ASICmaster™ User's Guide

PRE-PRODUCTION
For the latest version of this document, check the User area on Actel's website at:

http://www.actel.com/user

Windows® NT™ and UNIX® Environments

Actel Corporation, Sunnyvale, CA 94086

Printed in the United States of America

Part Number: 5579018-0

Release: June 1999

Trademarks

Actel and the Actel logotype are registered trademarks of Actel Corporation.

ProASIC and ASICmaster are trademarks of Gatefield Corporation.

Adobe and Acrobat Reader are registered trademarks from Adobe Systems, Inc.

Verilog is a registered trademark of Open Verilog International.

UNIX is a registered trademark of X/Open Company Limited.

Windows is a registered trademark and Windows NT is a trademark of Microsoft Corporation in the U.S. and other countries.

All other products or brand names mentioned are trademarks or registered trademarks of their respective holders.

Table of Contents

List of Figures

List of Tables

Introduction

ASICmaster is a layout system that reads structural netlists and produces optimal design layouts for Actel's ProASIC devices using timing driven place and route.

The *ASICmaster User's Guide* contains information and procedures for using ASICmaster to place and route designs for Actel's ProASIC family of devices. This includes an overview of ASICmaster and the design flow and information about using ASICmaster for layout, delay calculation, and program file generation.

Also is included information about viewing the results of place and route, and information about generating place and route constraint files and post layout back annotation files for timing simulation. Additionally, information about using ASICmaster in command line and batch is included.

Document Organization

The ASICmaster User's Guide is divided in to the following chapters:

Chapter 1 - **ASICmaster Overview** describes the design flow and outlines the tasks that can be performed using ASICmaster.

Chapter 2 - **Using ASICmaster** explains how to use ASICmaster to select a design and start a design project work session. It also describes how to use basic Design Executive tools, set project preferences, place and route a design, and generate files, such as the bitstream device programming file.

Chapter 3 - **Viewing Results** describes how to view ASICmaster results, including layout views, output data, and log files.

Chapter 4 - **Using Constraints** describes how to create place and route constraint files for ASICmaster.

Chapter 5 - Timing Analysis explains how to calculate post-layout delays and perform back annotation.

Chapter 6 - Using the ASICmaster in TTY and Batch Modes describes how to use ASICmaster in interactive command-line mode and batch mode.

Chapter 7 - **Troubleshooting** presents useful hints and advice for successful operation.

Appendix B- File and Command Selection describes user interface features not covered in other chapters.

Appendix B - **Supported Netlist Formats and Constructs** describes netlist formats and the constructs supported by ASICmaster.

Appendix C - **The Design (*.dtf) Directory** explains the contents of the .dtf design directory. Each .dtf directory contains an ASICmaster database for a design.

Appendix D - **Batch Script Syntax** describes the syntax for batch scripts that run ASICmaster.

Appendix E - **The .amrc Startup File** describes the .amrc file used to configure ASICmaster.

Appendix F - Product Support provides information about contacting Actel for customer and technical support.

Document Assumptions

The information in this guide is based on the following assumptions:

1. You have installed the ASICmaster software.
2. You are familiar with the principles of digital design.
3. You are familiar with UNIX or Microsoft Windows NT operating systems.

Documentation Conventions

Unless otherwise noted in the text, this guide follows these documentation conventions:

Information that is meant to be input by the user is formatted as follows:

- *Italic* type is used in text, syntax, and examples to identify placeholders that need to replaced with specific words or values. It is also used selectively for emphasis.
- <> (angle brackets) identify a key or key combination that to press, such as <g>, <Return> or <Ctrl+g>. (Ctrl refers to the Control key.)
- ASICmaster graphical user interfaces and the documentation conventions used in the Actel documentation represent <Ctrl>-*key* and <Alt>-*key* sequences
  in slightly different ways.
- For example:
- To indicate holding down the <Ctrl> key and pressing the <g> key, the *interfaces* use <Ctrl+G>.
- To indicate holding down the <Ctrl> key and pressing the <g> key, the *manual* uses <Ctrl+g>.
- To indicate holding down the <Ctrl> and <Shift> keys while pressing the <g> key, the *interfaces* use <Ctrl+Shift+G>.
- To indicate holding down the <Ctrl> and <Shift> keys while pressing the <g> key, the *manual* uses <Ctrl+Shift+G>.

  [ ] (square brackets) enclose lists of optional items.

  { } (braces) enclose lists of required items.

  | (vertical bar) separates:

ProASIC Series Manuals

This book is a part of the ProASIC series of manuals. All books in the series are listed below. Users can order these publications through an Actel sales representative.

ASICmaster Installation and Licensing Guide provides information and procedures for installing and licensing the ASICmaster software.

ASICmaster User's Guide provides information about the design flow for creating designs for ProASIC device. It includes information and procedures for placing and routing designs and also information on using timing constraints.

MEMORYmaster User's Guide provides information and procedures for generating embedded and distributed memories and instantiating then into a design.

ProASIC Macro Library Guide provides descriptions of ProASIC library elements for Actel's ProASIC device families. Symbols, truth tables, and timing parameters are included for all macros.

ProASIC Interface Guide provides information and procedures for designing Actel's ProASIC devices in Exemplar synthesis, Synopsys synthesis, Verilog simulation, and VHDL simulation environments.

1

ASICmaster Overview

This chapter describes the design flow for creating designs for Actel ProASIC devices using ASICmaster and third-party tools. It also describes the tasks, task cycles, and iterations to perform in ASICmaster.

Design Flow Illustrated

Figure 1-1 shows the design flow for an Actel ProASIC device using ASICmaster and third-party software.

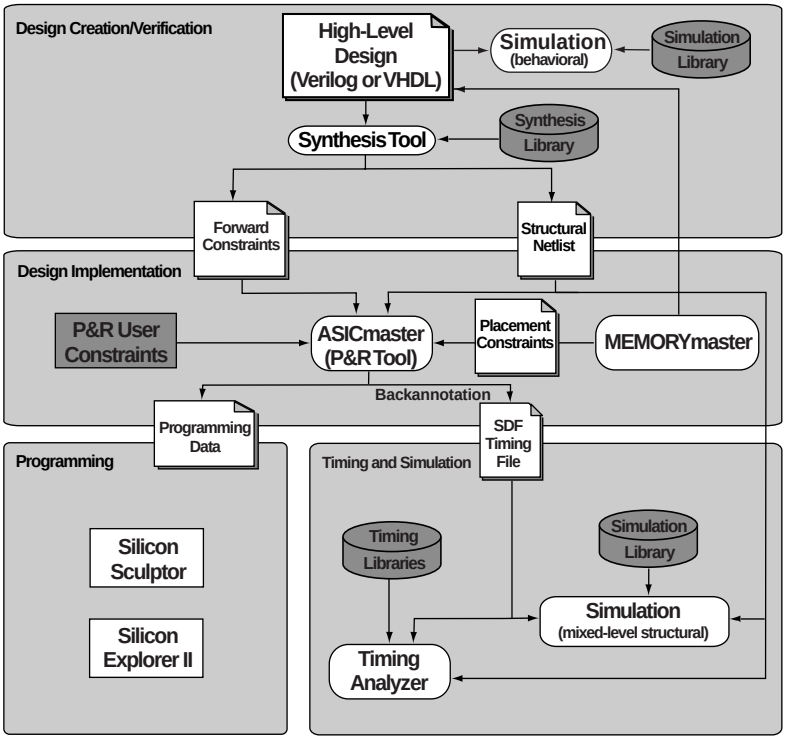


Figure 1-1. ASICmaster Design Flow

ASICmaster Task Flows and Iterations

This section describes the steps that are typically performed with ASICmaster after a design has been created for a ProASIC device, synthesized, and a netlist generated.

1. Start a project, set project preferences and select a device.
2. Import and check the netlist and constraint files.
3. Place and route the design. ASICmaster is also used to iterate the device-selection and/or netlist-import cycles (and, if necessary, the design analysis, design synthesis, and/or design definition processes).
4. Generate post-layout delays and timing back annotation files for simulation/timing analysis.
5. Use third-party tools to perform simulation and timing analysis using the post-layout delays. As needed, iterate the design analysis, device-selection, layout, and/or file-generation cycles (and, if necessary, the design synthesis and/or design definition processes).
6. Generate device programming bitstream file.
7. Program the device using one of the supported device programmers.
8. Generate the design datasheet, boundary scan file and pin map.

Starting a Project

After launching ASICmaster, the first task is setting the layout and device programming project preferences for the design to create a design database. ASICmaster uses the project preferences to add parameters to the database, control task operations, and alert users to potential and actual problems. It is important that ASICmaster has as much information as possible about the environment in which the design is being created. Project preferences to set include specifying the netlist that describes the design, the ProASIC device for the design implementation, the timing and back annotation file, and the pin-out file.

Additionally, specifying anticipated operating conditions, including temperature, voltage, and process allows ASICmaster to use the operating conditions as additional timing parameters to calculate post-layout delays

After successful timing verification and post-layout, the datasheet and the bitstream programming file are added to the JTAG file as additional project preferences.

Importing a Netlist

The netlist is the design information source for the project. Netlists can be in EDIF 2.0.0, VHDL, or Verilog format. Only one format can be used for a design. When a netlist is imported, ASICmaster checks the netlist format and design rules, imports the netlist and constraint files specified during project set up, and incorporates the data into the design project database. ASICmaster also checks the constraint files for the following:

- Verifies whether the design and device are compatible in terms of gate and I/O counts.
- Develops optimal parameters for the use of global resources and assigns them to high fanout nets when appropriate.
- Checks for consistency between the design and the project setup and physical design parameters and constraints.
- Optimizes the design by removing redundant, non-functioning, and non-required logic. critical path nets are shielded by inserting buffers to disconnect the path from high fanout loading.

Error Detection and Reporting

If errors are detected in a netlist, device, or constraint file, a message is displayed and the task run is terminated. Common reasons for failure are:

- One (or more) netlist file is in an unsupported format.
- The netlist files are not all in the same format.
- One (or more) netlist file fails a design rule or electrical check.
- The device and the netlists are incompatible in terms of gate count or I/O resources.
- Constraints are incompatible with the design description, device characteristics, or both.

The error report can be viewed to determine the cause of the error. Refer to “Viewing Results” on page 39 for additional information. Reports can also be viewed to get more information about the operation as a whole, including information on system warning messages.

Place and Route

The main function of ASICmaster is to place and route a design to yield the desired circuit performance. After the netlist and constraint files are

successfully imported, all the information necessary to guide a favorable place and route task has been collected. The following sections summarize the layout-related operations and indicate task iterations that might be performed to optimize task results.

Mapping Package I/O Pins to I/O Signals

When ASICmaster imports the netlist and device data, it automatically checks I/O requirements. The placement program automatically places each I/O signal in the optimum position for successful operation.

Manual assignment can also be performed using constraint files. Refer to "Using Constraints" on page 63 for additional information.

Note: A single output signal can be allocated to multiple output buffers to increase drive strength.

Avoiding Excessive Constraints

The usual reason for routing failure is that the design has been over constrained through too stiff timing budgets or fixed placements. Avoid over constraining designs, especially for the first few runs of a design. Constraints limit ASICmaster's ability to create optimal physical layouts. Specific constraints can be added later to supplement the design parameters that ASICmaster automatically generates and to guide place and route.

Generating and Back Annotating Delays

After achieving a satisfactory layout, post-layout delays can be generated and a back annotation file can be created if post-layout delay calculation is enabled. The post-layout calculation extracts pin-to-pin, interconnect net information, and RC data and performs delay analysis on each net and generates a back annotation file in SDF format.

This information is used to verify a design during simulation/timing analysis. If the simulation/analysis results indicate problems, the design should be put through synthesis and design definition cycles before repeating the layout cycle and any required ASICmaster tasks.

Design Optimization

ASICmaster report files and iterating tasks should be used to optimize the design. When iterating only to optimize the design after a previously successful layout, all subsequent processing usually succeeds. If the design

has only minor changes users can recycle the previous placement in an ECO mode and run placement refinement.

Optimizing Inverters

If ASICmaster is run with Netlist Optimization enabled, (default mode) inverters that do not limit fanout are removed from the design netlist.

Timing Optimization

Timing optimization can be performed by iterating the design analysis and layout cycle on a design until it meets timing and layout requirements. Refer to “Back Annotating Timing Information” on page 83 for additional information.

Generating Files

Once the design meet timing and layout requirements, the device programming bitstream, datasheet, JTAG and pin map files can be generated. Refer to “Design Executive Window” on page 14 for additional information.

Design Integrity

ASICmaster offers a combination of safeguards that protect the integrity of a design, and flexibility to override the safeguards when necessary. ASICmaster keeps track of a design project and ensures that follow the design flow is followed to maintain the integrity of the project database and to keep the design project synchronized with the design state.

By default the system performs tasks in the order shown in the task list in “ASICmaster Task Flows and Iterations” on page 2 using the results from each stage as input for the next stage. Additionally, only tasks required to produce the stated outputs are performed and users are prompted for required inputs for each task.

Although ASICmaster performs tasks in a specific order, the system can perform a subset of these operations to suit the current design state unless an operation violates one of the safeguards.

For example, if a design is placed and routed but post-layout delays violate timing requirements, ASICmaster can be directed to only optimize place and route and recalculate the delays. File checking or any other task previously is not repeated unless any of the input files have also changed between runs.

ASICmaster processes a design incrementally by default. It keeps track of successfully completed tasks and does not repeat them. If prerequisites for a task are missing, the system either automatically supplies them or prompts the user to supply them.

Design Hints

The design netlist is the main input for ASICmaster. When creating a design, device usage and ASICmaster features can be maximized by using the following the guidelines:

- Use standard design practices.
- Use the automatic global resource assignments that ASICmaster makes and, if appropriate, add constraints.
- Use the automatic mapping that ASICmaster makes between package pins and the design's I/O signals and, if appropriate, add constraints.
- Avoid over constraining a design so that ASICmaster can apply its algorithms to the best advantage for place and route.
- Implement tristates, memories, and hierarchical blocks with ASICmaster-supported elements.
- Create well structured and well organized design descriptions.
- Avoid asynchronous designs. They are difficult to verify, are less robust, and typically include problems such as hazard and race conditions.
- When gating clocks, ensure that the gate signals are stable when the clock is active. Otherwise, glitches might be propagated through the circuit.
- Use a minimum of logic before the first flip-flop or after the last flip-flop on signals that are to be driven on or off chip to provide as much time as possible to drive the signal off the chip and onto the next chip.

- To ensure that process variations do not adversely affect reliability, always design with worst-case processes in mind.

Using ProASIC Global Routing Resources

Use the automatic global resource assignments that ASICmaster makes. Each ProASIC device includes four dedicated global lines that all tiles in the array can access. These lines exhibit very low skew. When ASICmaster imports the netlist and device data, it can automatically assign unused global resources to the nets with the highest fanouts (for example, clock and reset signals).

If high fanout signals mapped by ASICmaster to global resources are buffered, ASICmaster automatically removes these buffers. ProASIC devices provide automatic buffering for global resources.

Signals can also be assigned to the global resources by using ASICmaster global primitives in netlists, and by specifying constraint files for ASICmaster. Refer to "Global Resource Constraints" on page 67 for additional information.

The checker contains a description of the highest fanout nets in a design and the results of automatic global assignments. If user constraints appear not to have been honored, refer to this report file.

Implementing Memories

All ProASIC devices support embedded memories. Synchronous or asynchronous two port RAMs and FIFOs can be generated by MEMORYmaster and imported into the synthesis and simulation environments. ASICmaster also supports distributed memories. Although this implementation of memories is much less efficient in terms of gate utilization, it may be appropriate for smaller memories in a design.Refer to the *MEMORYmaster User's guide* for additional information.

2

Using ASICmaster

This chapter explains how to select a design and start a design project work session using the ASICmaster graphical user interface. It describes how to use the Design Executive, set project preferences, place and route a design, and generate files such as the bitstream file of device-programming data. Refer to “File and Command Selection” on page 117 for additional information about using the graphical user interface.

Invoking ASICmaster

Each ASICmaster design project processes a single design and creates programming and other files for the design. When a project is started, ASICmaster creates a database for it.

ASICmaster can be used from the graphic user interface, in TTY mode[1], or in batch mode. Many designers use the graphical interface while they are learning to use ASICmaster. As they become more familiar with the system, they change to command-line or batch mode to speed up their work or to automatically repeat ASICmaster task cycles. This chapter describes ASICmaster primarily in terms of its graphic user interface. Refer to “Using ASICmaster in TTY and Batch Modes” on page 87 for information about using ASICmaster in batch and command-line mode.

To invoke ASICmaster:

PC

Choose ASICmaster 5.0 from the ASICmaster menu under Programs in the Start menu.

UNIX

- Type the following command at the prompt:

```
asicmaster&
```

1. *Available on UNIX only*

Main Menu

Starting ASICmaster in graphical mode displays the main menu and copyright information windows as shown in Figure 2-1.

To close the copyright window, click OK at the bottom of the window or click any button in the main menu.

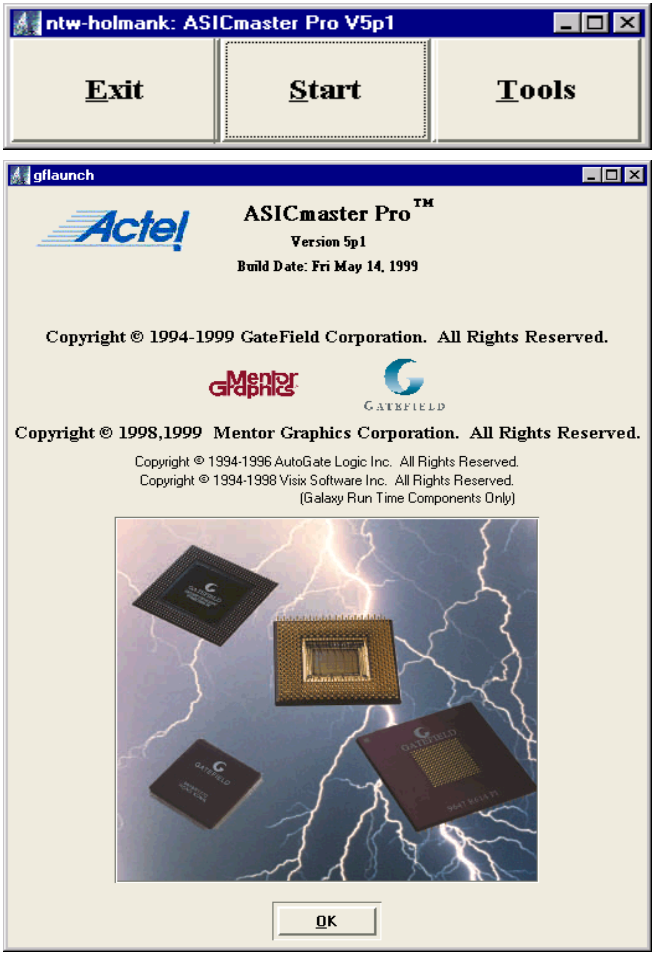


Figure 2-1. Main Menu and Copyright Information Window

The Following commands are available from the Main menu:

Exit

Terminates ASICmaster program.

Start

Starts an ASICmaster work session and displays the Design Selection window to select a design project.

Tools

Opens the tools menu, as shown in Figure 2-2 below.

Figure 2-2. Tools Menu Window

Tools Menu

The following commands are available from the Tools menu:

Start Design

Brings up the design selection window to begin a new design or open an existing one.

Build Embedded Memories

Invokes the Embedded MEMORYmaster, which creates RAMs or FIFOs that are mapped to embedded memory resources on ProASIC devices. Refer to the *MEMORYmaster User's Guide* for additional information.

