorplan window, aligning symbols reduces unnecessary routing between the placed logic elements. You can use the ratsnest to see the alignment of the placed logic.

The following considerations are important when floorplanning the design.

- Align tristate buffer enable signals. By placing tristate buffers with common enable signals in the same column, PAR will utilize a single longline to connect all the enables to the source.
- Align clock enable signals. By placing flip-flops with common clock enables in the same column, PAR will utilize a single longline to connect all the clock enables to the source.
- If an enable is sourced by an I/O, place the IOBs close to the column in which the longline runs to minimize the routing required for connections to that longline.
- Use the **Hierarchy** → **Group By** command to create groups of related logic for quicker placement into the floorplan. The following figure shows the Group By dialog box.

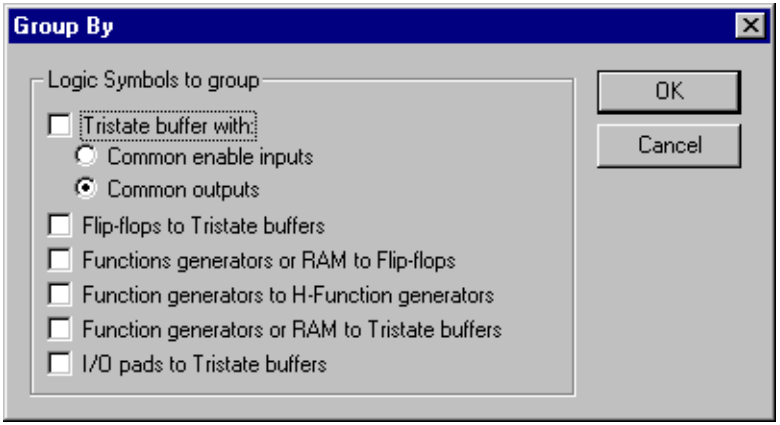


Figure 4-15 Group By Dialog Box

Perform the following steps to align logic symbols in the Floorplan window.

1. Prior to floorplanning, use the **Hierarchy** → **Group By** command to make groups of flip-flops to tristate buffers.
2. Floorplan the groups of flip-flops to tristate buffers (if small enough) into the same quadrant in the Floorplan window.

   For additional details, refer to the "Analyzing PAR Placement" section.

3. Floorplan the IOBs such that they are aligned with the flip-flop to tristate buffer groups. Check the ratsnest in the Floorplan window to see the alignment of the placed logic.

The example design shown in the following figure contains 16 flip-flops, 12 IOBs, 16 BUFTs, and one BUFGP.

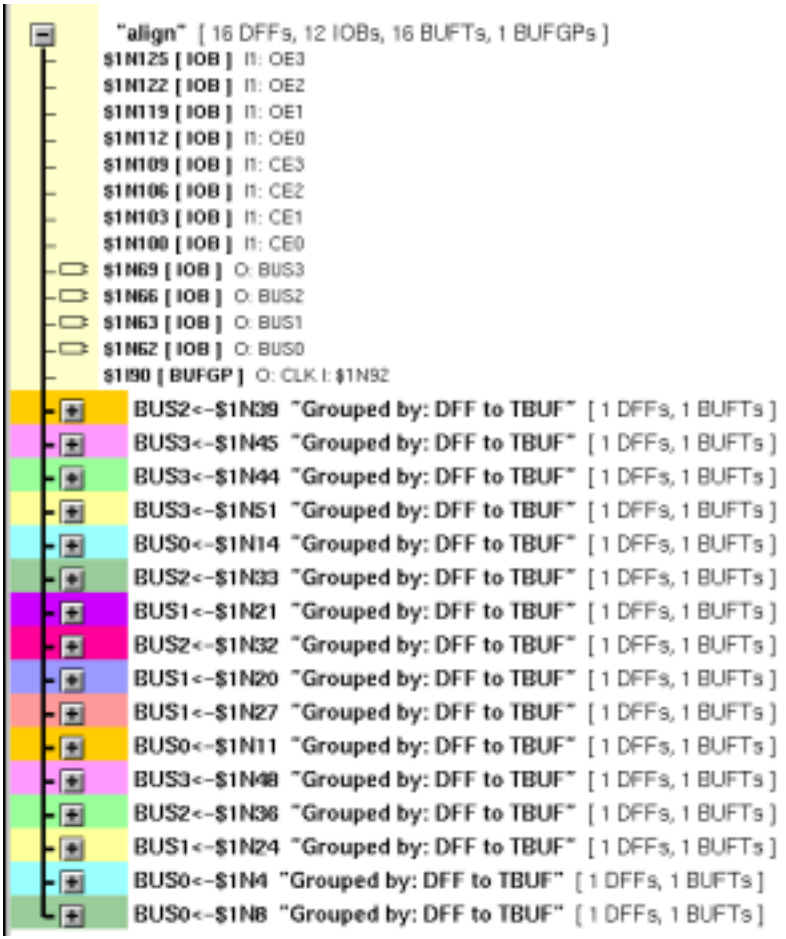


Figure 4-16 Example Design

The following figure shows the example design, floorplanned so that the logic symbols are aligned.