Build Distributed Memories

Invokes the Distributed MEMORYmaster, which creates RAMs or FIFOs that are mapped to logic tiles on ProASIC devices. Refer to the *MEMORYmaster User's Guide* for additional information.

Partition

Starts the Auspy Partitioning System.

Close

Closes the Tools menu.

Starting a Project Work Session

Clicking the Start button in the Main menu opens displays the Design Selection file browser to select a design, as shown in Figure 2-3.

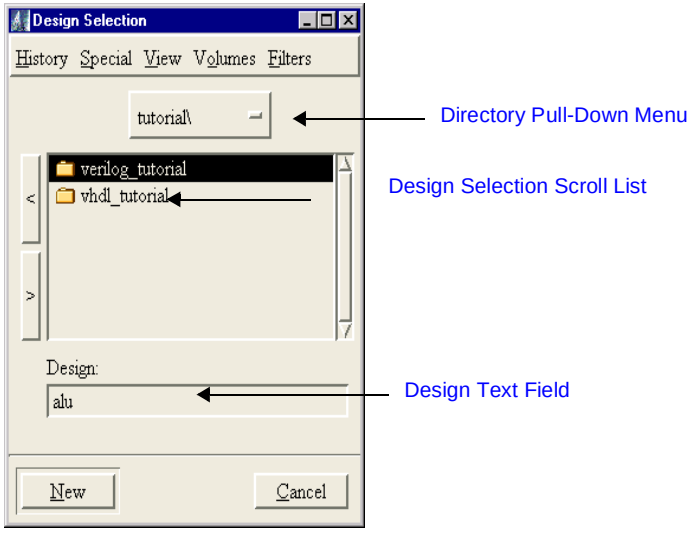


Figure 2-3. Design Selection File Browser

Selecting a Design

Each design project is identified by a design name. Existing designs can be selected or new designs can be created and design project databases are created in the Design Selection file browser. The design must reside in a directory that the user owns or has read, write, and execute permissions. because new files are created in the directory as ASICmaster is used.

The types of files shown in the file browser can be restricted (for example, to Verilog or EDIF or VHDL files) using the browser's Filters menu. Refer to "Restricting the Types of Files Listed" on page 118 for additional information.

To open an existing design file:

1. **Click the design name in the Design Selection scroll list in the browser.**
2. **Press Return or click the Open button.** A message window is displayed to notify you that ASICmaster is opening the design file.

To create and open a new design file:

1. **Enter the name of the file in the Design text field.** You do not need not include the .dtf extension.
2. **Press Return or click the New button.** A message window is displayed to notify you that ASICmaster is creating the design file.

Note: When the Design Selection file browser is displayed, the New and Apply buttons are inactive (dimmed) before entering or selecting a design. When a design name is entered in the Design text field, the buttons become active. If a selection from the directory pull-down menu is made, the New button is replaced by the Open button.

Design Executive Window

When a design file is opened or created, the Design Executive window is displayed. The design name appears in the Window Title Bar, as shown in Figure 2-4. The Design Executive is the primary user interface tool. It includes a menu bar, text fields for entering project preferences, and two buttons that start the main ASICmaster functions.

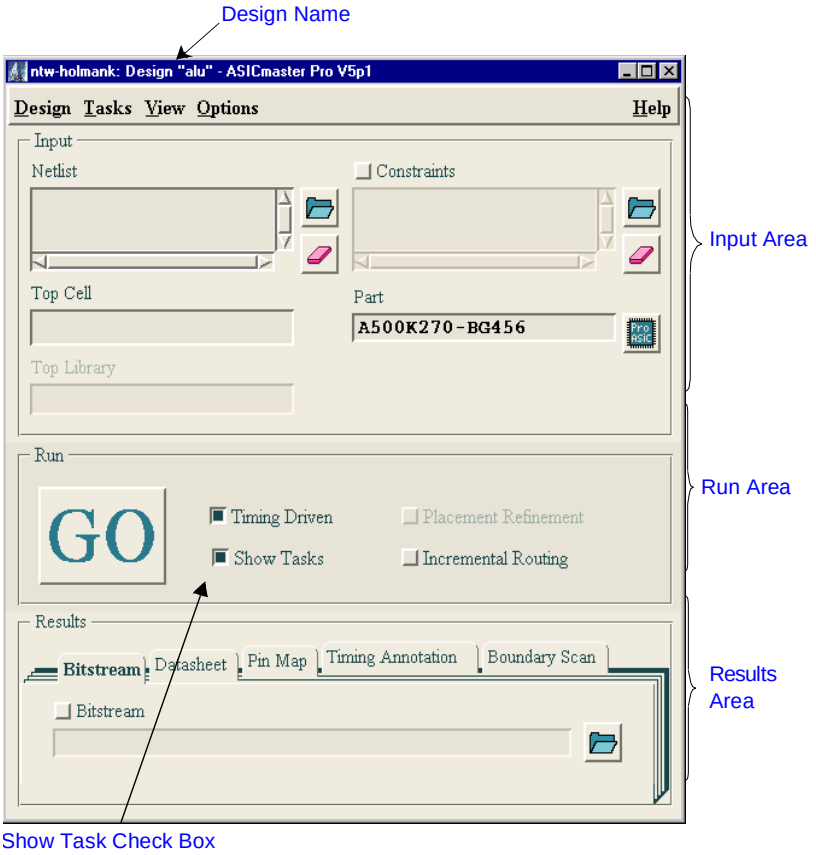


Figure 2-4. Design Executive Window

Design Executive Functions

The Design Executive is used to add and removed from lists, and to order files in lists. For example, if multiple netlists are being imported, they must be ordered so that files that rely on other files for definitions are listed in the order of dependency. The Design Executive is also used to set project preferences, to select the target ProASIC device, and to set place and route options. In addition it is used to select tasks for ASICmaster to perform, including enabling or disabling tasks such as delay calculation and datasheet generation.

The design's parameter settings are stored in an ASICmaster database and affect all task runs. If an existing design has been opened, the text fields in the Design Executive window shows the current settings.

Input Area

The Input area is used to specify what files are used in a design. The Input area contains two types of file management buttons, Folder buttons and Eraser buttons. The Folder buttons open file browsers, which are used to locate and specify files, such as netlists to import for the design.

The Eraser buttons in the Input area remove selected files in file lists in the Design Executive.

Run and Area

The Run area is used to specify place and route parameters and whether the Tasks selection list appears when users start a task run. It includes the Go button, which automatically starts performing tasks or prompts users to select tasks that have not been run successfully for a design, are appropriate for the current design state and match the project preference settings and any task selections made.

Results Area

The Results Area is used to enable delay calculation, file generation, including the device programming bitstream file, and specify names for results files. The Results area includes a Folder button.

Design Executive Menus

The Design Executive menus contain commands that can be selected to perform tasks. Refer to the sections below for additional information performing tasks with a specific menu. Refer to "File and Command Selection" on page 117 For information about using mnemonics and accelerator keys to select menus and enter commands.

Design Menu

"Saving Designs" on page 33.

Tasks Menu

"Limiting the Task Flow" on page 18.

View Menu

"Viewing Report Files" and "Viewing Results" on page 39.

Options Menu

"Specifying Netlist Formats" on page 21 and "Setting Operating Conditions" on page 31.

Setting Design Project Preferences and Managing Files

After starting ASICmaster, the first task is setting the layout and device-programming project preferences for the ProASIC device. The setup information becomes part of ASICmaster database for the design project. The following must be specified:

- One or more netlists.
- A target ProASIC device for the design.
- Place and route options, such as "Timing Driven."

The following preferences can also be set:

- Specify the ASICmaster task flow limits and specifications.
- Identify constraint files for ASICmaster to import.
- Enable post-layout delay calculation and generate back annotation files.
- Generate the device programming bitstream and other output files.

The project preferences are set by entering required data in the Design Executive window and by running commands from the Options and Tasks menus in the Design Executive window. The following sections explain how to limit the task flow, specify netlist files, and set the other parameters that the Design Executive controls.

Selecting Tasks

ASICmaster automatically identifies the prerequisite tasks and data for any run. For example, if constraint files are changed for a design, the next time place and route is run, the system automatically selects the Check Input task. The system does this because ASICmaster must validate each constraint file before creating a layout for the netlist. Before setting parameters and processing a design, ASICmaster automatically defines the task flow as follows:

- Import the netlist.
- Check the input data.
- Place and route the design.

When setting parameters and processing the design, ASICmaster adjusts the task flow and its requirements. At any given time, ASICmaster derives the flow and requirements based on:

- Parameter settings recently set in the Design Executive Window.
- Design flow rules. See Chapter 1, “ASICmaster Overview” on page 1.
- The current design state, defined by stored parameter settings.
- Netlist filenames and path name and the status of each netlist including whether ASICmaster has imported the file and whether the original file has been altered since the import operation.

The system's task flow and prerequisites can be changed at any time without interfering with ASICmaster's prerequisites for task runs and some of ASICmaster's current task prerequisites can be overridden to force the system to perform or omit specific tasks. ASICmaster monitors the state of the design and stores design-related parameter values and a last-modification time-stamp data in a file.

Refer to "The Design (*.dtf) Directory" on page 137 "Redefining Task Prerequisites and Flow" on page 36," and "Redefining Task Prerequisites and Flow" on page 36 for additional information.

Limiting the Task Flow

The task flow can be limited by defining the last task in the sequence of tasks that ASICmaster performs. For example, a design's layout can be optimized without generating the device programming bitstream or other external output files. If ASICmaster does not generate these files, runs proceed faster. There are two ways to limit the task flow:

- Use the Tasks Menu.
- Use the Tasks selection list in the Tasks window.

In some cases, the task flow can not be limited. For example, If a netlist for the design has been altered, the Tasks menu cannot be used to omit Check Input. Or if Import Netlist is selected, place and route cannot be omitted. See "Redefining Task Prerequisites and Flow" on page 36 for additional information about limiting the task flow.

Using the Tasks Menu

The Tasks menu, as shown in Figure 2-5 on page 19, limits ASICmaster to performing a selected task and all related tasks that are required for successful task performance for the current design state. It also Immediately executes those tasks or displays the file selection list.

To use the Tasks menu:

1. **Click the Tasks menu in the Design Executive Window.** The Tasks menu is displayed.

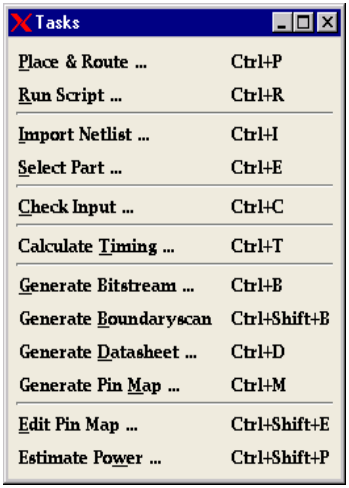


Figure 2-5. The Tasks Menu

2. **Choose a task by double-clicking its name.** ASICmaster limits the task flow and immediately performs tasks.

Using the Tasks Selection List

The Task Selection list, shown in Figure 2-6 on page 20 displays the tasks ASICmaster has selected for the current run. The Task Selection list can be used to override some of ASICmaster's task prerequisites to force the system to perform or omit specific tasks.

To use the Tasks Selection list during a ASICmaster run:

1. **Check the Show Tasks box in the Design Executive window.** The Tasks Selection list is only displayed during each time a task run is started if this box is checked. For information on using the list, see "Redefining Task Prerequisites and Flow" on page 36.

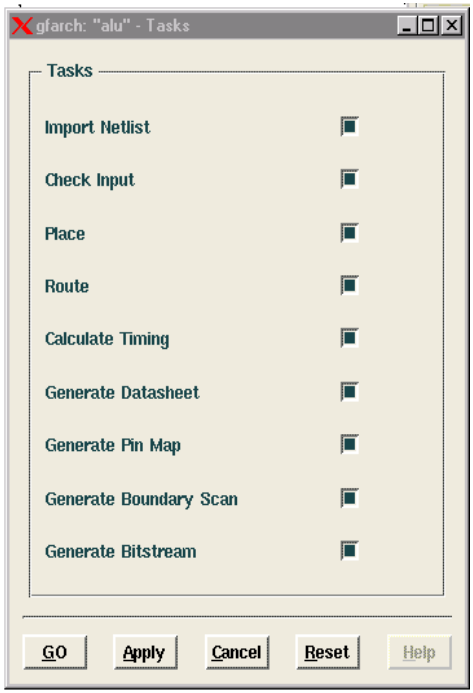


Figure 2-6. The Tasks Selection List

2. **Check the box next to the task(s) to perform.** Uncheck any tasks that you do not want performed.

3. **Click the GO or Apply button at the bottom of the Tasks window.** Selecting a task on the list does not immediately execute it or its associated tasks.

Specifying Netlist Formats

The netlist file type can be set in ASICmaster. ASICmaster supports EDIF 2.0.0, Verilog, and VHDL netlist formats. Before ASICmaster imports the netlists specified, it verifies that the netlist files being imported are in a supported format. Refer to “Supported Netlist Formats and Constructs” on page 129 for additional information

To set the netlist file format for a design:

1. **In the Design Executive window, choose the Design command from the Options menu.** The Design Options window is displayed, as shown in Figure 2-7.

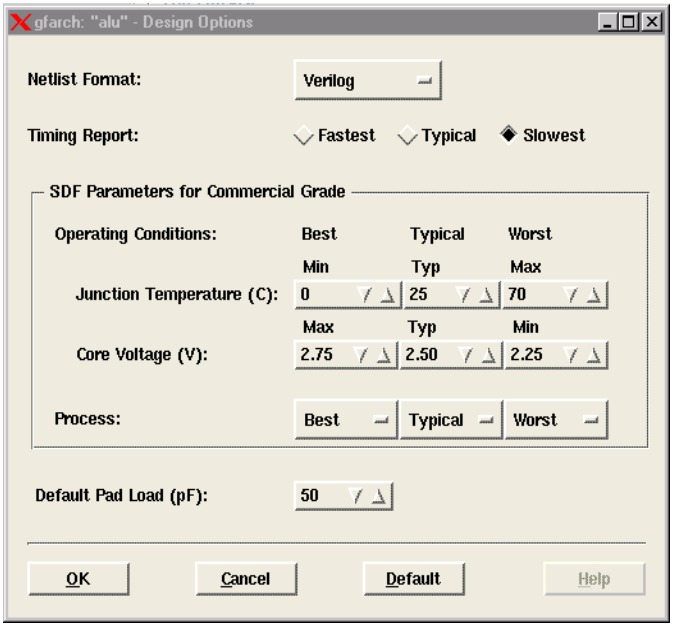


Figure 2-7. Design Options Window

2. **Click the Netlist Format pull down menu and choose the appropriate netlist format.**
3. **Click OK.**

Selecting Netlists

To begin a design, one or more netlist files must be selected. Once selected, the netlist(s) are displayed in the Input area of the Design Executive window, as shown in Figure 2-8.

Note: All netlists for a design must be in the same format.

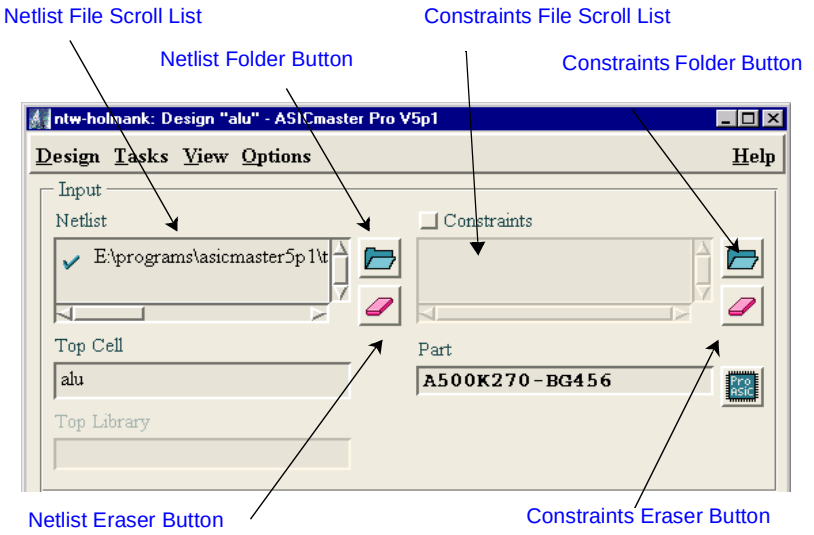


Figure 2-8. Design Executive Window Input Area

To add netlists:

1. **Click the Netlist Folder button in the Design Executive window.** The Netlist File Selection browser is displayed, as shown in Figure 2-9.If the file format has been set, the filenames that appear are limited to those in the format specified.

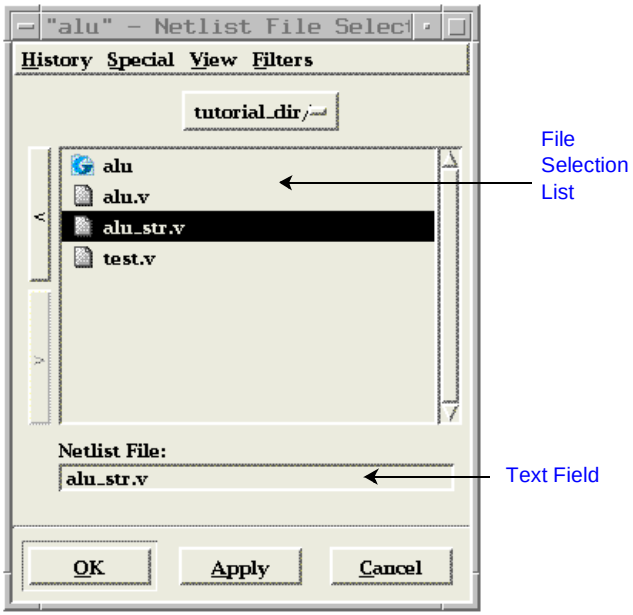


Figure 2-9. Netlist File Selection Browser

2. **(Optional) Specify file format.**If the file format has not been set, choose a format from the Filters menu to restrict the type of file that is displayed.

3. **Select a netlist.** Double-click the name of the netlist file in the file selection list. You can also enter the name of the netlist in the Netlist File text field. To select multiple files, press and hold the <Ctrl> key and click multiple filenames. This also deselects filenames.

4. **Click Apply or OK.** Apply adds the filename and leaves the file browser open. OK adds the filename and closes the file browser.

If an arrow precedes an added netlist, the netlist has not been checked and included into the design database. After successfully importing the netlist the arrow changes to a check mark, as shown in Figure 2-10.

Note: If specifying a file that no longer exists in the ASICmaster directory, its name is preceded with an exclamation mark (!).

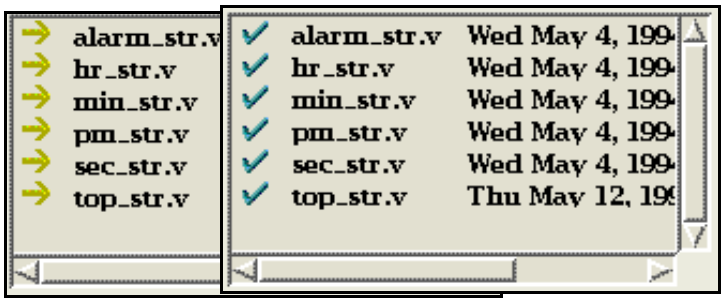


Figure 2-10. Netlist Files Before and After You Import the Netlists

To remove netlists:

Select the netlist in the Input area of the Design Executive window and click the Eraser button next to the Netlist file scroll list, as shown in Figure 2-8 on page 22.

Reordering Netlists

Netlists may need to be reordered. Netlists must be ordered so that any file that depends on another file is listed in order of dependency. A dependent file is listed below the file it is dependent upon.

To reorder netlists:

Select the netlist in the Input area of the Design Executive window and drag the filename to the new location.

If a file name is moved down in the list, it is inserted below the location selected. If a file name is moved up in the list, it is inserted above the location selected.

Specifying the Top Cell and Top Library

In most cases, ASICmaster automatically identifies the design's top cell and top library. These are displayed in the Top Cell and Top Library text fields in Input area of the Design Executive window. Specify the top cell only if its selection could be ambiguous. For example, Verilog netlists can include multiple cells within the design's top level. If the design is VHDL-based, the top cell must be specified.

The top cell should also be specified if a netlist includes a test fixture file at the top of the design. Otherwise, ASICmaster tries to select the test fixture cell.

The Top Library field applies only to EDIF and VHDL designs. Specify the library only if there is a need to eliminate ambiguity. For EDIF and VHDL designs, fill in the Top Library text field if a cell name in the Top Cell field has been specified and that cell name exist in multiple directories (libraries).

Specifying a ProASIC Device

A device must be specified for a design. Select the device based on capacity, package type and size. To check a device for capacity, compare the number of gates used in the design with the maximum number of usable gates in the device. Ensure that the number of usable gates in the device is equal to or greater than the number of gates in the design.

To select a package, select a device that uses the type needed (for example, BGA or PQFP) and make sure that the device has at least as many user I/Os as the design requires.

For complete information about the ProASIC devices, usable gates and packages, see the *ProASIC 500K Family* Data Sheet.

To specify the ProASIC device:

1. **Click the ProASIC button in the Input area of the Design Executive window.** The Part Selection file browser is displayed, as shown in Figure 2-11**.**

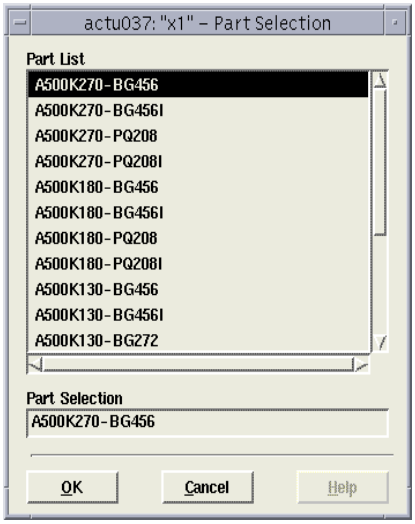


Figure 2-11. Part Selection Browser

2. **Select a device**.
3. **Click OK**. The device name appears in the Parts field of the Design Executive window.

Selecting Constraint Files

ASICmaster uses the netlist, device selection, and other Design Executive settings to automatically develop the following parameter settings for the database: Global resource, timing, I/O, and other physical design parameter settings and project preferences.It is possible to use constraint files to supplement the physical design parameters and to control the degree of netlist optimization. However, constraints should be limited as much as possible and introduced late in ASICmaster's iterative process. Constraints can limit ASICmaster layout and optimization efforts.

For example, if I/O assignments have already been assigned at the board level, a constraint file that maps I/O signals to package pins would be necessary to ensure proper pin placement. One way to achieve this goal is to use the I/O pin map that ASICmaster can generate for the ProASIC device (see "Enabling Delay Calculation and File Generation" on page 29).

To add constraint files:

1. **Click the Constraints Folder button in the Design Executive window (see Figure 2-8 on page 22).** The Constraint File Selection browser is displayed.If the file format has been set, the filenames that appear are limited to those in the format specified.
2. **(Optional) Specify file format.**If the file format has not been set, choose a format from the Filters menu to restrict the type of file that is displayed.
3. **Select a constraint file.** Double-click the name of the file in the file selection list. You can also enter the name of the file in the File text field. To select multiple files, press and hold the <Ctrl> key and click multiple filenames. This also deselects filenames.
4. **Click Apply or OK.** Apply adds the filename and leaves the file browser open. OK adds the filename and closes the file browser.

If an arrow precedes an constraint file, the has not been checked and included into the design database. After checking the file, the arrow changes to a check mark.

After adding a constraint file to the Constraints File list, it can be viewed using the View command or by double-clicking the file, which brings up the file in the specified editor.

Note: If specifying a file that no longer exists in the ASICmaster directory, its name is preceded with an exclamation mark (!).

To remove constraint files:

Select the constraint file in the Input area of the Design Executive window and click the Eraser button next to the Constraint file scroll list.

Reordering Constraint Files

If any constraint definitions appear in multiple files, order the files so that those containing the most up-to-date definitions appear last on the list.

To reorder netlists:

Select the constraint in the Input area of the Design Executive window and drag the filename to the new location.

If a file name is moved down in the list, it is inserted below the location selected. If a file name is moved up in the list, it is inserted above the location selected.

Specifying Place and Route Options

Place and route project parameter must be set to instruct ASICmaster to run in a timing driven and/or ECO mode. Three parameters can be set, Timing Driven, Placement Refinement, and Incremental Routing

Timing Driven

The Timing Driven check box enables the timing driven place and route system. In this mode, placement is optimized to minimize the routing congestion timing violations according to the path constraints defined by the user. In the absence of user defined constraints, internally generated path constraints are used. Actel recommends generating path constraints using a synthesis tool so that design specific characteristics are accounted for.

It is also possible to influence routing by specifying net criticalities. These nets are routed in the order of these criticalities, with the highest first. Also, in conflict situations with other nets, the net with the lowest criticality moves to a less optimal position.

Placement Refinement

The Placement Refinement check box can be used to improve an existing placement. This feature assumes that an initial placement constraint file has been specified for the design to be improved.

When the placement task is finished, ASICmaster always produces a placement constraints file in the design directory called:

`<design>.dtf/last_placement.gcf`

This file contains all the information about the latest placement. Blocks with fixed placement constraints regenerate fixed placements constraints, while the others generate initial placement constraints. The existing constraints files can be edited to remove any prior placement constraints and copy the "last_placement.gcf" file one level up and add it to the list of constraints files.

Move or copy the file to use it as a constraint file. Otherwise, it is overwritten by any subsequent placement if left in its original location.

Incremental Routing

The Incremental Routing check box allows a design to fully route when some nets failed to route during a previous run. It can also be used if the incoming netlist has undergone an E.C.O. (Engineering Change Order). Incremental routing should only be used if a low number of nets fail to route (less than 20 open nets or shorted segments).

A high number of failures usually indicates a less than optimal placement (if using manual placement through macros for example) or a design that is highly connected and does not fit in the device. If a high number of nets fail, relax constraints, remove placement constraints, or select a bigger device and rerun routing.

Enabling Delay Calculation and File Generation

The Results area in the Design Executive Window controls parameter settings for calculating post-layout delays and creating timing annotation files. It also controls settings for generating the bitstream device programming file, the boundary scan file, the I/O pin map, and the datasheet for the ProASIC device.

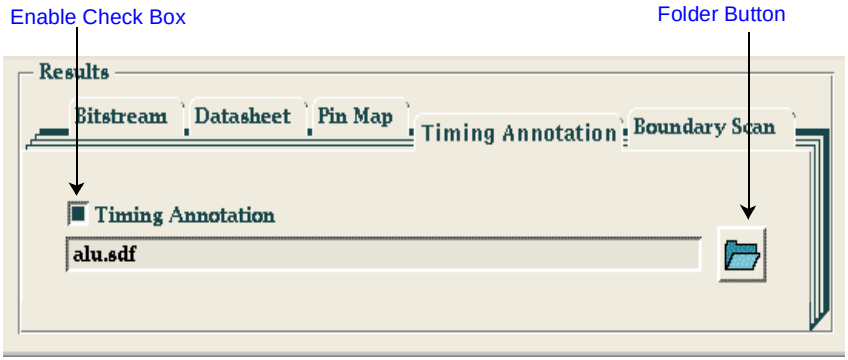


Figure 2-12. Results Area of the Design Executive Window

Use the following procedure to enable file generation for any of the results files shown in the Results area.

1. **Click the tab of the result file you want to generate.** Figure 2-12 shows the Timing Annotation tab.
2. **Generate the result file.** Click the enable check box. ASICmaster automatically creates and names the results file.

   You can also Click the Folder button in the Results area to open a file browser an specify a results filename or enter a pathname in a tab's text field. This automatically checks the function's enable check box. Use the appropriate file extension, listed in Table 2-1.

Table 2-1. File Extensions for Results Files

File Type	Extension
Bitstream device-programming data	.bit
Boundary Scan	.bsd
Datasheet	.txt
Pin map	.gcf
Timing annotation	.sdf

To disable file generation:

Un-check the Enable check box in the appropriate Results area tab or by un-checking the check box in the Task Selection list.

Setting Operating Conditions

Operating conditions (including temperature, voltage, and process) can be set for a design. Also, users can specify for which operating conditions they want the timing report done and which pad load. ASICmaster needs to assume for its I/O intrinsic calculations. By default a load of 35 pF is assumed. ASICmaster uses this information as it develops pre-layout and post-layout timing data. Table 2-2 describes the operating conditions.

Table 2-2. Operating Conditions

Operating Condition	Description
Junction Temperature[a]	range from 0°C to 70°C for commercial and -40°C to 85C for industrial
Minimum	range from –55°C to Typical
Typical	range from Minimum to Maximum
Maximum	range from Typical to 125°C
Voltage[b]	
Minimum	range from 2.00 volts to Typical
Typical	range from Minimum to Maximum
Maximum	range from Typical to 4.00 volts
Process[c]	
Best	Fastest possible process you could get
Typical	Most likely process that you would get
Worst	Slowest possible process you could get
Report[d]	
Fastest	For Maximum Voltage and Minimum Junction Temperature and for best Processes specified
Typical	For Typical Voltage and Typical Junction Temperature and for Typical Processes specified
Slowest	For Minimum Voltage and Maximum Junction Temperature and for Worst Processes specified

a. The temperature increment is 1°C.

b. The voltage increment is 0.05 V.

c. To ensure that process variations do not adversely affect reliability, always design with worst-case processes in mind.

d. To specify the operating conditions for which you want the timing values reported.

To set operating conditions:

1. **Select the Design from the Options menu in the Design Executive Window.** The Design Options Window is displayed, as shown in Figure 2-13.
2. **Set appropriate parameter values.** Table 2-2 on page 31 describes the operating conditions that can be set**.**
3. **Click OK.**

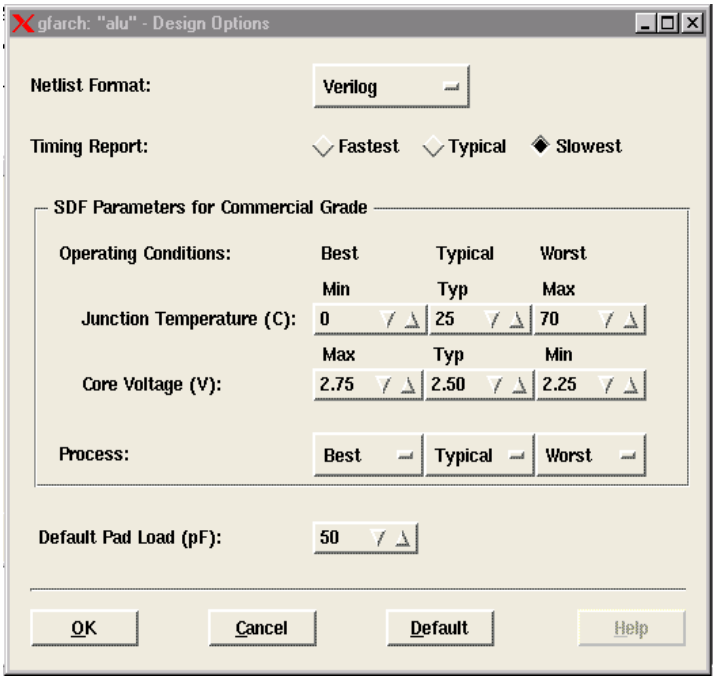


Figure 2-13. Operating Conditions

Setting the Text Editor

By default, ASICmaster starts the vi (UNIX) or Notepad (PC) text editor to edit files. The default setting can be changed by choosing another supported text editor in the search path.

To set the text editor:

1. **Select the User command from the Options menu in the Design Executive Window.** The User Options Window is displayed, as shown in Figure 2-14.

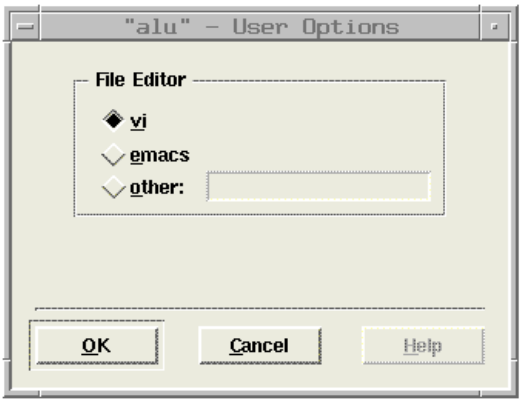


Figure 2-14. User Options Window

2. **Select an editor.** If Other is selected, a path to an executable must be specified in the text field. Make sure that this executable is in the search path.
3. **Click OK.**

Saving Designs

While working, save designs regularly using the Save or Save As command in the Design Executive's Design menu. ASICmaster automatically updates parameter settings in the .amrc start-up file. This file configures ASICmaster when it starts. Refer to "The .amrc Startup File" on page 147 for information about the .amrc file.

Running an ASICmaster Task Cycle

The Layout Viewer and log files can be displayed by using commands on the Design Executive's View menu. For more information on Design Executive menus and commands Refer to "File and Command Selection" on page 117 for information about available commands and "Viewing Results" on page 39 for information about viewing log files.

Session Report

The session report provides information about the task run and continuously updates as tasks progress. Figure 2-15 shows a sample session report at the start of the routing process after a successful place operation.

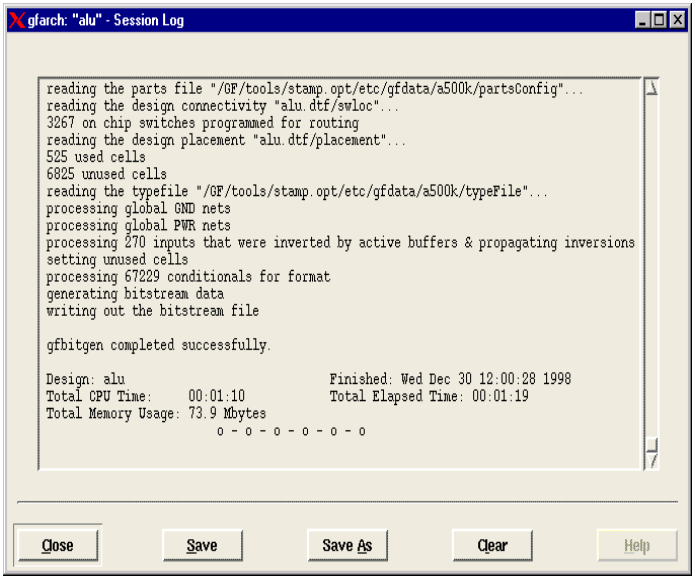


Figure 2-15. Sample Session Report

Starting the Task Run

To start a task run, select a task from the Tasks menu in the Design Executive Window or click the GO button in the Run area. Figure 2-16 shows where the menu and button are located.

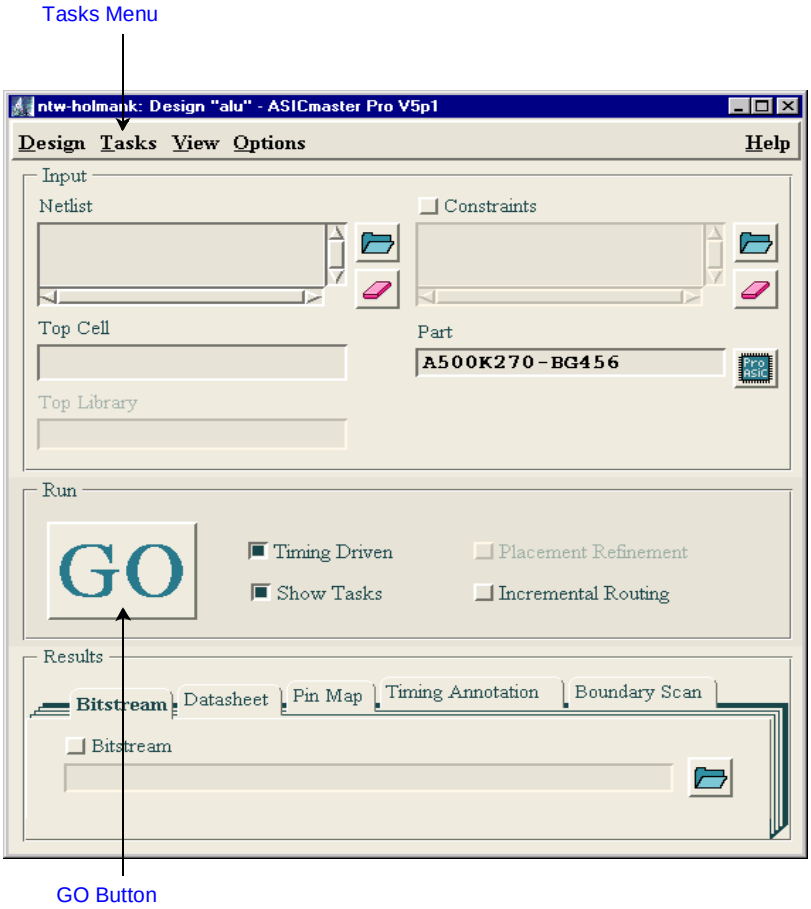


Figure 2-16. Starting a ASICmaster Task Run

Redefining Task Prerequisites and Flow

Whenever the GO button in the Design Executive window is clicked, the Tasks Selection List is displayed, as shown in Figure 2-17. The Tasks list shows ASICmaster's task selections for the run. The Task selections are synchronized with the current design state (see "Selecting Tasks" on page 17), the parameters users have set (described in this chapter), and the design-flow rules (see "ASICmaster Overview" on page 1).

Use the Tasks selection list to limit the task flow to a selected task and tasks related to its successful completion or to override some of ASICmaster's task prerequisites to force the system to perform or omit specific tasks.

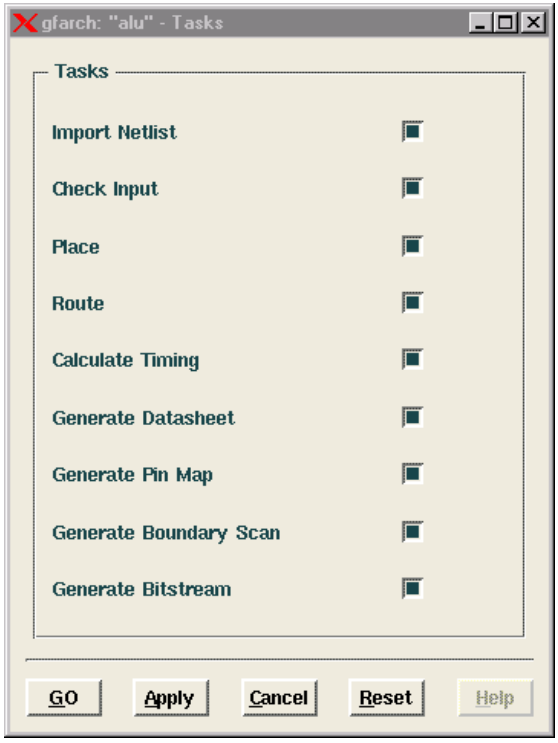


Figure 2-17. Task Selection List

This section describes how to override the prerequisites. For information on how to limit the task flow, see "Limiting the Task Flow" on page 18.