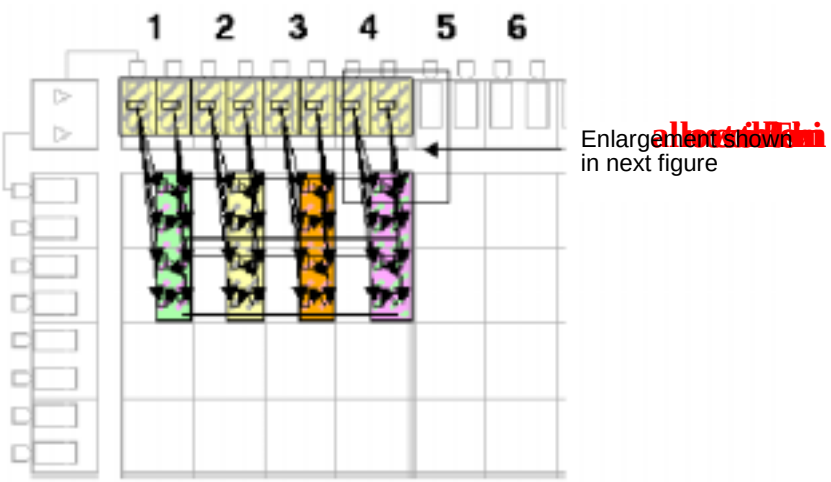


Figure 4-17 Properly Aligned Logic Symbols

The following figure shows a closer view of a part of the floorplanned design shown in the previous figure. Note the alignment of the nets sourced by IOBs $1N100 and $1N112. BUS3 is aligned to a horizontal longline.

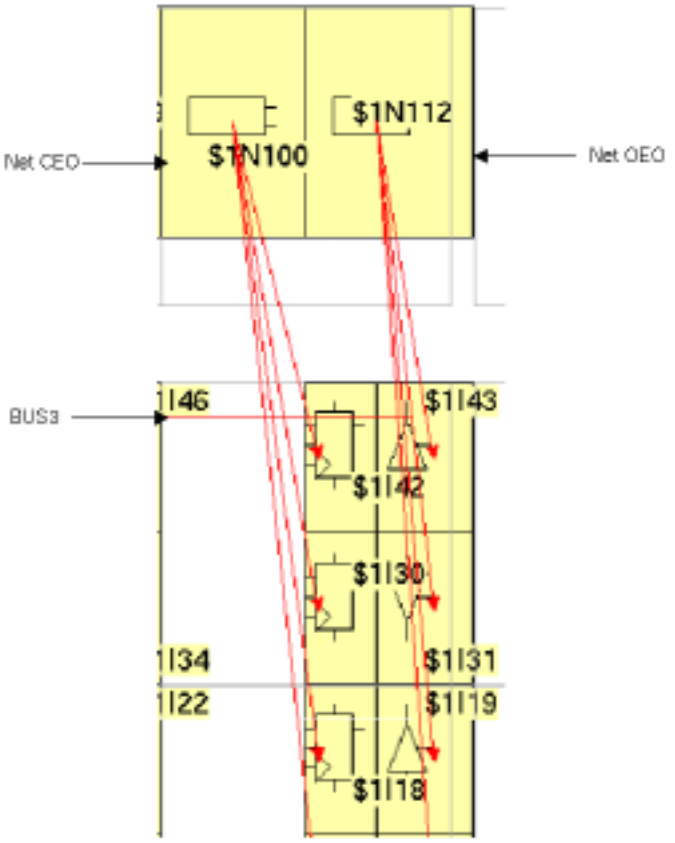


Figure 4-18 Close-up of Aligned Logic Symbols

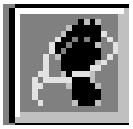
Working with Patterns

This procedure explains how to work with patterns when you place selected logic in the Floorplan window. You can use the **`Floorplan → Capture Pattern`** and **`Floorplan → Impose Pattern`** commands or use the **`Capture Pattern`** and **`Impose Pattern`** toolbar buttons. These commands allow you to create a reference pattern of placed logic that you can use later when you are placing similar logic. The reference pattern represents the relative placement and spacing of individual pieces of floorplanned logic.

Note Except for RPMs, you must have logic placed in the Floorplan window before you can use the **Capture Pattern** command. RPMs have a pattern. You can pick up an RPM and capture its pattern before dropping it into the Floorplan window.

Creating a Pattern

You can select the **Floorplan** → **Capture Pattern** command or use the **Capture Pattern** toolbar button shown in the following figure.

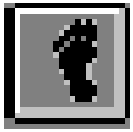


Follow these steps to create a pattern that captures the placement and spacing information of placed logic in the Floorplan window.

1. Select some logic from the Design Hierarchy window and place it in the desired locations in the Floorplan window.
2. Select the logic in the Floorplan window, which makes the pattern that you want to capture for future use.
3. Select **Floorplan** → **Capture Pattern** or click the **Capture Pattern** toolbar button.

Using a Pattern

Imposing a pattern can only be done after you have captured a pattern. You can select the **Floorplan** → **Impose Pattern** command or use the **Impose Pattern** toolbar button shown in the following figure.



Follow these steps to use a reference pattern to place selected logic from the design hierarchy into the Floorplan window.

Note Use the **Impose Pattern** command on logic that is similar to the placed logic whose placement was captured as a pattern.

1. Select the desired logic from the design hierarchy.
2. Select **Floorplan** → **Impose Pattern** or click the **Impose Pattern** toolbar button.
3. Move the selected logic into the Floorplan window.

   The Floorplanner reads the placement and spacing information from the reference pattern to place the selected logic.

   Note You can use the **Impose Pattern** command on transitory logic or selected logic that is already placed in the Floorplan window.

How to Interleave Buses

This procedure explains how to interleave buses. The **Floorplan** → **Distribute Options** command makes interleaving easy. Interleaving spreads out the resources associated with a bus, such that other logic can be interspersed with the bus. A goal of interleaving is to minimize the distance between similar bits of interrelated buses.

Perform the following steps each time you begin this procedure.

1. Determine the spacing requirements that suits your particular interleave scheme.
2. Select **Floorplan** → **Distribute Options** and set the Interleave Factor to change the spacing value.

   Note An interleave factor of 2 causes the Floorplanner to place selected logic in every second available resource in the floorplan; a value of 3 causes the Floorplanner to place selected logic in every third available resource.

Design Example

The following figure shows the example design. It contains a 16-bit bus (REGISTER_A / Q) that talks to an 8-bit bus (LITTLE_BUS_OUT). The object is to interleave the high-order bits of REGISTER_A / Q with the low-order bits, and do the same for the input IOBs (DIN).