To change the system's task flow:

1. **Check the check box to the right of a tasks to toggle the box on or off.** A check box that appears dimmed (gray) indicates that you previously disabled that task (as described in "Limiting the Task Flow" on page 18). to enable the task, check the box again.
2. **Click the GO button in the Task Selection list to start the task run.**
3. **To restore ASICmaster's default selections, click the Reset button at the bottom of the Tasks Selection list.**

Estimating Power Consumption

ASICmaster can be used to estimate the power consumption of a design.

To estimate power consumption:

1. **Select the Estimate Power command from the Tasks menu.** The Power Estimation window appears as shown in Figure 2-18 on page 38.
2. **Enter all required parameters that characterize the particular design.** The power consumption will automatically be updated.

   If power consumption is calculated using a different method, select Entered from the pull-down menu in the Results area of the Power Estimation window and enter the power figures to calculate the junction temperature. The junction temperature is shown as a function of the environment, the packaged used, and the power consumption. Refer to the *A500K ProASIC Family* Data Sheet. For more information about power consumption.

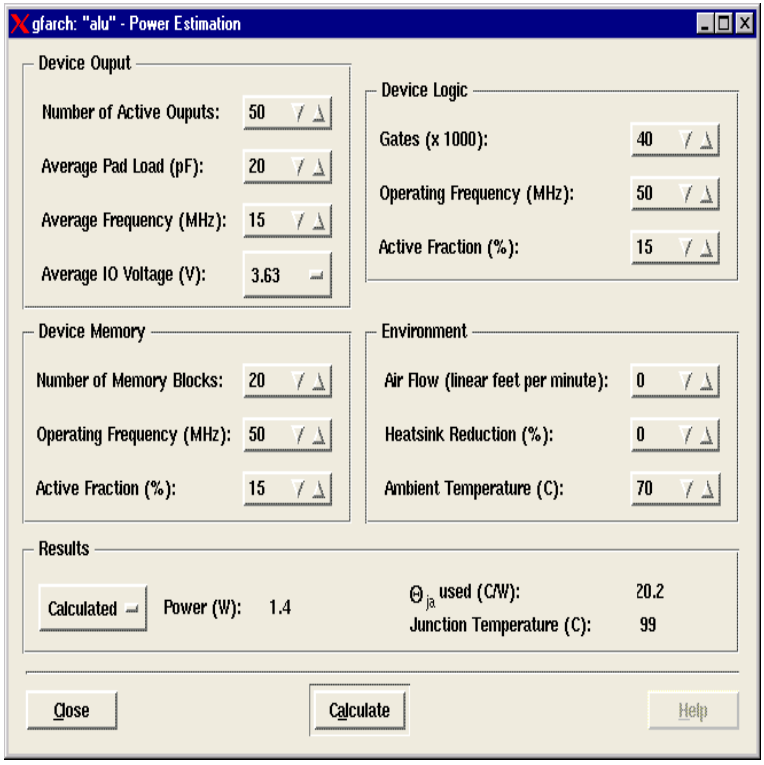


Figure 2-18. Power Estimation Window

3

Viewing Results

This chapter explains how to access and interpret the results that ASICmaster produces, includin as report files, output data files and the design layout.

Viewing Report Files

ASICmaster creates three types of reports, Session reports, Command reports, and Task reports. Each report is saved in a file with the .log extension.

To view a report files, select the apporpriate command from the View menu in the Design Executive window, as shown in Figure 3-1.

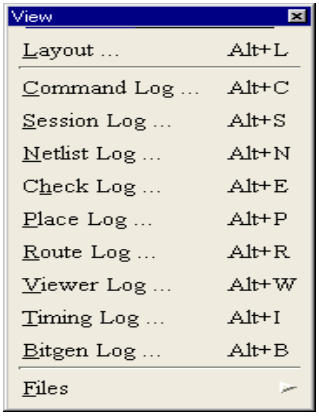


Figure 3-1. View Menu

Session Report

The session report displays information as tasks are running and can remain displayed as ASICmaster runs. All tasks messages such as PROBLEM and NOTE are displayed in the Session Report Window as shown in Figure 3-2.

The following describe of the buttons in the Session report window.

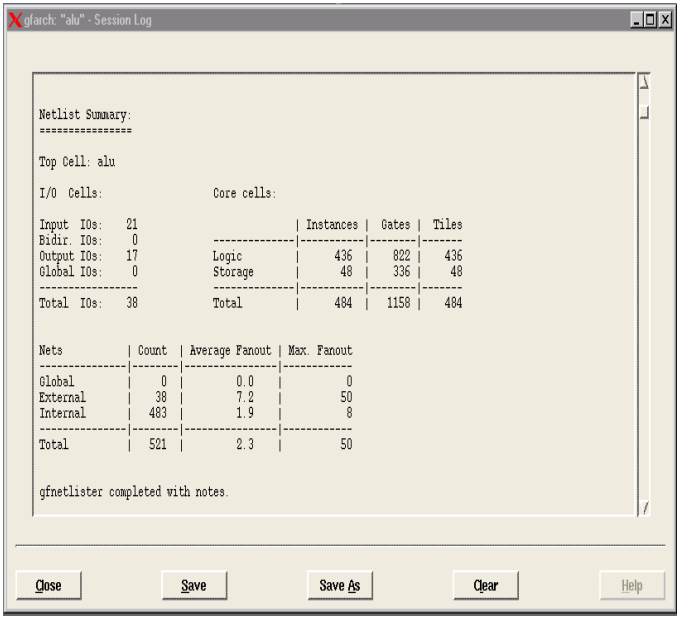


Figure 3-2. Session Report

Close

Closes the report window.

Save

Saves the current information to a filename previously specified.

Save As

Displays a file browser in which a new filename can be specified to save the current report to that file.

Clear

Clears the contents of the report window.

Command Report

The Command report, as shown in Figure 3-3, maintains a history of all keyboard and mouse input commands and saves them to batch-file format when the report is saved. The system does not automatically save the Command report.

The command report can also be used to enter commands in the Command text field, similar to TTY mode, append commands to the report, and immediately execute them. Refer to “Batch Script Syntax” on page 143 for information about command line syntax.

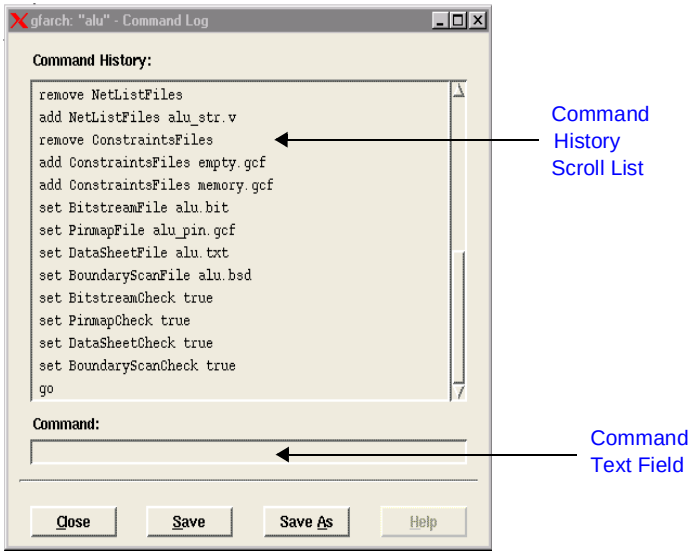


Figure 3-3. Command Report Window

To enter, append or execute a command:

Type a command in the Command report Command text field and press Enter or Double-click a command in the report’s Command History scroll list.

To save a batch file or report:

1. **In the Command Report window, click the Save As button.** A file browserdisplayed.

- **Enter a filename in the text entry field.** Use the .scr extension for batch script files and the .log extension for report

2. **Click OK.**

To execute a batch file:

Select the Run Script command from the Tasks menu in the Design Executive window. Users can also reference the batch script file as an option to the “asicmaster” command. Refer to “Using ASICmaster in TTY and Batch Modes” on page 87 for additional information about batch file execution and creation.

Task Reports

ASICmaster generates a separate report for every important operation it performs. These reports contain information about the results of an operation’s most recent run and can be viewed while ASICmaster is running. Task reports can be displayed as ASICmaster runs. However, ASICmaster does not automatically update task reports. To update the task report, click Update button in the Task report window.

The following Task reports can be viewed by selecting the appropriate command from the View menu in the Design Executive window:

- Netlist Log
- Check Log
- Place Log
- Route Log
- Viewer Log
- Timing Log
- Bitgen Log

Task Report Format

All Task reports have a similar format. The header of the report shows design information, the start date, and the start time. Parameters and switches on the command line are shown next. All messages are then recorded. A message summary follows the individual messages. The trailer displays information such as the finish date and time of the latest update, the total CPU time used and the total elapsed time of the run.

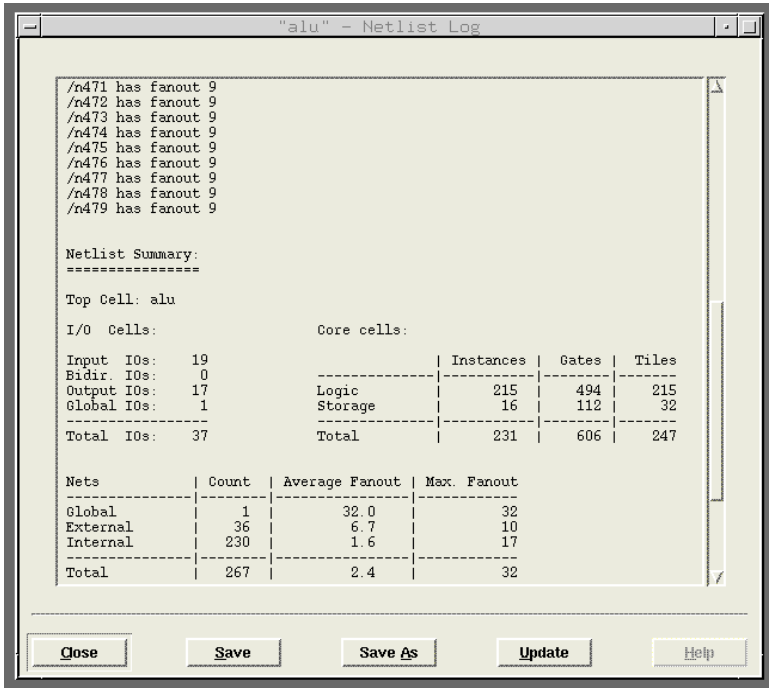


Figure 3-4. Sample Netlist Report

Task Report Messages

Task reports display the following types of messages:

PROBLEM

Indicates an error related to design data. The task will not complete its run until the error is corrected.

NOTE

Displays a warning related to design data to notify users that the system has taken action to resolve a problem. The task continues its run.

INFO

Displays general information related to design data and task activity.

SYSTEM

Indicates an error related to the ASICmaster system.

Netlist Report

The Netlist report provides information about netlist size, fanout, global signals, macrocell use prior to optimization. Other useful information is also included in this report.

Check Report

Check report contains information about the device specified, the utilization of core and I/O slots, and global resources.It also displays the number and type of constraints processed. and the amounts and types of netlist optimization performed.

The Check report can be used to:

- Learn about constraints that were imported
- Check the fanout distribution
- Examine information about high fanout nets that were promoted to global routing resources
- Determine the number of blocks removed during netlist optimization.

Place Report

The place report displays information about the placement results, the various improvement phases that were performed during the place operation, and the timing budget and the slack that was achieved.

Route Report

The route report provides information about the routing results. It can be used to determine if incremental routing was enabled, if routing

completed successfully, and the reason for routing failures, such as the number of open nets.

Viewer, Timing, Bitgen Reports

The Viewer, Timing, and the Bitgen reports record any exceptional situations or conditions that occurred during the latest run of the task.

Timing Report

The timing report contains information about the longest PAD to FF, FF to FF, FF to PAD, and PAD to PAD paths as well as information about the deepest paths.

Viewing Output Data Files

Output data filescan be viewed to examine (and in some cases to edit) the results of ASICmaster runs. ASICmaster generates the following output data files:

- Timing annotation
- Device-programming bitstream
- Datasheet
- Pin map
- Boundary Scan Description

To display an output data file:

1. **Choose Files from the View menu in the Design Executive Window.** n extended menu that lists the most recent output files

generated by ASICmaster for the current design is displayed, as shown in Figure 3-5.

2. **Select the file to display.** Netlist and Constraint files can also be displayed by selecting the Netlist or Constraints commands on the extended menu. The files are displayed using the default ASICmaster text editor or with an editor that been specified in the Option field in the Design Executive Window..

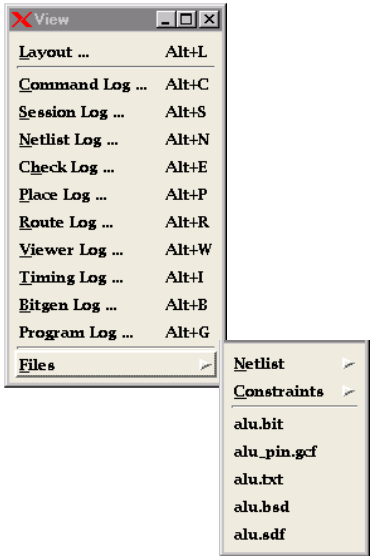


Figure 3-5. View menu

Bitstream

The bitstream device programming file contains a hexadecimal memory map of the device layout. It is downloaded to a device programmer and used to program a ProASIC device.

WARNING: Do not edit this file.

Pin Map

The I/O pin map file contains signal assignments to device package pins. The file is written in the constraint file format and can be used as an input constraint file. By doing this, current pin assignments are frozen the next

time the place task is run. This file can be edited by choosing the Pin Map Editor command from the Task menu in the Design Executive window. Refer to “Placement Contraints” on page 73 for information about using the pin map as a constraint and “Constraint File Syntax” on page 173 for information about constraint file syntax.

Datasheet

The datasheet, as shown in Figure 3-6, displays a summary of the configured ProASIC device after the layout is available. It contains footprint information and package pin assignments for I/O signals.

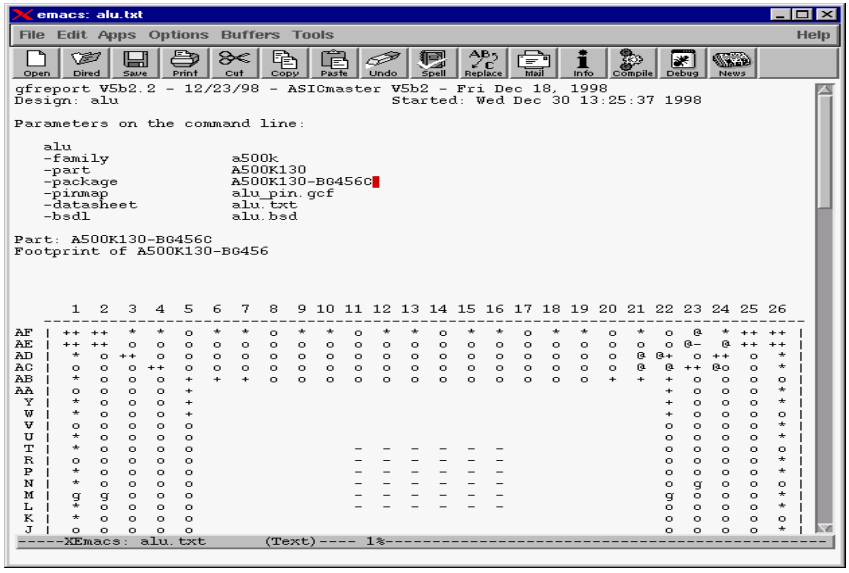


Figure 3-6. Sample Datasheet

Timing Annotation

The timing annotation file contains intrinsic and interconnect post-layout delay information calculated by ASICmaster. This information is written in an SDF format accepted by the static timing analyzer. This file can be used to backannotate post-layout delays to the design environment, a simulator, or a timing analyzer.

Viewing the Design Layout

The design layout displays the results of place and route. The results can be used to get information to guide later place and route operations, if necessary. The Viewer creates no new data. It is a tool for identifying problems and providing insights to solve them. Figure 3-7 shows the Viewer window. The design layout shown in the Viewer is symbolic and does not reflect the actual ProASIC device resources. It shows relative placement positions and symbolic routes reflecting the congestion density of the layout. This section describes how to use the Viewer, window descriptions and a summary of menu commands and accelerator keys

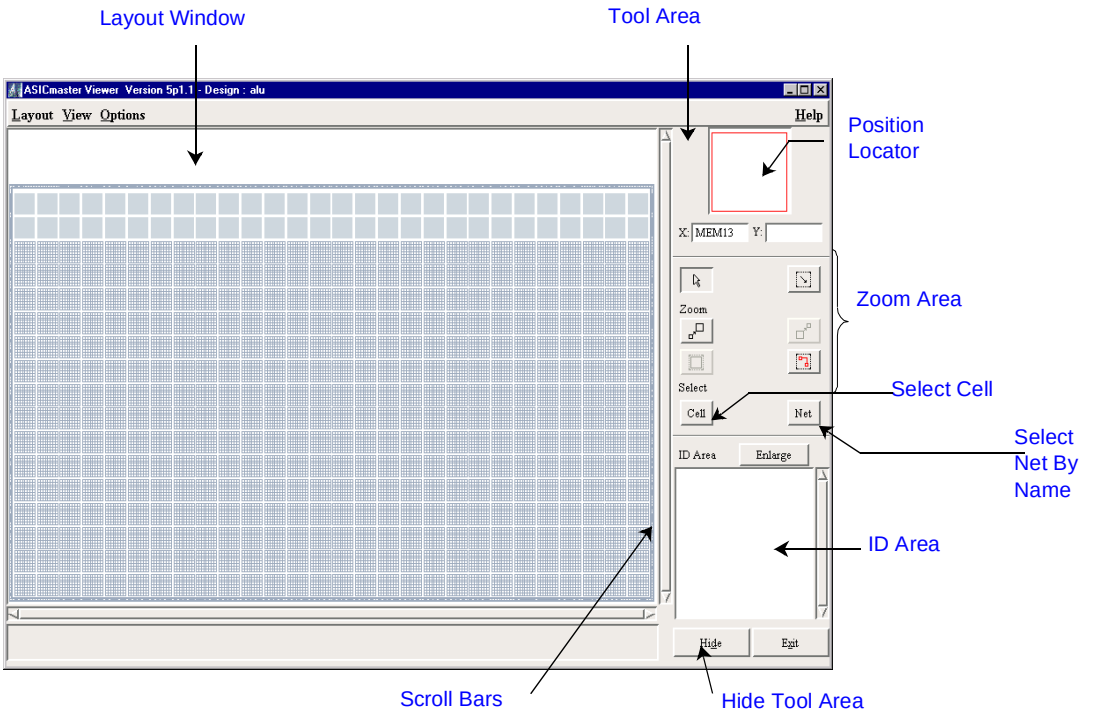


Figure 3-7. ASICmaster Viewer Window

To access the Viewer:

choose the Layout command from the View menu in the Design Executive window. Although the Viewer can be invoked at any time during the layout design process, it is most useful after placement and before routing or after placement and routing.

The Viewer can remain open while performing design tasks such as routing. When the tasks are complete, layout can be updated. In the Viewer's Layout menu, select Update. The Viewer retrieves and displays the most recent design information.

To display multiple versions of the Viewer:

3. **choose the Layout command from the View menu in the Design Executive window and leave it open.**
4. **Make changes to the original design, then rerun place and route.**
5. **Display a second Viewer by choosing Layout from the View menu in the Design Executive window.**

Tool Area

The Viewer's Tool area, shown in Figure 3-7 on page 48, provides buttons for performing the operations described in the following sections.

Hide

To enlarge the available layout display and hide the Tool area, use the Hide button. To redisplay the Tool area, select Show Tool Area from the Options menu.

Zoom In

Click this button to magnify the design layout by a factor of 2.

Zoom Out

Click this button to shrink the design layout by a factor of 2.

Zoom Selected

Click this button to fit all selected objects into the Layout window.

Zoom Area

Click this button to maginfy a specific area in the Layout window by clicking and dragging on the area.

Full View

Click this button to fit the entire design into the Layout window.

The Locator

The Locator, on the top right corner of the Tools Area, as shown in Figure 3-8, provides a quick way to move the Layout window to any position in a magnified layout.

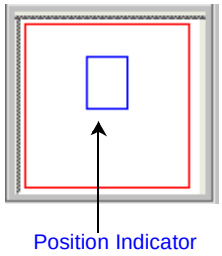
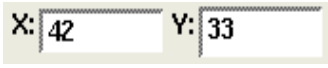


Figure 3-8. Position Indicator

Position Locator

The X and Y coordinate fields of the Position Locator display the current mouse position in the Layout window. These are the X and Y coordinates of the tile where the cursor is placed. Their minimal values (1,1) are indicated in the lower-left corner of the Layout window, and their maximum values, indicated in the top-right corner and depend on the device that is targeted. This is the same coordinate system used in constraint files to define placement locations.

In the I/O tiles, the X field shows the location. For example, N 5 stands for 5th pin on the North side,while the Y field remains empty.



Pointer

The Pointer is used for object selection and is the default operation in the Viewer. If it is not active, click the Pointer button to turn it on. Information about a selected objects is displayed in the ID area.

To select one object, move the pointer over the object and click the left mouse button. To Select all objects in an area, click and drag across the objects to selected.

Information about multiple selected objects is not automatically displayed in the ID area. to display the information, choose the Identify comand from the View menu.

Net

Click the Net button to display the Select Net by Name window, as shown in Figure 3-9, to search for and select nets by name.

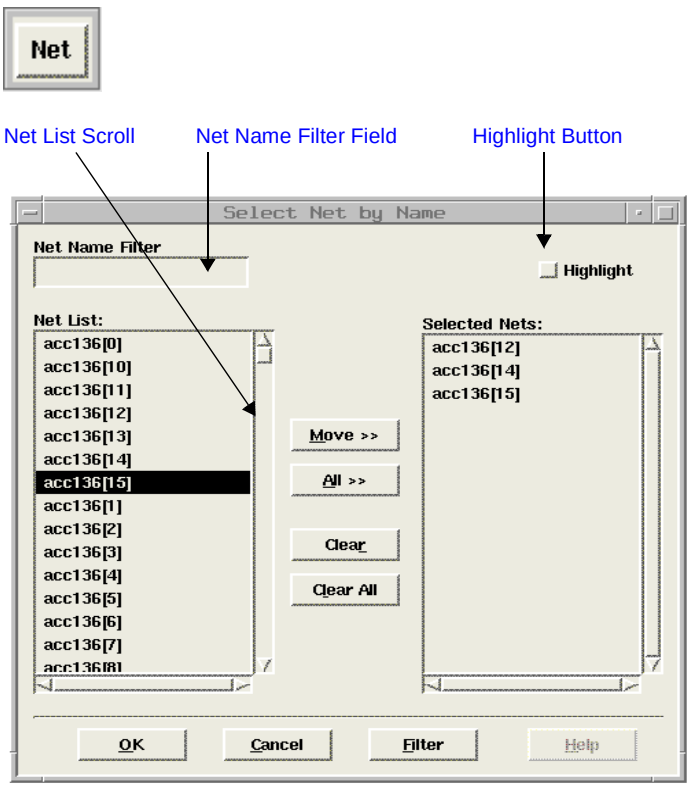


Figure 3-9. The Select Net By Name Window

The Select Net by Name window allows yout to add or remove objects in the selected list and provides the following features:

- Net Name Filter text field, which supports the use of wildcard characters to filter out nets.
- Net List, a scroll list in which either all nets are displayed or a subset of the nets are displayed when a filter is defined. Users can select nets they intend to highlight in layout view.
- Selected Nets, a scroll list that contains all entries selected from the Net List area.
- Highlight button to provide a quick way to highlight all nets in the Selected Nets list.

These following buttons are located in the middle area of the Select Net by Name window.

Move	Copies the selected entries in the Net List to Selected Nets.
All	Copies all entries in the Net List to the Selected Nets list.
Clear	Clears the selected items in the Selected Nets list.
Clear All	Clears all entries in the Selected Nets list.

These following buttons are located at the bottom of the Select Net by Name window.

OK	Confirms the selection process and closes the window.
Cancel	Cancels the selection process.
Filter	Starts the filter function.
Help	Starts the help system.

Using the Filter Function

The filter function performs a simple search. To perform the search, users must first specify a pattern. A pattern is a combination of strings and wildcard characters. The filter function accepts two commonly used wildcard characters:

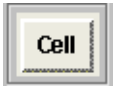
- The question mark (?) matches exactly one single character.
- The asterisk (*) matches zero or more characters in sequence.

For example, if users want to search for all the internal nets of subcircuit L3/addu_1, enter the following in the Net Name text field:

```
L3/addu_1/*<Return>
```

Cell

Click the Cell button to display the Select Cell by Name window, to search for and select cells by name. This is useful if users want to select multiple cells.The Select Cell by Name works the same as the Select Net by Name window. Refer to “Net” on page 52 for additional information.



Viewing Selected Nets

In some cases, such as a highly congested area, the number of nets displayed can be limited to only a few nets. Use the following procedure:

1. **Select the net(s) you intend to examine using the Select Net by Name window or by dragging in the Layout area.**
2. **Choose Show Only Selected Nets command from the Options menu in the Viewer, as shown in Figure 3-10**. The Viewer displays only the selected nets in the Layout area until the Show Only Selected Nets command is un-selected.

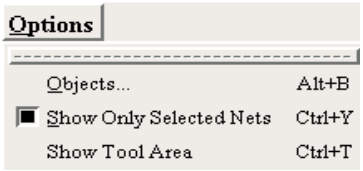


Figure 3-10. Options Window

ID Area

The ID area displays information about an object when it is selected. When multiple objects are selected, object information is not automatically displayed. To show information about all selected objects, choose the Identify command from the View menu.

Enlarging the ID Area

To enlarge the ID Area, click the Enlarge button, which opens a separate ID Area window. ID information can then still be displayed if the Tool area is hidden The information displayed in the ID window window is the same as that in the ID Area.

Highlighting Objects

You can highlight and unhighlight objects using the Highlight and Unhighlight commands in the View menu to identify them. For example, t all objects in a subcircuit can be highlighted. The highlight color currently defined in the object setting is used. Different highlight colors can be used for different subcircuits.

To unhighlight all selected objects:

Choose Unhighlight All command in the View menu.

Customizing Viewer Setup

The design layout in the Viewer window can be customized to assign new colors to objects and to control which objects are displayed using the Objects window. ASICmaster maintains the Viewer's default object setup in the .amrc file, which the Viewer searches for at startup. This secthion contains information about customizing the Viewer setup. The Objects window is shown in Figure 3-11 on page 56.

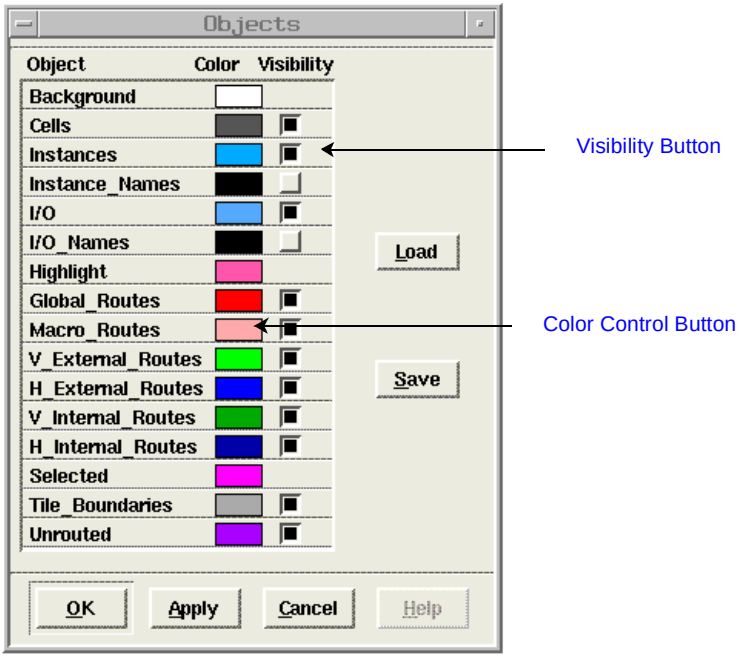


Figure 3-11. The Objects Window

The Objects window displays the visibility status and colors of the different types of objects that appear in the Viewer. The window allows you to modify an object's color and determine an object's visibility in the design layout. You can also save your customized object setup for later use. The Background, Highlight, and Selected objects do not have visibility controls because they are always visible.

The following buttons are located in the Objects window.

Color Control	Displays the color chooser, with which you change an object's color.
Visibility	Determines whether the associated object is visible in the design layout.
OK	Confirms the object setup change process and closes the Objects window.
Apply	Applies an object setup change and leaves the window open.
Cancel	Cancels the object setup change and closes the window.
Load	Displays a file browser to let you select the input file and then loads the current setup from the file you specified.
Save	The first time you click Save, it displays a file browser to let you specify the output file and then saves the current setup to the file you specified. Subsequent saves automatically save to the previously specified filename.
Help	Opens the help system.

To customize the Viewer setup.

1. **Choose the Objects command from the Options menu in the Viewer window.** The Objects window is displayed, as shown in Figure 3-11 on page 56.
2. **Click an object's Color Control button to display the Color Chooser.**
3. **Pull down the Color Method menu to choose a color method.** The methods are HLS, RGB, CMY, or Gray.
4. **Change the object's color by using either the color wheel/plane or the color component values.**
5. **Click OK.**
6. **Set an object's visibility by toggling its Visibility check box.**

7. **Repeat Steps 4 through 6 until all objects are set.**
8. **Click Save.**

Printing the Layout

To print the layout, choose Print from the Layout menu in the Viewer window. The Viewer uses the default printer for all printing. In UNIX, the PRINTER environment variable defines the default printer.

To change the printer setup:

1. **Choose the Printer command from the Layout menu in the Viewer window.** The printer window is displayed.
2. **Specify the name of the printer to use.**
3. **Click Ok.**

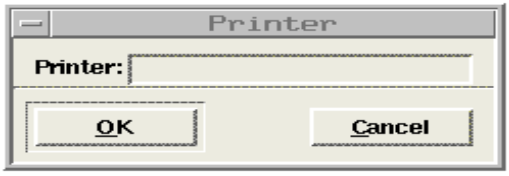


Figure 3-12. Printer Window

Exiting the Viewer

The Viewer window can be exited at any time by choosing Exit from the Layout menu. If the results of a run were save, they can be redisplayed at any time.

Viewer Menu Commands

The following tables list and describe the commands that are available from the Viewer.

The Layout Menu

The following table describes the commands in the Layout menu.

Table 3-1. Viewer Layout Menu Commands

Command	Key Sequence	Function
Update	<Ctrl+l>	Updates the design layout. Useful when viewing the layout while placing/routing.
Print	<Ctrl+p>	Sends the Layout window to a PostScript printer. Uses the default printer (where $PRINTER points to) unless another printer is specified.
Printer	<Alt+p>	Displays the Printer window in which the user can specify a printer for subsequent Print commands.
Exit	<Ctrl+x>	Exits the Viewer.

The View Menu

The following table describes the commands in the View menu.

Table 3-2. Viewer View Menu Commands

Command	Key Sequence	Function
Highlight	<Ctrl+h>	Highlights a group of objects in colors other than the object colors defined in the Objects window. Highlighting and Unhighlighting does not alter the current selection.
Unhighlight	<Ctrl+u>	Removes the highlight mark from the selected objects.
Unhighlight All	<Alt+u>	Removes the highlight mark from all objects.
Identify	<Ctrl+n>	Displays information about the selected objects in the ID area.
Zoom In	<Ctrl+i>	Magnifies the design layout around the center by a factor of 2.
Zoom Out	<Ctrl+o>	Shrinks the design layout around the center by a factor of 2.
Zoom Selected	<Ctrl+s>	Fits all selected objects into the Layout window

Table 3-2. Viewer View Menu Commands (Continued)

Command	Key Sequence	Function
Zoom Last Area	<Ctrl+a>	Magnifies a pre-defined area. You define this area by dragging a rectangle in the Layout window. This command is useful if you have a specific area you view often.
Full View	<Ctrl+f>	Fits the entire design into the Layout window.
Refresh	<Ctrl+r>	Redraws the Layout window.
Select Cell By Name	<Alt+c>	Displays a window in which you can select cells by name. The command is useful if you want to select cells together.
Select Net By Name	<Alt+n>	Displays a window in which you can select nets by name. The command is useful if you want to select nets together, and is also available in the Tool area.
Deselect All	<Ctrl+d>	Deselects all selected objects.

The Options Menu

The following table describes the commands on the Options menu.

Table 3-3. Viewer Options Menu Commands

Command	Key Sequence	Function
Objects	<Alt+b>	Displays the Objects window where you can alter object colors and visibility flags.
Show Only Selected Nets	<Ctrl+y>	Toggle the Show Only Selected Nets flag. When the flag is set, the Viewer only displays the selected nets. The command is useful if you want to view only one or two nets in a highly congested area.

Table 3-3. Viewer Options Menu Commands

Command	Key Sequence	Function
Point Tools		Select the appropriate pointer tool.
Select	<Alt+s>	Changes the current pointer tool to Select, which selects objects in the Layout Window. The Select command is also available in the Tool area.
Zoom Area	<Alt+y>	Changes the current pointer tool to Zoom Area, which highlights and magnifies an area you define by dragging a rectangle in the Layout window. The Zoom Area command is also available in the Tool area.
Hide/Show Tool Area	<Ctrl+t>	Hides or shows the Tool area. If you need a larger Layout Window, choose Hide Tool Area. A Hide button is also available in the Tool area.

4

Using Constraints

This chapter describes the constraints that can be used to guide place and route, and to set optimization criteria. These constraints may be the forward timing SDF file generated by the synthesis tool. They may also be relative to the placement, the global resources and the netlist optimization. In this case, these constraints should be included in a file with the .gcf extension.

The Design Executive is used to import constraint files, which must use the language and syntax described in “Constraint File Syntax” on page 173.

Types of Constraints

Constraints are used to ensure that a design meets timing performance, and required pin assignments. The types of parameters that ASICmaster sets and the types of constraints that can be defined in constraint files include:

- Timing path constraints
- Global resource constraints
- Netlist optimization constraints
- Placement constraints
- JTAG constraints
- I/O constraints

Timing Constraints

Timing constraints are used to ensure that a design meets the required timing performance. Constraints can be entered using an ASICmaster constraints file or using an SDF path contraints file. SDF files can be written by synthesis tools. The two formats cannot be combined in one file. However, SDF files and with ASICmaster constraints files can be used in the same design.

Timing constraints must be reasonable. Over constraining a design may result in an increase in place and route run times, while not improving circuit performance.

The placer considers timing constraints and attempts to meet them. After routing, ASICmaster displays messages to identify the constraints that cannot be met.

Guidelines for Creating Timing Constraints

To understand the complexity of a design and its performance, perform placement and routing with no constraints to see if routing can complete without constraints. If routing completes successfully, create the timing annotation files and backannotate the post-layout delays to see if the physical design meets timing requirements.

If these requirements are not met, you can guide timing driven place and route by forward annotating the SDF generated by the synthesis tool. These nets are identified when analyzing the timing reports.

If you are using a synthesis tool such as Synopsys Design Compiler, Actel recommends that you use it to generate an SDF file containing path constraints.

Constraint File Syntax

The following section describes Notice, that in the case of duplicated constraints, the last one specified for a specific item overwrites any previous similar constraints already specified for the considered item.

Statement

An ASICmaster constraint consists of a statement and an argument, terminated by a semicolon. Statements are not case sensitive. However, cell instance, net, and port names used as arguments may be quoted and are case sensitive. Except for white spaces, all ASCII characters can be used. Comment are allowed in constraints files and must be preceded by two forward slashes (//). Time values are given in nanoseconds.

net_critical_ports

Use this statement to specify a specific subset of critical ports on a net.

For example, the following statement identifies two inputs of the net "/u1/u2/net1" that are more critical than all other connections on that net. All other connections on the net will be buffered to reduce fanout delay on the specified inputs:

```
net_critical_ports /u1/u2/net1 nandbk1.A sigproc.C;
```

set_critical

Use this statement to specify critical nets and their relative criticality over other critical nets.

```
set_critical criticality_number hier_net_name
       [ ,  hier_net_name … ] ;
```

where

The criticality_number is from 2 to 5 (1 being the default criticality for every net).

hier_net_name is the full hierarchical net name.

For example, the statements below sets the timing of "u1/u2/net1" more critical than "u1/u2/net5 and u1/u2/net3":

```
set_critical 2 /u1/u2/net5, u1/u2/net3;
set_critical 5 /u1/u2/net1;
```

set_critical_port

Use this statement to identify device I/O ports that have above-normal criticality. The criticality number scales is the same for the set_critical statement.

```
set_critical_port criticality_number signal_name
      [ ,  signal_name … ] ;
```

where

The signal_name is the name of a user-defined signal associated with a specific I/O pin on the part.

For example, the following statement sets all nets associated with device ports IOBus[3] and IOBus[5] to have criticality 3:

```
set_critical_port 3 IOBus[3], IOBus[5];
```

set_max_path_delay

Use this statement to constrain the maximum delay on paths. The calculate timing task will report a note in the timing log if this delay is not met.

```
set_max_path_delay delay_value
 hier_inst_name  .inst_port_name
[ ,  hier_inst_name  .inst_port_name , … ] ;
```

where

Delay_value - is a floating integer.

Delay values - are in nanoseconds.

Hier_inst_name - is the hierarchical path to a cell instance.

Inst_port_name - is a port name of a cell instance.

For example:

```
set_max_path_delay 12.5 "mult4/mult/nand2_2".Y, "mult4/mult/
nand3_1".A,"mult4/mult/nand3_1".Y,"mult4/mult/nor2_2".A;
```

timing_buffer_fanout

Use this statement to control the buffering for net fanout reduction. The value defines the minimum fanout required for optimization. A value of 0 disables timing buffer insertion.

```
timing_buffer_fanout POSNUMBER;
```

For example, the following statement resets the fanout limit from the default value 8 to the value 12:

```
timing_buffer_fanout 12;
```

Global Resource Constraints

Each ProASIC device includes four global networks that have access to every tile. These four global networks provide high speed, low skew performance to signals such as clocks and global reset.

Once the netlist is imported, ASICmaster sets global resource parameters and promotes the highest fanout nets to the remaining global resources unless the "dont_fix_globals" statement has been specified in a constraint file . To do this, the checker program demotes appropriate global cell instantiations in the design netlist.

Note: When using the dont_fix_globals statement, global assignments made in the constraint files and design netlist will be honored (the constraint file entries will take precedence).

These global resource parameters can be supplemented by including global resource constraints in a constraint file. Global resource constraints can define which signals are assigned to global resources and define which signals cannot be promoted to global resources. Global resource constraints can also override the default action that selects high fanout nets for use by the global resources.

If global resources overrides the default action, assignments that do not include any of the four highest fanout nets will generate a warning.