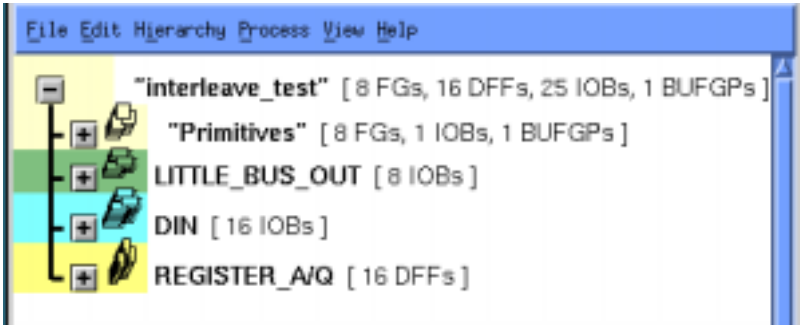


Figure 4-19 Interleave Design Hierarchy

The following figure shows the expanded hierarchy of the interleave design.

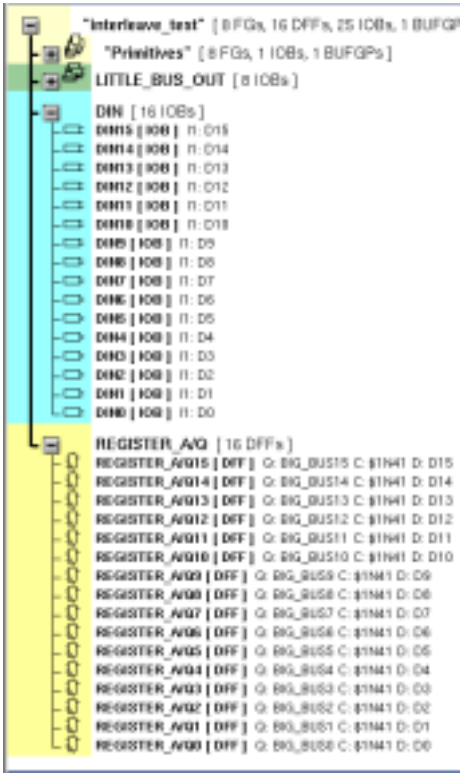


Figure 4-20 Interleave Design Hierarchy Expanded

The following procedure shows how to interleave the REGISTER_A / Q and DIN bits.

1. Use Group from the Hierarchy menu to create four new groups from the expanded hierarchy. This will make floorplanning easier.

   - REGISTER_A_LSB
   - REGISTER_A_MSB
   - DIN_LSB
   - DIN_MSB

   The following figure shows the interleave design hierarchy with the four new groups. Note the "`Grouped by: User`" annotation on the symbol line.

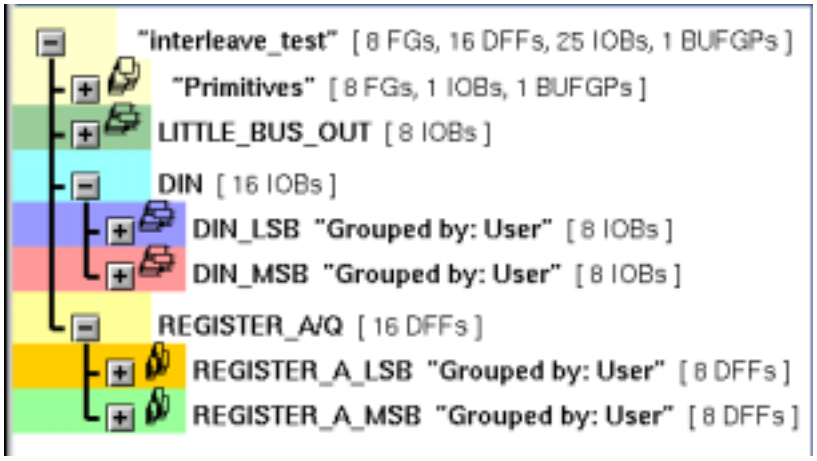


Figure 4-21 Interleave Design with New Groups

Note The placement used throughout this example is exploded to make it easier to view; it is not optimal floorplanning placement. During floorplanning you would choose a placement that is closer together, resulting in shorter interconnections.

The next six figures show step-by-step the sequence for placing the groups of logic into the Floorplan window. The seventh figure shows the final placement in the Floorplan window of the interleave design. The ratsnest display shows the relative distance and connectivity of the nets.