Priority Order for Global Promotion

While assigning signals to global resources, ASICmaster considers this information in the given priority:

1. Set_global and Set_io statements
2. Nets with the highest potential fanout above 32 (after removal of all buffers and inverters).
3. Global cell instantiation in a netlist

Note: By default, a net of less than 32 fanout will not be promoted to global automatically, unless the set_global or set_io constraint statements is used for this net. Users can override this threshold of 32 by using the set_auto_global_fanout constraint statement.

Promotion Scheme

Figure 4-1 illustrates the promotion scheme for promoting global fanout nets.

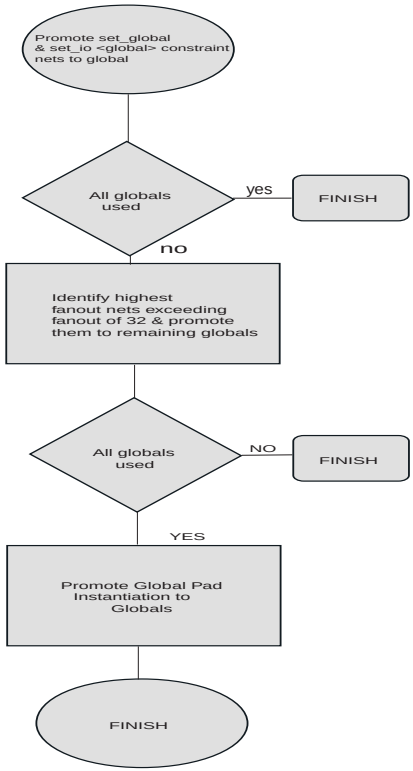


Figure 4-1. Promotion Scheme

Global Resources

The following section describes the global resource constraints that can be set.

set_auto_global

Use this statement to specify the maximum number of global resources to be used. The checker assigns global resources to high fanout signals

automatically. This statement also controls the number of global resources to be used.

If the user specifies a number that exceeds the actual number of global resources available in the device, the checker ignores the statement. If the user specifies 0, no automatic assignment to global resources will take place.

```
set_auto_global number;
```

For example, the following statement specifies that of the possible four global lines available, the checker program can automatically promote two high fanout nets:

```
set_auto_global 2;
```

set_auto_global_fanout

Use this statement to set the minimum fanout a net must have to be considered for automatic promotion to a global. By default this is set to 32.

```
set_auto_global_fanout number;
```

For example, the following statement determines that a net must have at least 12 fanouts before the checker program will consider it for automatic promotion to a global resource:

```
set_auto_global_fanout 12;
```

set_global

Use this statement to classify nets as global nets.

```
{ set_global }  hier_net_name
        [ ,  hier_net_name … ];
```

For example:

```
set_global u1/u3/net_clk, u3/u1/net_7;
```

set_noglobal

Use this statement for classifying nets to avoid automatic promotion to global nets.

```
{ set_noglobal }  hier_net_name
[ ,  hier_net_name … ];
```

For example:

```
set_noglobal u2/u8/net_14;
```

If the net was previously assigned to a global resource, this statement will demote it from the global resource.

dont_fix_globals

Use this statement to turn off the default action that automatically corrects the choice of global assignment to use only the highest fanout nets.

The syntax is:

```
dont_fix_globals;
```

Netlist Optimization Constraints

Netlist optimization attempts to remove all cells from a netlist that have no affect on the functional behavior of the circuit. This reduces the overall size of a design and produces faster place and route times. This optimization is based on constants and inverters pushing and takes advantage of inverted inputs of the basic logic elements. After optimization, a design may fit into a smaller ProASIC device.

Netlist optimization can be controlled by including netlist optimization constraints in constraint files and then importing the files to ASICmaster.

By default all optimizations will be performed on the netlist. To control the amount of optimization that takes place, netlist optimization constraints can be used.

Netlist optimization constraints can turn off all optimizations or disable the default action that allows all optimizations to limit the type of optimizations performed.

The constraints can also define a maximum fanout to be allowed after optimizations are performed and isolate particular instances and hierarchical blocks from the effect of optimization.

After completion of netlist optimization, the design is a functionally identical representation of the design produced internally for use by ASICmaster. However, ASICmaster does not write out a modified netlist. View the design's layout after successful placement and routing. After optimization, a number of instances that do not contribute to the functionality of the design may have been removed. To keep the SDF file consistent with the original input netlist, deleted cells are written with zero delay so that backannotation is performed transparently.

Netlist Optimization Constraint Syntax

The following types of netlist optimizations are available for all netlist optimization constraints:

buffer removes all buffers in the design provided that the maximum fanout is not exceeded.

clocktree removes all back-traceable inverters and buffers that exist in nets connected to clock inputs on all flip-flop cell types.

const replaces all logical elements with one or more inputs connected to a constant (logical “1” or “0”) by the appropriate logic function. If the replacement logic function is identified as an inverter or buffer, that element is removed.

dangling recursively removes all cells driving unconnected nets.

inverter removes all inverters in the design provided that the maximum fanout is not exceeded.

The following sections describe the netlist optimization constraints that are available.

dont_optimize

This statement turns off all netlist optimizations. When followed by one or more of the netlist optimization types, this statement turns off the named optimization(s).

```
dont_optimize [{ inverter buffer clocktree resettree const
dangling}];
```

optimize

This statement turns on all netlist optimizations (the default mode). When followed by one or more of the netlist optimization types, this statement enables only the named optimization(s).

```
optimize [{ inverter buffer clocktree resettree const
dangling} ];
```

set_max_fanout

Use this statement to avoid removing the buffers instantiated during synthesis. ASICmaster does not insert buffers for nets that have a fanout higher than the specified value. It only removes buffers if this statement is not used. For instance, set_max_fanout 8; will remove all buffers and inverters that are driving a load of less than 8.

```
set_max_fanout
```

dont_touch

This statement allows the user to selectively disable optimization of named hierarchical instances. The wildcard "*" can be used to isolate all sub-blocks under the named block.

```
dont_touch hier_net_name
[ ,  hier_net_name … ] ;
```

For example:

```
optimize buffer inverter;

set_max_fanout 16;

dont_touch /U1/myblock/*;
```

The statement in this example will enable only the buffer and inverter optimization types and optimization will be done on all instances except those contained in the block called /U1/myblock.

Placement Contraints

This section describes placement constraints. It is possible to use placement constraints to specify block-instance and macro placement.

Users can specify initial, fixed, region, and macro placements. Also, placement obstructions (locations that are not to be used and thus to be keep empty during placement instances) can be specified.

For example, a constraint that places two connected blocks close together usually improves the timing performance for those blocks. Similarly, a constraint that assigns an I/O pin to a particular net forces the router to make the connection between the driving or receiving cell and the I/O itself.

Like all constraints, placement constraints limit ASICmaster's freedom when processing the design. For instance, assigning a fixed location makes that location unavailable during placement optimization. Such removal usually limits the program's ability to produce a chip-wide solution.

Creating the I/O Allocation Pin Map

If the printed circuit board (PCB) has been designed that will include the ProASIC part, there may be a need to fix the pin assignments for the physical design. The easiest way to do this is to include an ASICmaster I/O pin map in the Design Executive's list of constraint files. The pin map allocates the design's I/O signals to available package pins.

Creating a file of constraints to map I/O signals and package pins can performed two ways:

- Automatically using ASICmaster with the generate pin map task. ASICmaster also provides a pin map editor. If the user generates the map automatically, ASICmaster uses the constraints syntax.

  For example, when used as an input constraint file, a file containing

  ```
  set_io N14 OUT2;
  set_io R15 A;
  ```

  forces signals OUT2 and A to pins N14 and R15 respectively on the selected package.

- Manually using a text editor.

Package Pin and Pad Location

Generally, users will be concerned with the mapping of signals (ports) to the pins of the selected package. However, there may be cases where users wants to control the allocation of signals to particular pads. This is accomplished by assigning ports to the pad location rather than to the package pin. Because all pads are pre-bonded to package pins, the effect is to assign ports to package pins, with the emphasis on pad location rather than package pin.

Pad location is described by the letters N (North), S (South), E (East) or W (West) followed by a space and a number. This location code determines the direction and offset of the pad with respect to the die and is consistent with the position of the I/O pads displayed on the layout viewer. The top edge of the viewer contains the North pads and the right edge contains the East pads. The number refers to the pad position along its edge. For example, N 48 corresponds to the 48th pad along the North edge of the die. Figure 4-2 shows the numbering system used for pad location.

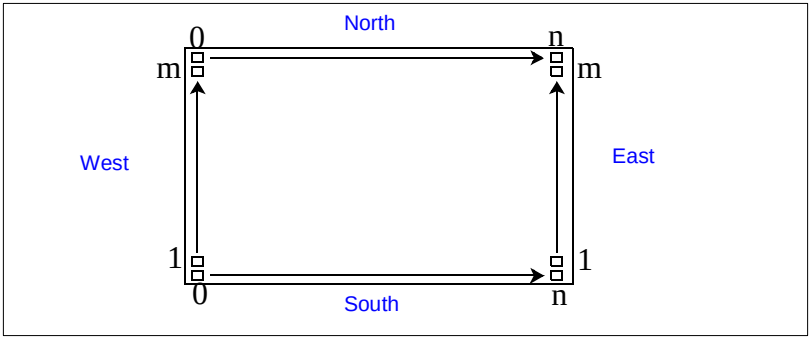


Figure 4-2. Pad Locations

Note: The first pads shown on the layout viewer are pad 0 for the West and East directions. However, they are not physically available.

Individual Placement Constraint Syntax

This section describes the types of individual placement constraints that are available. Individual placement constraints can be useful if users want to force certain critical blocks to be fixed locations, or guide the placement program for the initial location of blocks.

set_empty_location

Use this statement to specify a location in which no cell should be placed.

```
set_empty_location ( x , y );

set_empty_location (x bl, y bl   x tr, y tr) ;
```

where

x , y - (required) are the (x, y) tile coordinates that specify the empty cell location.

x_{bl}, y_{bl} x_{tr}, y_{tr} - (required) are the x, y tile coordinates for the bottom left and top right corner of the region.

For example, the following statement informs the placement program that location (3, 7) is unavailable for cell placement:

```
set_empty_location (3,7);
```

set_empty_io

Use this statement to specify a location in which no I/O pin should be placed. The location can be specified by side and offset or by name.

```
set_empty_io { package_pin | pad location};
```

For example, the following statement forces pin B5 and the pin associated with the fourth tile on the North side to be empty:

```
set_empty_io B5, (N,4);
```

set_initial_io

Use this statement to initially assign package pins to I/O ports or locate I/O ports at a specified side of a device. The placer can reassign or relocate the cells during placement and routing.

```
set_initial_io { package_pin | pad location} io_port_name
       [ ,  io_port_name  , … ] ;
```

where

package_pin - is a package pin number for a specified I/O cell. If you use *package_pin*, only one *io_port_name* argument is allowed. (required if no pin location is given)

pad location - is one of N, S, E, or W, followed by a pad location number on the chip; constrains the pin location of a specified I/O cell to a specific pad location on the chip. Only one *io_port_name* argument is allowed. (required if no package pin or side is given)

io_port_name - (required) is the name of an I/O port to be assigned to a package pin or located at a specified edge of a package.

The following example statement initially places the I/O associated with net in3 to package pin A11:

```
set_initial_io A11 in3;
```

The following example statement initially places the I/O associated with net in4 on the fourth tile on the North side:

```
set_initial_io (N,4) in4;
```

set_initial_location

Use this statement to initially locate a cell instance at specified x,y coordinates. The placer can relocate the cell instance during place and route.

```
set_initial_location ( x, y ) hier_inst_name;
```

where

x , y - (required) are the x, y tile coordinates for the location of a specified cell instance.

hier_inst_name - (required) is the hierarchical path to a cell instance.

For example:

```
set_initial_location (43,105) bk3/fp5/nand3_4;
```

set_io

Use this statement to either assign package pins to I/O ports or locate I/O ports at a specified side or location of a device. The placer cannot reassign or relocate the cells during place and route.

For example:

```
set_io A9 in1;
set_io (S,22) in2;
```

set_location

Use this statement to locate a cell instance at specified x,y coordinates. The placer cannot relocate the cell instance during place and route.

```
set_location ( x, y ) hier_inst_name ;
```

where

x, y - (required) are the x, y tile coordinates for the location of a specified cell.

hier_inst_name - (required) is the hierarchical path to a cell instance.

For example:

```
set_location (1,15) u4/u3/nand3_4;
```

set_initial_location

Use this statement to initially locate a cell instance at specified x, y coordinates. The placer can relocate the cell instance during place and route.

```
set_initial_location ( x, y ) hier_inst_name;
```

where

x , y - (required) are the x, y tile coordinates for the location of a specified cell instance.

hier_inst_name - (required) is the hierarchical path to a cell instance.

For example:

```
set_initial_location (43,105) bk3/fp5/nand3_4;
```

This statement has been extended to allow you to initially place a sub-design instance by calling its macro and then applying a translation and rotation.

```
set_initial_location (x, y) hier_subdesgin_inst_name
         macro_name  [transformations];
```

where

hier_subdesign_inst_name - is the hierarchical name of the instance of the sub-design.

x, y - is the final location of the lower left corner of the macro after all transformations have been completed.

macro_name - is the name of previously defined macro.

transformations - are optional and any of the following in any order:

```
flip lr        flip cell from left to right
flip ud        flip cell from up to down
rotate 90 cw   rotate 90 ° clockwise
rotate 180 cw  rotate 180 ° clockwise
```

```
rotate 270 cw  rotate 270 ° clockwise
rotate 90 ccw  rotate 90 ° counter-clockwise
rotate 180 ccw rotate 180 ° counter-clockwise
rotate 270 ccw rotate 270 ° counter-clockwise
```

The transformations are processed in the order in which they are defined in the statement.

For example:

```
set_initial_location (3,3) a/b mult flip lr;
```

Region Placement Constraint Syntax

This section describes the type of region placement constraints that are available

set_location

Use this statement to locate a complete sub-circuit, identified by a wildcard, within a specified region. The placer will keep all cell instances specified in the wildcard within the coordinates of the placement region. The set_location statement can only be used for cells in the core array. No I/O pad can have a set location. For I/O pads, use the set_io statement.

```
set_location ( x bl, y bl    x tr, y tr )   hier_name_wildcard ;
```

where

x_{bl} , y_{bl} x_{tr}, y_{tr} - (required) are the x, y tile coordinates for the bottom left and top right corner of the region.

hier_name_wildcard - (required) is the hierarchical path wildcard to cell instances. Only simple wild card characters are supported: "?" stands for one character and "*" stands for zero to many characters.

For example:

```
set_location (2,2 20,10) /u5/u4/*;
```

set_empty_location

Users can also specify a region (4,7 20,11) which is unavailable for cell placement:

```
set_empty_location ( x , y );
set_empty_location (x bl, y bl   x tr, y tr) ;
```

where

x , y - (required) are the x, y tile coordinates that specify the empty cell location.

x_{bl}, y_{bl} x_{tr}, y_{tr} - (required) are the x, y tile coordinates for the bottom left and top right corner of the region.

For example:

```
set_empty_location (4,7 20,11);
```

Macro Placement Constraint Syntax

This section describes the types of macro placement constraints that are available.

Macro placement constraints can be useful if users want to reuse a previously produced placement for a subcircuit of a design or if the design is using a predefined core.

```
macro name (x1, y1    x2, y2) {
macro_statements


}
```

where

name - is the macro name identifier

x1, y1- is the lower left coordinate of the macro

x2, y2 - is the upper right coordinate of the macro

Use this statement to define the locations of a sub-design as a macro so that users are able to reuse this placement in different instantiations of the sub-design. The macro is defined in terms of individual core placements. The

hierarchical instance names, appearing in these constraints, are considering the macro as the top of the design.

For example:

```
macro mult (1,1 6,6) {

    }
```

Macro placement and set_location and set_initial_location

These statements have been extended to allow user to place or initially place a sub-design instance by calling its macro and then applying a translation and rotation.

For example:

```
set_location (3,3) a/b mult flip lr;
```

JTAG Constraints

These statements control whether the user intends to perform JTAG operations on the device. This restricts the use of 4 package pins to JTAG functions (TDI, TDO, TMS and TCK). In extended mode an extra package pin gets reserved for a reset operation (TRST). By default these package pins are configured as regular IO signal pins.

Use one of these statements to reserve the JTAG pins.

```
use_JTAG;

use_extended_JTAG;
```

5
Back Annotating Timing Information

This chapter explains how to enable timing extraction, set delay formats and timing values and back annotate delay data into CAE environments. ASICmaster can be used to extract accurate timing data from placed and routed designs. The system writes the extracted data to a timing annotation SDF file. In an ASICmaster-supported CAE environment, this file is used to back annotate delay data to the original design environment and perform post-layout simulations and static timing analysis. ASICmaster can generate post-layout timing delays such as the loaded intrinsic delays of the logic elements used in the design and the delays associated with interconnecting nets between the pins of connected elements.

The SDF-Distributed format uses the Standard Delay Format. This format uses both the IOPATH and PORT statements to back annotate the intrinsic delay and the interconnect delay. For more information about using the SDF format, see the *ProASIC Interface Guide*.

Enable Timing Extraction

This section describes how to use ASICmaster Design Executive to extract accurate timing data and generate a timing annotation file by enabling timing extraction and setting operating conditions.

The following sections describe setting delay-calculation parameters with the Design Executive.

To enable timing extraction

1. **Enable the Timing Annotation check box.** Check the check box in the Timing Annotation tab in the Results area as shown in Figure 5-1 on page 84.
2. **Name the Timing Annotation file.** Enter the name directly in the text field or click the Timing Annotation folder button. A file browser appears. Use it to select an existing file or to enter a new one in the file browser.

ASICmaster executes timing calculation and extraction of the timing data from the design database, if the timing annotation task is enabled.

CAUTION: The system generates timing files (containing the extracted timings) on a per-design basis. All cell instances, net names, and so forth are unique to

the design that ASICmaster is currently processing. The entries in the timing file must match entries in the design's netlist.

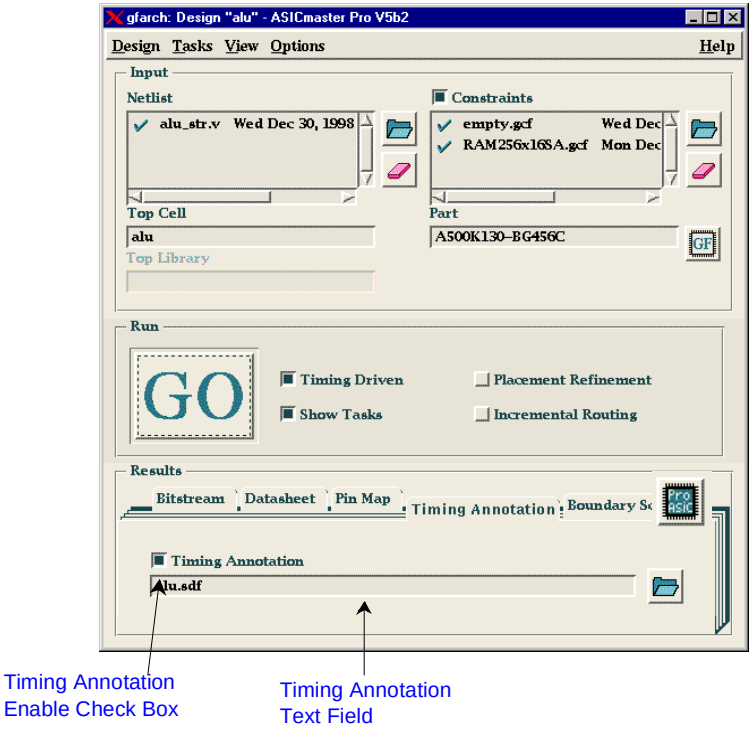


Figure 5-1. Design Executive Window

Generating a Timing File

If Timing Annotation is enabled, a file is automatically generated. Otherwise, select the Generate Timing task in the Tasks selection window (see Figure 5-2). In this case, ASICmaster uses defaults for all delay-calculation parameters and assigns the design name .sdf as a default

SDF back annotation file name. The timing file is created only when the routing program completes successfully.

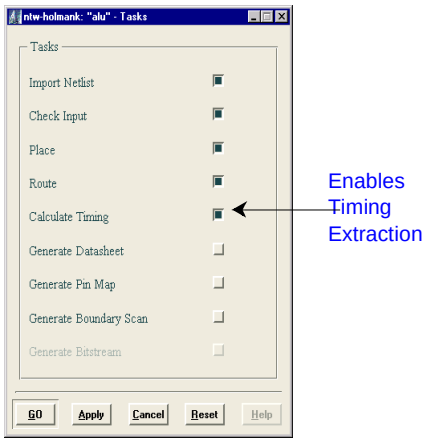


Figure 5-2. Tasks Selection Window

Setting Operating Conditions

ASICmaster uses timing delay extraction to calculate the interconnection delays and the loaded pin-to-pin delays. It uses the intrinsic timing values and other technology-dependent data to determine the post-layout timing delays in a design.

For each extracted timing data point, the calculating program determines three numbers covering a range of values. The range is (min:typ:max) from the minimum expected delay to the maximum expected delay. The default is the typical value. The range of timing delays allows the user to simulate and analyze a design with either typical values or those at the edges of the design space. Designs can be simulated using maximum delay values to verify its operation given the worst-case combination of conditions and timing analysis can be run in min-max mode to determine the timing margin exhibited by a design.

Default Pad Load

By default ASICmaster assumes a pad load of 35 pF for all IO's. The user can overrule this assumption by setting a different default pad load. This load is taken into account when calculating the intrinsic delay for IO's. If the user wants to specify specific pad loads for specific IO's, the best strategy is to set the default pad load to 0 pF and to specify these loads in the synthesis environment. The synthesis libraries have drive impedances specified for every IO primitive.

Back Annotating Post-Layout Delays in the CAE Environment

After successful place and route tasks, the exact timing of the cells and their interconnections are available. Actel recommends using them to perform simulation or timing analysis to determine the maximum speed at which the design will run, based on the selected ProASIC device. It is also possible to use these operations to verify that the functionality of the design has not been affected by the place and route operations or by any other ASICmaster process.

After creating the timing file, the user can reference it (back annotate the timings) to a supported CAE environment. ASICmaster currently supports timing back annotation for PrimeTime, Synopsys Design Compiler, Verilog, and VHDL systems. Refer to the *ProASIC Interface Guide* for more information.

6

Using ASICmaster in TTY and Batch Modes

This chapter describes using ASICmaster in TTY (interactive command-line) mode[1]and in batch mode. Users can use TTY or batch mode to automatically repeat ASICmaster design cycles or to speed up work. In batch mode, ASICmaster responds to requirements not met by exiting the script and returning to either the ASICmaster prompt or the UNIX prompt.

In TTY and batch modes, the set of available commands is limited and does not support commands that involve a GUI. Therefore, View commands are not available. However, output files can be viewed with a text editor.

To view TTY or batch layout results, enter the graphics command on the command line or start the ASICmaster graphic user interface and use the Viewer in the Design Executive.

Command Line Options

This section describe the command-line options for using ASICmaster in TTY and Batch mode. Table 6-1.describes command-line options.

Table 6-1. ASICmaster Command-line Options

Command-line Option	Function
-batch *batch_file*	Runs the commands contained in the batch file specified. If ASIC-master encounters an exit command, it exits batch mode immedi-ately.
-go	Starts ASICmaster as if there were a script with the initialize and go commands. Runs as a batch script and directs all messages to the screen. Does *not* require a command file.
-graphics	Enables graphic display. When used in conjunction with the -batch or -go option, it allows batch execution to display ASICmaster graphics.

1. Available on UNIX only.

Table 6-1. ASICmaster Command-line Options (Continued)

Command-line Option	Function
-new	Creates a new design project database.
-tty	The default when a batch file is specified. Suppresses all graphics and prompts the user to enter batch commands on the command line.

Using TTY Mode[1]

To start ASICmaster in TTY mode, enter the following command:

```
asicmaster design_name [-new][-tty|-graphics] [-batch
batch_file |-go]
```

The command must include a design name or the -new option. If the name of the design entered does not exist, and the -new option is not used, ASICmaster displays messages to alert the user.

Once initialized ASICmaster displays the prompt, indicating that it is ready to accept interactive commands from the command line. Enter any of the TTY commands described in "TTY and Batch Commands" on page 93.

Using Batch Mode

ASICmaster can be started in batch mode from the command line or in the Design Executive. When ASICmaster starts in batch mode, it immediately runs the batch commands in the file specified. If the file includes a command that ASICmaster does not support in batch mode or cannot successfully execute, the system displays an error message and terminates the batch script.

1. Available on UNIX only.

Starting Batch Mode from the Command Line

To start ASICmaster from the command-line enter the following command:

```
asicmaster design_name -batch script_file [-graphics |-new]
```

Include the "-graphics" option for a graphical display of the results. Include the "-new" option to create a new design for the batch file to process.

Starting Batch Mode from Design Executive

Use the following procedure to start ASICmaster in batch mode from the Design Executive Window.

1. **Select Run Script from the Tasks menu.** The Script File Selection browser appears.
2. **Specify the script file to use in the batch session.** Script files use the .scr extension ASICmaster immediately reads in the file and runs the commands it contains.

Creating a Batch Script File

Batch script files can be created using the batch language in any editor to create a new script or by capturing ASICmaster activities and converting them to a correctly formatted batch script file using the Command Report in Design Executive

Note: For convenience and accuracy, Actel recommends using ASICmaster to create batch scripts, as described in the following section.

Creating a Batch Script File Using ASICmaster

Use the following procedure to create a batch script file the Design Executive.

1. **Invoke ASICmaster.**
2. **Use the Design Executive to enter commands, specify inputs and outputs, and enable tasks.**

3. **Enter the commands and run the tasks that you want the batch script to contain.**
4. **From the Design Executive's View menu, click Command Log.** The Command Log Window appears as shown in Figure 6-1 on page 91.
5. **Click Save or Save As.** The Script File Selection browser appears.
6. **In the browser, specify a filename for the batch file.** Append the .scr or .log extension to the filename.

Note: The first time the Save command is used to save a design project's command log, a filename must be specified. Subsequent Save executions use the last name specified until the user changes it.

7. **Click OK at the bottom of the Script File Selection browser to save the Command Log data to the file specified.**

This action caused ASICmaster to add the new netlist to the Design Executive's Netlist list and empty the Design Executive's Constraints file list.

The Command report window contains a history of all of the run's operations that were activated by commands supported in batch mode. Each time ASICmaster executes a command it tracks it in the Command Log Window.

The command history illustrated in Figure 6-1 shows that a new netlist file named alu.str.v for a new design was specified.

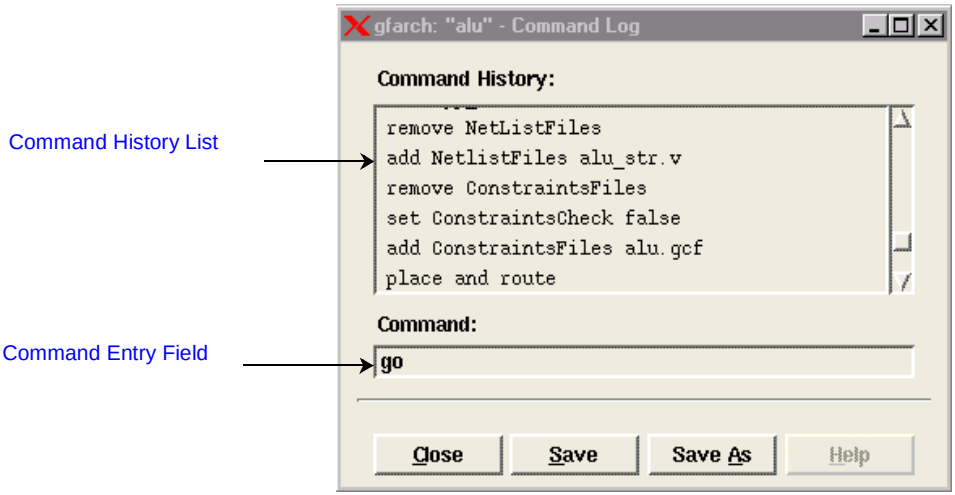


Figure 6-1. Command Log Window

Creating a Batch Script File with a Text Editor

A text editor can be used to manually create or edit a batch script file using the commands described in "TTY and Batch Commands" on page 93. Figure 6-2 shows an example batch script file.

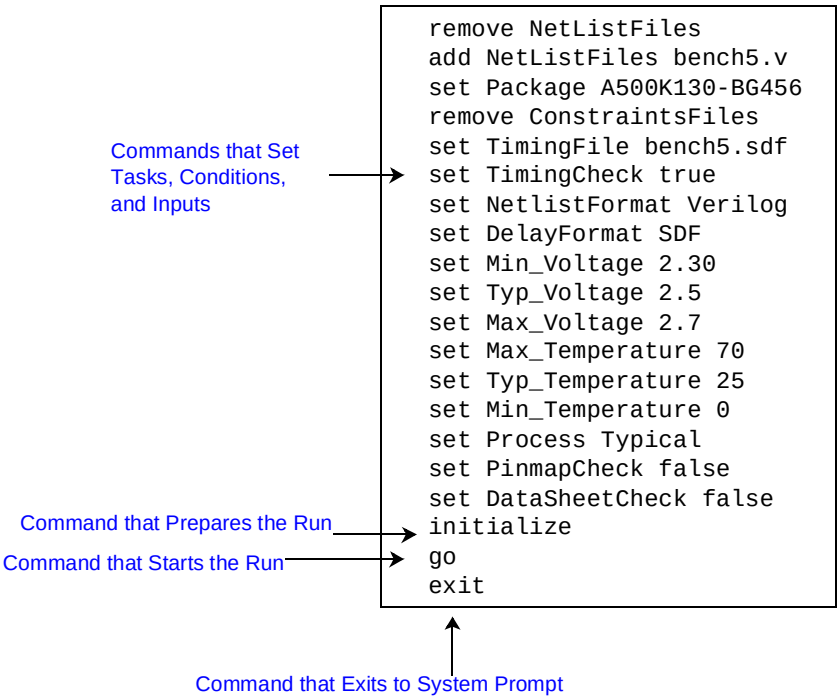


Figure 6-2. Batch Script Sample

This script manually imports the netlist "bench5.v," sets the targeted package, and design conditions. It also specifies the timing SDF file, commands the program to run the design and after this process to quit.

TTY and Batch Commands

Commands in a batch file are used to specify functions to be performed. The function should be performed in the following order:

1. Specify the tasks ASICmaster is to perform.
2. Specify the inputs to ASICmaster, and their formats and other characteristics.
3. Specify the design's target device.
4. Specify the operating conditions for timing calculations.
5. Specify the place and route conditions.
6. Specify ASICmaster outputs and their formats.
7. Execute the task run.

In addition, general operations such as displaying design-state parameter values and changing from TTY mode to graphics mode can be performed.

The following sections describe the different types of commands that can be used and presents the individual commands.

Specifying Tasks

There are two types of commands that can be performed, flow-limiting, and flow-redefinition commands. Flow-limiting commands define how far ASICmaster goes in executing the standard task flow by defining the last task in the sequence ASICmaster performs, limiting ASICmaster to performing all other tasks required to successfully perform the last task for the current design state, and immediately executing tasks.

When performing a limited task sequence, ASICmaster tracks the current design state and its requirements. ASICmaster either automatically performs tasks or in TTY mode it prompts the user to perform tasks that the current design state requires. In batch mode, ASICmaster exits the script with an error message so that the user can perform the tasks.

For information about design-state parameters, see "The Design (*.dtf) Directory" on page 137.

Using the TTY and batch flow-limiting commands is equivalent to using the Design Executive's Task menu. The Tasks menu commands include Check Input. The flow-limiting commands include Check.

Using either command causes ASICmaster to perform all required actions to check the design's netlist and other inputs or prevents the system from performing any tasks that come later in the design flow (for example, placement and routing).

Flow-Limiting Commands

This following section describes each flow-limiting command.

initialize

This command defines the task flow as the unlimited task flow. ASICmaster selects all tasks that are compatible with the design state and executes them. The function of the initialize command is similar to that of the reset command except that the reset command retains any task limit set. The initialize command drops the limit and defines the task flow as all tasks that are consistent with the current design state.

Example Initialize Command

```
initialize

// Has the same effect as pushing Go in ASICmaster.
// This calculates which tasks need to be run
// and updates the Tasks list.

remove ConstraintsFiles

// Removes all the constraint files.

add ConstraintsFiles no_optimize_constraints

// Adds a constraint file with the name
// no_optimize_constraints, which means that the
// checker must be run to read in these constraints.


place and route

// limits the flow to finishing place and route.
```

```
reset

// After making any change, you must use reset to
// recalculate the tasks to be run. This command does
// not change the task flow limit.If we would use the
// initialize command instead, it would reevaluate
// all tasks that need to be run.

set RouteTask false // on or off after reset
set CheckTask true // and before go.

go  // starts the run.
```

Import Netlist

This command limits the task flow to importing an up-to-date netlist and performs all required tasks.

Check

This command limits the task flow to performing all tasks required for checking the netlist, constraints, and other input, and performs all required tasks.

Place and Route

This command limits the task flow to performing all tasks required to produce an up-to-date physical layout, and performs all required tasks.

Timing

This command limits the task flow to performing all tasks required to provide up-to-date post-layout timing data, and performs all required tasks.

Bitstream

This command limits the task flow to performing all tasks required to provide an up-to-date bitstream device-programming file, and performs all required tasks.

Datasheet

This command limits the task flow to performing all tasks required to provide an up-to-date datasheet, and performs all required tasks.

Boundaryscan

This command limits the task flow to performing all tasks required to provide an up-to-date boundary scan description file, and performs all required tasks.

Pinmap

This command limits the task flow to performing all tasks required to provide an up-to-date pin map, and performs all required tasks.

Flow-Redefinition Commands

It is possible to overrule ASICmaster's default task flow for the current design state to force the system to perform or omit specific tasks. Use the flow-redefinition commands to allow ASICmaster to perform a set of tasks that would, by default, be incompatible with the current design state or reset the task set to conform to ASICmaster's default task flow and the current design state.

Using the TTY and batch flow-redefinition commands is equivalent to using the Tasks selection list. For information about the Tasks selection list, see "Using ASICmaster" on page 9.

The sections that follow describe each flow-redefinition command. Most sections include a table of command parameters.

set task

Each flow-redefinition command (except the reset command) is a version of the set command.

```
set task { true | false }
```

Table 6-2 defines the flow-redefinition command's task parameters and values.

Table 6-2. Set Task Parameters

Parameter: *task*	Value Type[1]	Value Definition
NetlistTask	Boolean	Overrules the default for the import netlist task
CheckTask	Boolean	Overrules the default for the check input task

Table 6-2. Set Task Parameters (Continued)

Parameter: *task*	Value Type[1]	Value Definition
PlaceTask	Boolean	Overrules the default for the place task
RouteTask	Boolean	Overrules the default for the route task
TimingTask	Boolean	Overrules the default for the generate timing annotation task
DatasheetTask	Boolean	Overrules the default for the generate datasheet task
BoundaryscanTask	Boolean	Overrules the default for the generate boundary scan task
PinmapTask	Boolean	Overrules the default for the generate pin map task
BitstreamTask	Boolean	Overrules the default for the generate device-programming bitstream task

reset

The reset command is used to return to an earlier design state after specifying flow-redefinition set task commands. The initialize command can also be used for this purpose as described in “Flow-Limiting Commands” on page 94.

If the set command has been used to make any changes, the reset command must be used to re-calibrate the tasks that are to be run.

The reset command can be used to reinstate the default ASICmaster task requirements and retain the current task-flow limit.

The reset command retains any task-flow limits the user has set, but the initialize command drops the limit and defines the task flow as all tasks that are consistent with the current design state.

Specifying Netlist and Constraint Files

Use the remove and add commands to set up appropriately ordered lists of netlist and constraint files.

remove and add

The remove command deletes all filenames from the netlist or constraints input-file list. Typically, use remove before adding netlists or constraints to ensure that the list was empty.

```
remove file_type
```

For example:

```
remove ConstraintsFiles

remove NetListFiles
```

The add command appends filenames to the netlist or constraints input-file list. Typically, use the add command after the remove command that cleared the list.

```
add file_type file
```

For example**:**

```
add ConstraintsFiles mydesign.sdf
```

Table 6-3. NetListFiles and ConstraintsFiles Parameters

Parameter: *file_type*	**Parameter: *file***
NetListFiles	Single filename
ConstraintsFiles	Single filename

A separate line is needed to add multiple files

. For example:

```
add ConstraintsFiles mydesign1.sdf
add ConstraintsFiles mydesign2.sdf
```

set ConstraintsCheck

This command controls whether ASICmaster will take the constraints files into account or not.

```
set ConstraintsCheck { TRUE | FALSE }
```

For example:

```
set ConstraintsCheck TRUE
```

Specify Netlist File Format

set NetlistFormat

This command sets the default filter for the Netlist Selection file browser.

```
set NetlistFormat netlist_format
```

For example:

```
set NetlistFormat Verilog
```

Table 6-4. The NetlistFormat Parameter

Parameter	Value Type: *netlist_format*	Valid Values
NetlistFormat	String	Verilog, VHDL, EDIF

For more information about specifying the netlist format, see “Specifying Netlist Formats” on page 21.