2. Place the REGISTER_A_MSB group in the design.

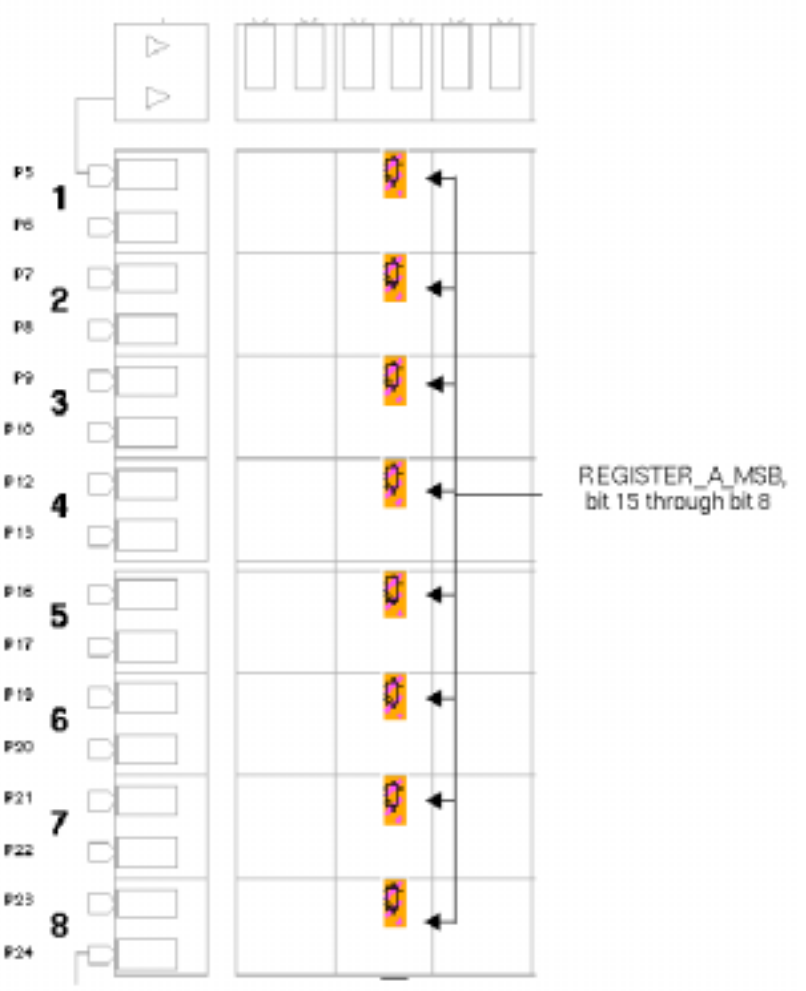


Figure 4-22 Placement of the REGISTER_A_MSB Group

3. Interleave the REGISTER_A_LSB group with the REGISTER_A_MSB group.

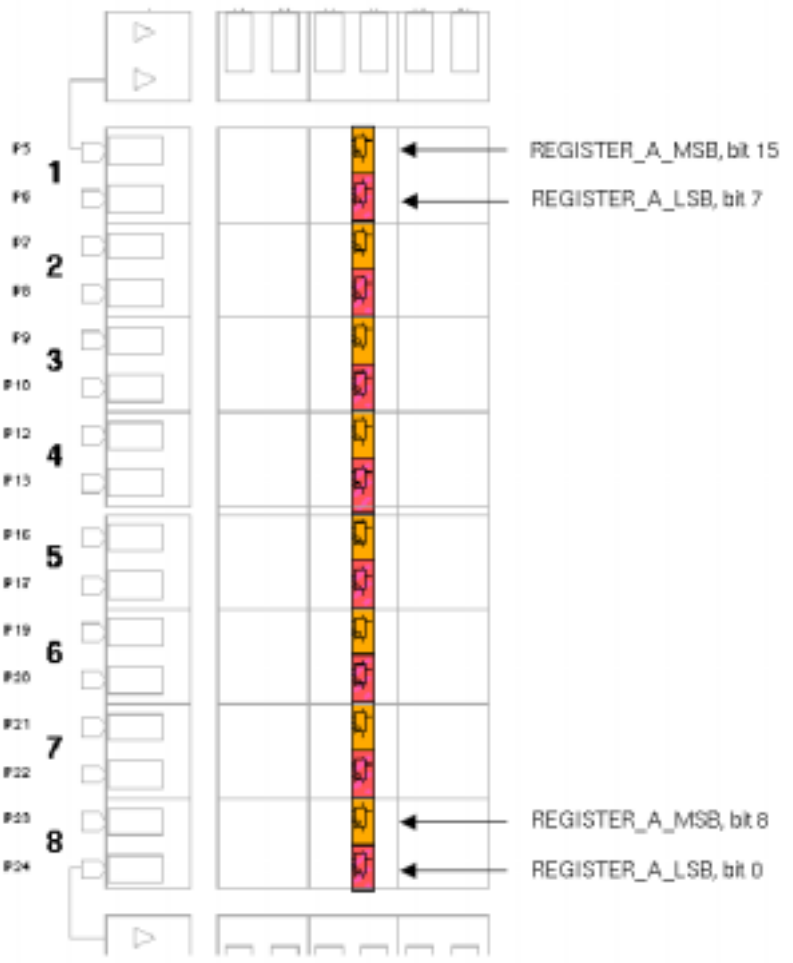


Figure 4-23 REGISTER_A_LSB Group Interleaved with REGISTER_A_MSB

4. Place the DIN_MSB group to align with the REGISTER_A_MSB group.

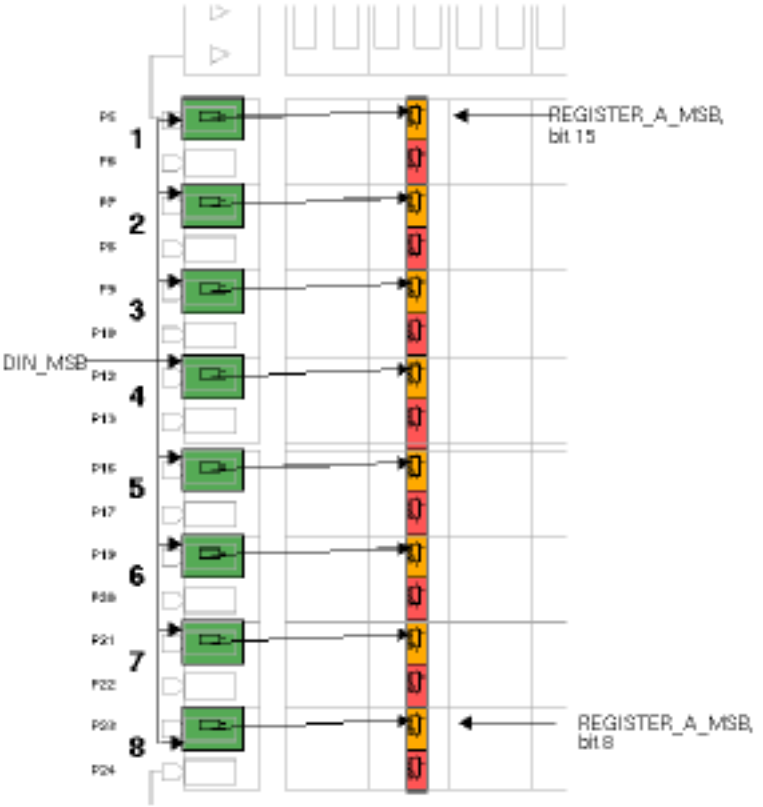


Figure 4-24 DIN_MSB Placed to Align with REGISTER_A_MSB

5. Interleave the DIN_LSB group with the DIN_MSB group. This aligns DIN_LSB with REGISTER_A_LSB.

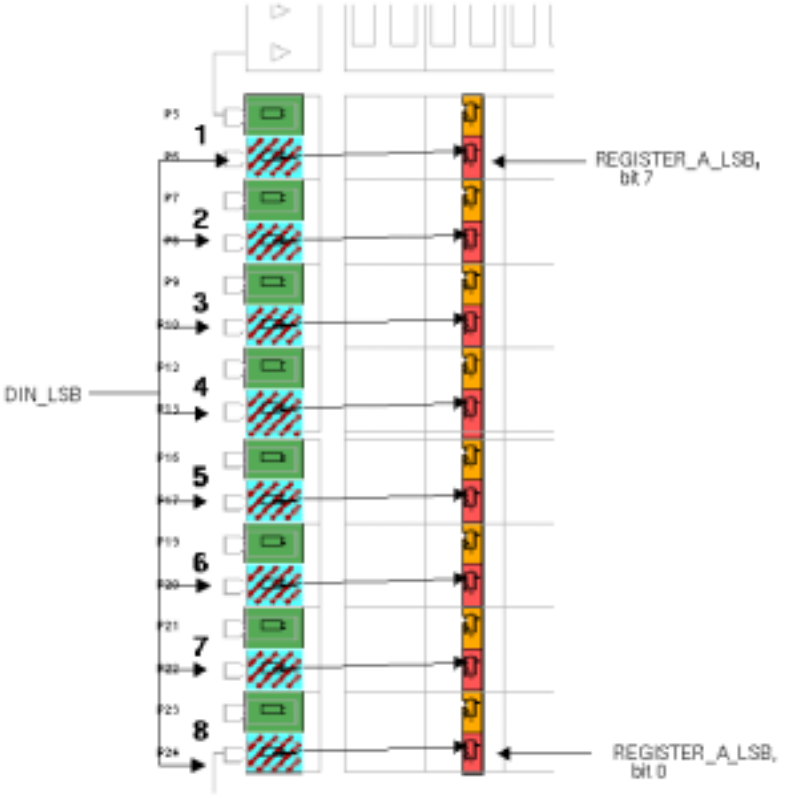


Figure 4-25 DIN_LSB Placed to Align with REGISTER_A_LSB

6. Place 2-in-1 multiplexers to align with the A / Q bus.

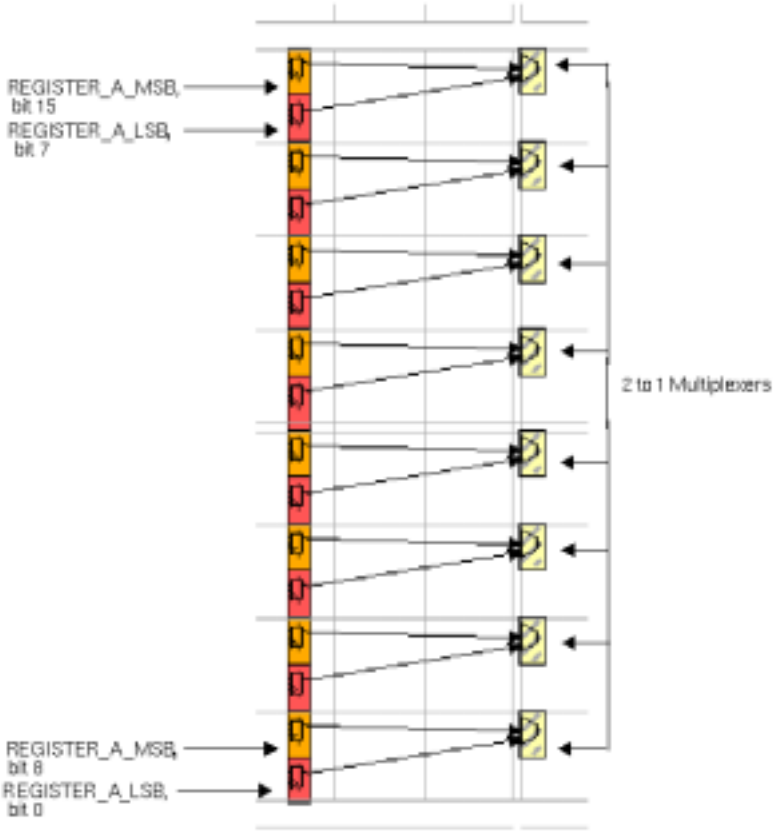


Figure 4-26 2-in-1 Multiplexers Placed to Align with A/Q Bus

7. Place the LITTLE_BUS_OUT group to align with the multiplexers.

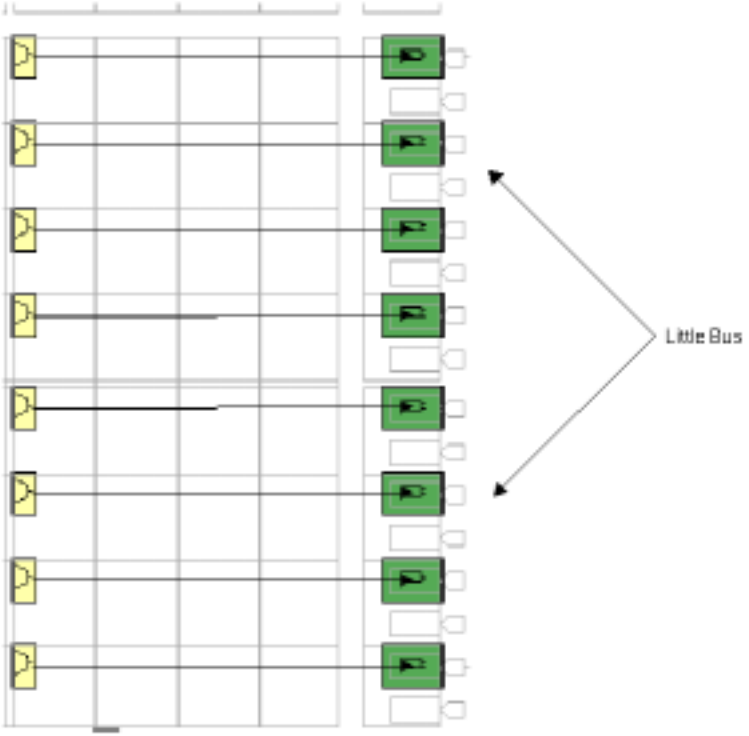


Figure 4-27 Little Bus Placed to Align with Multiplexers

The following figure shows the final design.

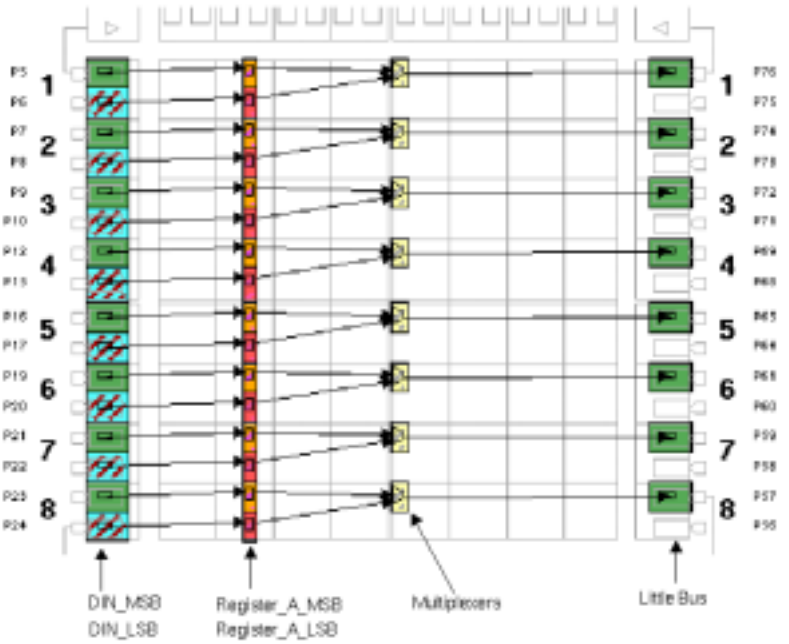


Figure 4-28 Final Placement of Interleave Design

Iterative Floorplanning

This procedure explains how to floorplan your design iteratively. In this method, you manually place the structured logic of the design. Next, run MAP and PAR to automatically map and place the random logic of the design. Repeat this cycle of manual and automatic map and placement until you have placed the critical paths of the design in the FPGA floorplan.

Perform the following steps to floorplan iteratively.

1. Work with the most critical path first.
2. From this part of your design, floorplan the structured logic.
3. Select **`File`** → **`Save`** to save the floorplan and generate an MFP file.
4. Run MAP and PAR, using the new MFP file, to obtain a placed design.

When PAR is done, select **File** → **Update** to read the placed design file in the Floorplanner. You can either make changes to the original floorplan and repeat these steps, or move on to floorplanning more logic.

Floorplanning Incremental Schematic Changes

Xilinx defines incremental designing as making changes, at the design entry stage, to a design that has been previously implemented in an FPGA, with or without floorplanning. These changes can include the following.

- Adding logic
- Removing logic
- Changing existing logic

The following procedure explains only how to make incremental changes when you have used the Floorplanner to floorplan the original design.

Note This procedure may not work as well with HDL designs. Using the HDL coding tips in the *Synopsys Synthesis and Simulation Design Guide* will make incremental floorplanning on HDL designs easier.

Design Example

The schematic-based design has been floorplanned and a change in the design requires that you add some logic to the original schematic.

1. Make the necessary changes to the schematic design.

   Note Whenever you make changes to the schematic, you must run NGDBuild and MAP to regenerate a new mapped or placed NCD file.

2. Select **File** → **Update** to open a file open dialog box.
3. Select the new NGD and NCD files to read in to the Floorplanner.

   The Floorplanner reads in the *design*.ncd. Whenever an FNF file exists for the design, Floorplanner reads it in addition to reading the NGD and NCD files. The logic that was floorplanned is again placed into the same location in the Floorplan window.

The next three figures show the design example schematic, the design hierarchy of the NGD and NCD files, and the floorplanned design.

The following figure shows the original schematic of the design example.

Figure 4-29 Design Example

The following figure shows the design hierarchy of the example in the Design Hierarchy window.

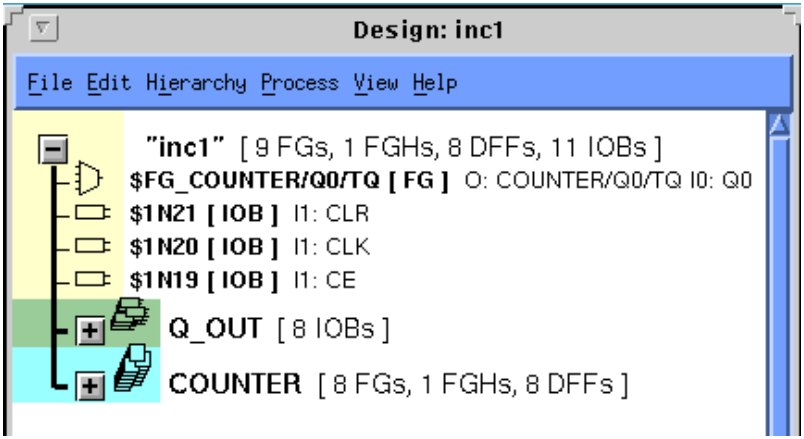


Figure 4-30 Design Hierarchy of Design Example

The following figure shows the floorplanned design example.

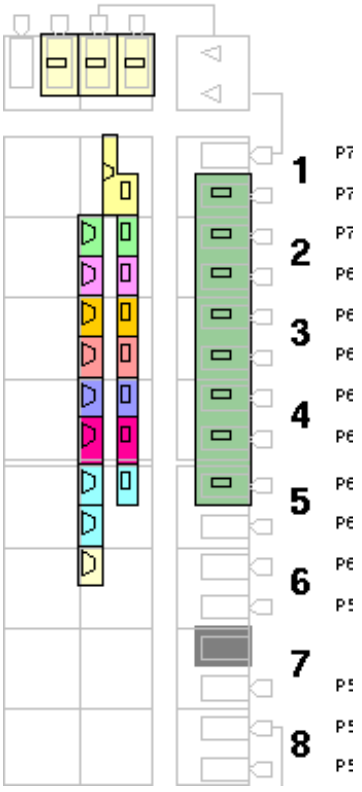


Figure 4-31 Floorplanned Design Example

The following figure shows the design example schematic that has been modified. A two-input AND gate, an OBUF, and an OPAD have been added to the design.

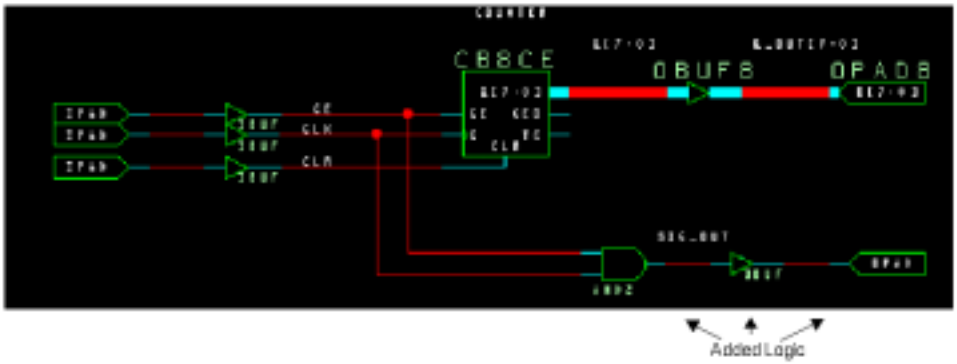


Figure 4-32 Changes to the Original Schematic

When the design is reloaded into the Floorplanner, the newly added logic appears in the Design hierarchy. In the following figure, two new symbols $1N57 [IOB] and $FG_SIG_OUT [FG] still remain in the hierarchy (as indicated by the logic symbol icon) and may be placed into the floorplan.

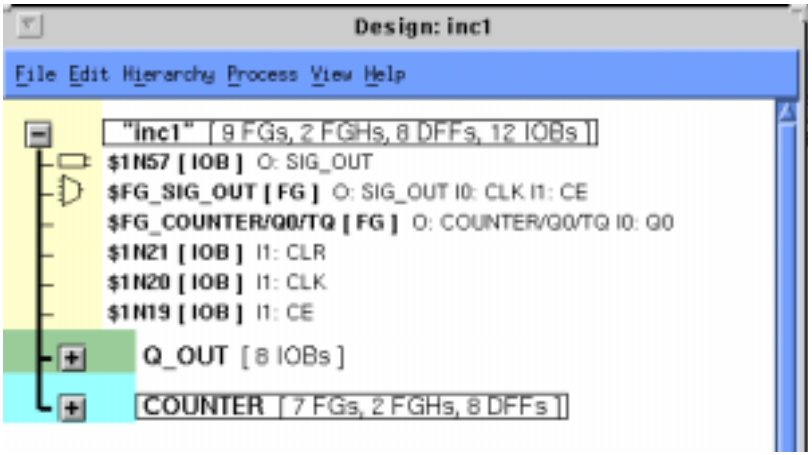


Figure 4-33 Design Hierarchy of New NCD File

Making Small Modifications to Automatic Placement

This procedure describes how to make small changes to an automatically placed design in order to fix packing or placement problems. A scenario when this might be required is when the automatic place-

ment tools meet timing on all paths except one, and the one path that is failing is due to bad logic placement in the path. Using the Floorplanner you can copy the entire automatic placement to the Floorplan window, and then make the small modifications necessary to fix the timing problems. You can then re-map and place and route the design.

1. Use the Placement window and timing reports to identify problems in the automatic placement.
2. Select **Floorplan** → **Replace All with Placement** to copy the entire placed design from the placed design over to the Floorplan window.
3. Make the changes to the Floorplan window to correct any placement that was causing timing problems in the initial timing analysis.
4. Select **File** → **Save** to save the floorplan and generate an MFP file.
5. Map the design using the newly created MFP file.
6. Place and route the design to determine if the new placement constraints allows the design to meet the required timing.

Lock Down I/Os from Automatically Placed Design

This procedure describes how to select the I/Os from an automatically placed and routed design and lock them down in the Floorplan window. This procedure could be extended to show how other internal logic components can be locked down in the Floorplan window from the Placement Window.

1. Select **Window** → **Tile Compare** to display the Placement and Floorplan windows next to each other.
2. Select **Edit** → **Find** to display the Find dialog window.
3. In the Find dialog window, set the Types drop-down to “I/O Pads”.
4. Click **Find**.
5. Click **Select All** to select all the I/Os in the design.

6. Select **Floorplan** → **Constrain from Placement** to copy all the I/O locations from the Placement window to the Floorplan window.
7. You can now make modifications to the automatic I/O placement by moving the I/Os in the Floorplan window.

Getting Started With an Unfamiliar Design

This procedure explains how to use the Floorplanner to familiarize yourself with the connectivity of someone else's design.

There may be occasions when you must floorplan a design with which you are unfamiliar. There are some basic steps you can take to make the floorplanning tasks more efficient.

- View the schematic or HDL to understand the structure of the design and data paths.
- Determine the design's structure, such as RAM and buses.
- Try to determine the performance requirements of the design.

When you load a design, the Floorplanner uses net and symbol names to create a design hierarchy. Some designs may contain little information about the hierarchy. For such designs, you could create a fake hierarchy by following these steps.

1. Select **Hierarchy** → **Group By** to open the dialog box.
2. Group columns of tristate buffers, and simple function generator, flip-flop, and tristate buffer relationships.
3. Click **OK** when you are done.

Creating Hierarchy at a Higher Level

To create groups of logic at a higher level, follow these steps.

1. Select **Edit** → **Find** to open the Find dialog box.
2. Use the dialog box to find and select all of the IOBs in the design.
3. Create a group of these IOBs using the F3 key, or the **Hierarchy** → **Group** command.

   The grouped IOBs appear in a stack in the Design Hierarchy window.

4. Expand that stack, then scroll through the list, looking at the pin types on the symbols.

Creating Subgroups

1. Make subgroups of those IOBs with pins named "I" (input IOBs), "O" (output IOBs), and those IOBs with both "I" and "O" pins (bidirectional IOBs).
2. Select the group of input IOBs.
3. Identify the symbols in the group that directly load the new group using either the **`Edit` → `Select Loads`** command, or open the **`Edit` → `Find`** dialog box and choosing the Loading Selected Logic connection.
4. Make a group of those symbols.
5. Repeat steps 3 and 4 to create additional groups of related symbols. Stop when the groups become too large, too small, or too complex to be of value.

   Note Each time you make a subgroup, manually de-select the nodes that have been previously grouped.
6. Select the group of Output IOBs.
7. Perform the same sequence (steps 2 through 5) as before, but change the connection type or search criteria to Sourcing Selecting Logic.
8. Each time that symbols are selected, scroll through the lists and adjust the selections as appropriate.

   The goal is to break large groups into more manageable sizes, without creating a large number of small groups.
9. Repeat the process of making subgroups (steps 7 and 8) with the group of bidirectional IOBs, for each direction.
10. If there are other known structures in the design such as RPMs or tristate buffer columns, use them as a starting point for making other subgroups.

Floorplanning the New Hierarchy

After you have manageable groups of logic for the design, you can begin floorplanning.

1. Use the Placement window and timing reports to identify problems in the automatic placement.
2. Floorplan the IOBs.

   Use the **Edit → Select Loads** and **Edit → Select Sources** commands, as well as the **Edit → Find** dialog box to guide you in a systematic manner.
3. Use the ratsnest display in the Floorplan window to identify which symbols need rearranging to reduce interconnect congestion.

Glossary

BEL

A basic element, for example, an individual flip-flop or LUT.

block

A group consisting of one or more logic functions.

BUFT

Tristate buffer.

CLB

Configurable Logic Block.

critical path

A signal in a section of combinatorial logic that limits the speed of the logic. Synchronous elements begin and end a critical path, which may include an I/O pad.

design hierarchy

A graphical representation of the MAP file in the Floorplanner Design Hierarchy window.

DFF

D-Latch or flip-flop.

guide file

An NCD file representing a previously placed and routed design, which is used in a subsequent place and route operation.

HDL

Hardware Description Language. The most common HDLs in use today are Verilog and VHDL. They describe designs in a technology-independent manner using a high level of abstraction.

IOB (input/output block)

An IOB is a collection or grouping of basic elements (BELs) that implement the input and output functions of an FPGA device.

I/O blocks

The input / output logic of the device containing pin drivers, registers, and latches, and tristate control functions.

I/O pads

The input / output pads interface the design logic with the pins of the device.

logic icon

Graphical representation of a logic resource, such as a flip-flop, buffer, or register.

logic icons in transit

Selected logic that is being moved from one location to another in the Floorplanner.

longlines

Each CLB column has four dedicated global vertical longlines, and each of these lines connects to a primary global net or to any secondary global net. These lines are very fast.

map

The process of assigning a design's logic elements to the specific FPGA physical elements that actually implement logic functions in a target device.

menu bar

The area located at the top of the main window that provides access to

the menus.

net

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.

optimize

The process of transforming a design to decrease its area or to increase its speed performance.

pad

The physical bonding pad on an integrated circuit. All signals on a chip must enter and leave by way of a pad. Pads are connected to package pins in order for signals to enter or leave an integrated circuit package.

place

The process of assigning logic from your design to physical cell locations in the FPGA.

ratsnest

Lines that indicate connectivity between logic in the Floorplanner or Placement window.

resource graphics

Graphical representation of elements in the target FPGA Floorplan window, such as function generators, registers, tristate buffers in the CLB, and IOBs.

route

The process of assigning logical nets to physical wire segments in the FPGA that interconnect logic cells.

router

The utility that connects all appropriate pins to create the design's nets.

schematic

An electronic drawing representing a design in terms of primitive elements.

selecting logic

In the Floorplanner, the process of using the mouse to choose logic in either the Design Hierarchy window or the Floorplan window for placement, movement, or processing.

status bar

An area located at the bottom of a window that provides information about the commands that you are about to select or that are being processed.

tristate buffer

A logic primitive with three possible output states.

toolbar

A field located under the menu bar at the top of your window. It contains a series of icons (buttons) that you click on to execute some of the most commonly used commands. These buttons are an alternative to the menu commands.

UCF

The User Constraints File (UCF) is an ASCII file specifying constraints on the logical design. These constraints affect how the logical design is implemented in the target device. The file can also be used to override constraints.