Specify Netlist Top Cell and Top Library

set TopCell and set TopLib

These set command parameters specify the netlist top cell and top library.

set { TopCell | TopLib } *name*

For example:

```
set TopCell alu
```

Table 6-5. Netlist TopCell and TopLib Parameters

Parameter	Value Type: *name*	Valid Value
TopCell	String	Name of the selected top cell in a netlist
TopLib	String	Name of the library of the top cell (Applicable to EDIF and VHDL)

set Package

The set command Package parameter specifies the design's device.

```
set Package name
```

For example:

```
set Package A500K130-BG456
```

Table 6-6. Package and Part Parameters

Parameter	Value Type	Valid Value
Package	String	The ProASIC device name

Set Commands for Placement Refinement and Incremental Routing

The set command parameters listed in Table 6-7 specify the place and route conditions.

```
set PlacementRefinement true
```

Table 6-7. Place and Route Parameters

Parameter	Value Type	Valid Value
TimingDriven	Boolean	True |False
PlacementRefinement	Boolean	True | False
IncrementalRouting	Boolean	True | False

Set Commands for Specifying Output Filenames

The set command parameters defined in Table 6-8 on page 101

```
set { BitstreamFile | BoundaryscanFile | DatasheetFile | Pin-
mapFile | TimingFile } filename
```

For example:

set BitstreamFile alu.bit

Table 6-8. Output Filename Parameters

Parameter: *output_file_type*	Value Type	Valid Value: *filename*
BitstreamFile	Filename	The filename for the bitstream device-programming file
DatasheetFile	Filename	The filename for the datasheet
BoundaryscanFile	Filename	The filename for the boundary scan
PinmapFile	Filename	The filename for the pin map
TimingFile	Filename	The filename for the timing annotation data

Set Commands for Temperature, Voltage, and Process

The set command parameters defined in Figure 6-9 specify anticipated device operating conditions, which ASICmaster uses to calculate post-layout delays.

```
set { { Min_ | Max_ | Typ_ }Temperature | {  Min_ | Max_ | Typ_
}Voltage | {  Min_ | Max_ | Typ_ }Process } } value
```

For example:

```
set Min_Temperature -40

set Max_Voltage 2.65

set Minprocess Best

set Report Fastest
```

Table 6-9. Operating Condition Parameters

Parameter: *condition*	**Value Type**	**Valid Value**
Min_Temperature	Number	Minimum temperature used in timing calculations
Min_Voltage	Number	Minimum voltage used in timing calculations
Max_Temperature	Number	Maximum temperature used in timing calculations
Max_Voltage	Number	Maximum voltage used in timing calculations
Typ_Temperature	Number	Typical temperature used in timing calculations
Typ_Voltage	Number	Typical voltage used in timing calculations
Default_Pad_Load	Number	Default Pad Load in pF, used in timing calculations
Min_Process	String	One of the following: Best |Typical|Worst
Typ_Process	String	Used fot the typical value of the triplet
Max_Process	String	Used for the max value of the SDF triplet
Report	String	One of the following: Fastest |Typical | Slowest

Set Commands for Output Files

The set command parameters defined in Figure 6-10 on page 103 generate output files.

```
set { BitstreamCheck | PinmapCheck | DataSheetCheck | Bound-
aryScanCheck | TimingCheck } { TRUE | FALSE }
```

For example:

```
set TimingCheck TRUE
```

Table 6-10. Set Check Parameters

Parameter: check	Value Type[1]	Value Definition
BitstreamCheck	Boolean	Overrules the default for the task that generates the bitstream device-programming file
PinmapCheck	Boolean	Overrules the default for the task that generates the pin map file
BoundaryScanCheck	Boolean	Overrules the default for the task that generates a boundary scan description file.
DataSheetCheck	Boolean	Overrules the default for the task that generates a datasheet
TimingCheck	Boolean	Overrules the default for the task that generates a timing annotation file

1: Boolean values are true or false.

show

The show command allows the user to view design state parameter values in TTY mode. If the command parameter is not specified, all parameter values are listed in Figure 6-11 on page 105.

```
show parameter
```

help

This command, typically used only in TTY mode, displays the syntax of all commands and parameters available in TTY mode.

```
help
```

graphics

This command is available only in TTY mode, and changes the ASICmaster operating mode from TTY to the graphic mode.

```
graphics
```

go

In TTY or batch mode, the go command executes either the tasks selected with the flow-limiting and flow-redefinition commands, or ASICmaster's default task flow and uses the data specified with the other general function commands previously described. The go command is the only batch command that executes tasks. All other commands set up the system for execution.

Also, the go command updates place and route results only if needed and perform additional tasks only if the tasks create data that is needed to produce an up-to-date layout.

The following command sequence forces ASICmaster to run import netlist, check the input, and place and route.

```
place and route
set NetlistTask true
set RouteTask true
go
```

exit and quit

These commands are used to exit ASICmaster. They also automatically terminate an ASICmaster batch mode session.

```
exit
```

The Set Command

Table 6-11 presents the set command in reference format.

Table 6-11. Set Operations

Parameter: *task*	**Value Type[1]**	**Value Definition**
Set Task		
BitstreamTask	Boolean	Overrules the default for the generate device-programming bitstream task
BoundaryscanTask	Boolean	Overrules the default for the generate boundary-scantask
CheckTask	Boolean	Overrules the default for the check design task
DatasheetTask	Boolean	Overrules the default for the generate datasheet task
NetlistTask	Boolean	Overrules the default for the import netlist task
PinmapTask	Boolean	Overrules the default for the generate pin map task
PlaceTask	Boolean	Overrules the default for the place task
RouteTask	Boolean	Overrules the default for the route task
TimingTask	Boolean	Overrules the default for the generate timing annotation task
Set Check		
BitstreamCheck	Boolean	Overrules the default for the task that generates the bitstream device-programming file
BoundaryscanCheck	Boolean	Overrules the default for the task that generates the boundary scan description file
PinmapCheck	Boolean	Overrules the default for the task that generates the pinmap file

Table 6-11. Set Operations (Continued)

Parameter: *task*	Value Type[1]	Value Definition
DataSheetCheck	Boolean	Overrules the default for the task that generates a datasheet
TimingCheck	Boolean	Overrules the default for the task that generates a timing annotation file
Netlist		
NetlistFormat	String	Verilog VHDL EDIF
TopCell	String	Name of the selected top cell in the netlist
TopLib	String	Name of the Library of the top cell (applicable in EDIF)
Constraints		
ConstraintsCheck	Boolean	Define whether Constraints files should be taken into account
File Names		
BitstreamFile	Filename	The filename for the bitstream device-programming file
BoundaryScanFile	Filename	The filename for the boundary scan
DataSheetFile	Filename	The filename for the datasheet
PinmapFile	Filename	The filename for the pin map
TimingFile	Filename	The filename for the timing annotation
Physical Parameters		
IncrementalRouting	Boolean	Define whether the router will pickup the routes it generated last time

Table 6-11. Set Operations (Continued)

Parameter: *task*	**Value Type[1]**	**Value Definition**
Package	String	The ProASIC device name
Timing Driven	Boolean	Define whether timing driven place and route will be activated
PlacementRefinement	Boolean	Define whether the placer will run refinement schedules
Operating Condition Values		
Default_Pad_Load	Number	Default pad load in pF, used in timing calculations
Min_Process	String	One of the following: Best Typical Worst
Min_Temperature	Number	Minimum temperature used in timing calculations
Min_Voltage	Number	Minimum voltage used in timing calculations
Max_Process	String	One of the following: Best Typical Worst
Max_Temperature	Number	Maximum temperature used in timing calculations
Max_Voltage	Number	Maximum voltage used in timing calculations
Typ_Process	String	One of the following: Best Typical Worst
Typ_Temperature	Number	Typical temperature used in timing calculations
Typ_Voltage	Number	Typical voltage used in timing calculations
Report	String	One of the following: Fastest Typical Slowest

A

Troubleshooting

This chapter explains how to correct netlist and design problems, place and route problems, back annotation problems, and how to get additional help. During iterative design work, users optimize and troubleshoot designs and device-programming processes. ASICmaster automatically avoids a number of potential problems by alerting users early in an ASICmaster task flow to netlist problems, and to gate and I/O incompatibilities between the design and the device selected.

ASICmaster also identifies any semantic, syntactic, and electrical rule violations, and determines for each task whether all required inputs are available and current. Otherwise, ASICmaster automatically generates the required inputs or prompts the user to generate them.

When the system alerts the user of problems, examine the relevant ASICmaster log files and apply the appropriate changes.

General Optimization and Troubleshooting Guidelines

Before addressing the current situation, be sure that all preceding ASICmaster tasks have completed successfully. Examine all report files to understand warnings and problems reported by ASICmaster.

When addressing a given problem, make the least changes possible. For example, if the place and route tasks do not give the expected results, make sure that the constraints file is correct. Next, check for the input checker in the log files to see whether ASICmaster correctly interpreted the constraints file or rejected them.

Correcting Netlist and Design Problems

ASICmaster automatically avoids a number of potential netlist and design problems by identifying and alerting users of electrical rule violations, as well as gate and I/O incompatibilities between the design and the device selected.

For each syntax error, ASICmaster immediately displays error messages the provide the following information: The file name, line number, problem item and a description of the error and the required correction.

Netlist Syntax Problems

When a netlist is changed in an editor, syntax errors can occur. Although ASICmaster parsers have been designed to find these errors, Actel recommends that users submit the netlist to a simulation tool after manual changes have been made to it to functionally verify the design.

Netlist Semantic Problems

When manually editing netlists or merging multiple netlist files, semantic errors might be introduced. For example, if a pin is referenced on a cell by a cell-instance call to that cell but the pin does not exist on that cell, a semantic error results. The information is syntactically correct, but not all references can be resolved.

Language Subset Violations

Language subset violations are most commonly caused by one of the following:

- Submitting RTL code lines that would have been used for synthesizing the design.
- Submitting a netlist in which all leaf cells are not ASICmaster primitives (for a complete list, see the *ProASIC Macro Library Guide*).
- Using synthesis compilation results in which mapping of ASICmaster target library was not completely successful (unmapped modules were left in the netlist).
- Importing an EDIF netlist that was generated with incorrect parameter settings for producing an EDIF 2.0.0, level 0, Keyword Level 0 netlist.

Electrical Design Rule Problems

ASICmaster imposes the following electrical design rules on designs and alerts users if any are violated:

- Each net must have only one driver.
- Each input port of an ASICmaster primitive must be connected.

Unused input ports must be tied to Power or Ground, depending on the logic of the remaining function. In Verilog, these signals are declared as Supply0 for ground and Supply1 for power. In EDIF, a property is added to the net with the values GND for ground and

POWER for power. ASICmaster primitives can also be used, PWR for power and GND for ground. Simply connect one of them to your net.

Each bidirectional port of an ASICmaster primitive must be connected. Unused bidirectional ports must be tied to Power or Ground, depending on the logic of the remaining function. Each external signal must be connected to a ASICmaster I/O primitive.

Device-Related Checks

After the netlist is imported, the Netlist log includes values for the total number of I/O and core cells. Use this information to select the appropriate device for the design.

During the input check, ASICmaster attempts to fit a design onto the selected part. If this attempt fails, ASICmaster immediately alerts the user of the situation. To correct this, select a larger device.

Constraint-Related Checks

ASICmaster checks constraints for compatibility with the design and the selected device. Criteria include the following:

- Names of nets and pins must match existing nets and pins in the netlist.
- Net criticalities must be within the specified range: 1 to 5 (highest).
- Locations must exist.
- Package pins must exist on the package and be accessible.

ASICmaster also checks the syntax and semantics of SDF files submitted to the tool. Errors in these files should not occur, because they are automatically generated by a synthesis tool.

Note: Such errors might occur because SDF file has been edited manually.

Common Problems and Solutions

This section provides information about common problems that users may encounter while using ASICmaster and the solutions to those problems.

(no_driver)

Problem: [no_driver]: The following nets have no driver. (Place & Route cannot continue until this situation has been corrected.)

net5677

Solution: Make sure that all inputs to all cells are driven. If there are unconnected inputs connect them to PWR or GND. (Be careful not to change the functionality of the design)

(multiple_drivers)

Problem: [multiple_drivers]: The following nets have multiple drivers. (Place & Route cannot continue until this situation has been corrected.)

net5691

Solution: Make sure that the netlist does not contain any wire-and or wire-or constructs. All nets are allowed to be driven by just one driver.

(no_io_defined)

Problem: [no_io_defined] The following external ports connect directly to core cells. No I/O pad has been defined on these external signals. (Place & Route cannot continue until this situation has been corrected.)

data[3]

Solution: Make sure all ports on the top level of your design are connected to PAD cells. This is accomplished by using the "insert_pads" command in the Synopsys Design Compiler or by using the "set chip TRUE" command with Exemplar Spectrum.

Error (#373)

Error [#373]: Unit "IEEE.std_logic_arith" is not in library "IEEE". Try analyzing the unit again.

Problem: [parser_problem]: Error in VHDL source files. Please fix this problem, since the parser is unable to safely process these data.

Solution: The netlist references to other Library Packages than IEEE std_logic_1164.all or A500K.all. This is often the case if a RTL level HDL file is submitted.

Error (#373)

Error [#373]: Unit “WORK.tutorial_pkg” is not in library “WORK”. Try analyzing the unit again.

Problem: [parser_problem]: Error in VHDL source files. Please fix this problem, since the parser is unable to safely process these data.

Solution: The netlist references a file in the WORK directory, which can not be found. Add this file to the input file list to ASICmaster.

Error (#442)

Error [#442]: /export/home/Projects_Stefan/alu/src/aul_trouble_vhd: Line 70, Failed to resolve expression and procedure call /export/home/Projects_Stefan/alu/src/alu_trouble.vhd: Line 70, Failed to resolve expression/procedure call.

Problem: [parser_problem]: Error in VHDL source files. Please fix this problem, since the parser is unable to safely process these data.

Solution: The netlist task has found an undefined function call. This is very likely if a RTL level HDL file is submitted.

(invalid_primitives)

Problem: [invalid_primitives]: The following leaf cells are not ASICmaster primitives. Please make sure you are using the ASICmaster design kit, in building your design.

Solution: The file submitted contains leaf cells which are not part of the ProASIC A500K Family leaf cell set. This problem occurs if the synthesis did not work on all parts of the submitted design or if more than one target library was used during synthesis.

(parser_problem)

Problem: [parser_problem]: In VHDL a top cell needs to be specified. Please fix this problem, since the parser is unable to safely process these data.

Solution: A VHDL or EDIF netlist format is used. Specify the top level name in the appropriate box of the ASICmaster GUI and re-run the task.

(mc_no_macro_loc)

Problem: [mc_no_macro_loc]: System: Memory “ram_inst/M0/TILE0” of type “RAM256x9SA/TILE0” has no location. You should submit in the constraint file a location for this memory. Please consult the release notes.

Solution: The design uses embedded memory blocks. Submit the automatically generated constraints file from MEMORYmaster to the tool. This will enable the automatic placement of the memory blocks. If you want to specify concrete locations use the set_location constraint.

(mc_no_macro_pos)

Problem: [mc_no_macro_pos]: Memory “ram_inst/M0/TILE0” of type “RAM256x9SA/TILE0” has wrong location (1,1). Allowed locations are: (1,57) (17,57) (33,57) (49,57) (65,57) (81,57).

Solution: The locations for the memory block placed by hand are not valid. Use the locations listed in this problem message in the set_location statement.

(Invalid_region)

Problem: [Invalid_region]: Line 1, file /export/home/Projects_Stefan/alu/dtf/gcf/set_loc.gcf: Invalid region (1, 45 32, 58) set for hierarchy “*”. Limits are (1-96, 1-56).

Solution: The region specified falls outside of the chip. Specify a region that falls within the limits of the chip, as specified in the Problem message.

Common Simulation Problems and Solutions

This section provides information about common simulation problems and the solutions to those problems.

Cannot Open File

Problem: # ** Error: Cannot open file ./meminit.dat: No such file or directory

Solution: Copy the meminit.dat file from the <am_install_dir>/etc/deskits/vhdl/lib into the directory the simulator is started from.

WARNING (1)

Problem: WARNING [1]: ../src/RAM256x16SSA.vhd(38): No default binding for component: "ram256x9sa". (No entity named "ram256x9sa" was found)

Solution: Make sure all entities using A500K library primitives have the following line included in their library setup:

```
Library A500K;
Use A500K.all;
```

Correcting Place and Route Problems

The netlist and check tasks in ASICmaster are designed in such a way that no errors should occur when the place and route task is run. Normally, place and route fails when a design is over constrained. To solve this problem, run design with relaxed constraints or perform incremental routing.

System errors should not occur during normal operations and usually indicate that ASICmaster is not correctly installed or one or more files associated with ASICmaster has been manually modified. To correct this, reinstall the software or the internal design-associated manually modified file(s). Then reimport the netlist and rerun all tasks.

WARNING: Actel does not guarantee the results of using the tools in this manner. If problems persist, please refer to “Product Support” on page 223 for information getting technical support. Before placing the call have the system available for execution and be prepared to provide the exact text of the error message received.

Common Error Messages

This section describes common error messages and how correct them.

AMHOME not set

Indicates that the AMHOME environment variable is not set to point to the ASICmaster installation directory.

Set the variable to the appropriate value and rerun the task cycle. For information about AMHOME, see the *ASICmaster Installation and Licensing Guide*.

out of memory

Indicates that the placer's memory requirements exceed the virtual memory available on your system.

Make additional memory available by increasing the amount of swap space on the machine or by eliminating other processes that are using virtual memory.

invalid blocks or nets file

Indicates the usage of a version of ASICmaster that is newer than the version used for the most recent design run.

Reimport the netlist and rerun all other tasks.

Back Annotation Problems

This section describes back annotation problems and proposes solutions for them. For more information about back annotating post-layout delays, see "Back Annotating Timing Information" on page 83.

Simulation Problems or Timing Analysis after Back Annotation

The most common problem is that the SDF file does not match with the design being simulated or analyzed for timing. Check whether the SDF file is overlaid with the netlist it generated for and whether it is applied to the right hierarchy of the simulation files.

B

File and Command Selection

This chapter explains how to use file browsers and keyboard accelerators for traversing directories, viewing and selecting files and entering commands. It also includes sections on using file extensions and tear-off menus for keeping frequently used menus displayed on the screen.

Using File Extensions

The names of all ASICmaster design directories have the .dtf extension. ASICmaster expects other types of files to have the default extensions listed in Table B-1. When creating files for ASICmaster, use these default extensions whenever possible.

Table B-1. ASICmaster File Extensions

Extension	File Type
.bit	Bitstream device-programming file
.bsd	Boundary scan file in BSDL format
.dtf	ASICmaster design directory (always enforced)
.edif	EDIF netlist
.gcf	constraints or pin maps in ASICmaster's format
.log	Task log, session log, command log
.scr	Batch script
.sdf	Timing annotation data or constraints in SDF format
.txt	Datasheet
.v	Verilog netlist
.vhd	VHDL netlist

The design directory extension (.dtf) is always enforced; A directory without the .dtf extension cannot be recognized as the design directory.

When files are generated, the default extensions are applied automatically, unless a filename is specified with a different extension. For example, if ASICmaster is used to generate a datasheet file without specifying a name or extension, ASICmaster automatically adds the design name to the filename and appends the .txt extension.

When opening a file browser for one of the types of files listed in Table B-1 on page 117, ASICmaster lists all files with appropriate extensions. For example, when the Constraints File Selection Browser is opened to select constraint files for an ASICmaster run, the browser lists .gcf and .sdf files. Similarly, the Constraints file browser's Filters command offers the .gcf and .sdf extensions as options.

Restricting the Types of Files Listed

To restrict the types of files listed in a file browser, use the Filters command. Each type of file browser offers different filters.

For example, the Design Selection browser allows users to either display all files in the current directory or restrict the display to only EDIF, VHDL or Verilog files (Figure B-1). By default, a filter remains in effect for all work sessions for the design. To remove a filter, select the All Files option.

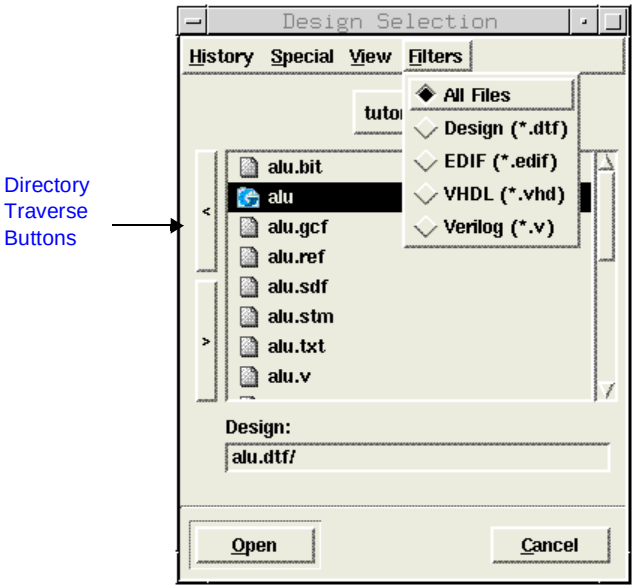


Figure B-1. Filter Menu on Design Selection File Browsers

Traversing Directories

Directories can be traversed one at a time or using directory hierarchy. To traverse directories one at a time, double click a directory in the list or select a directory and click one of the Directory traverse buttons. Figure B-2 illustrates how to traverse directories.

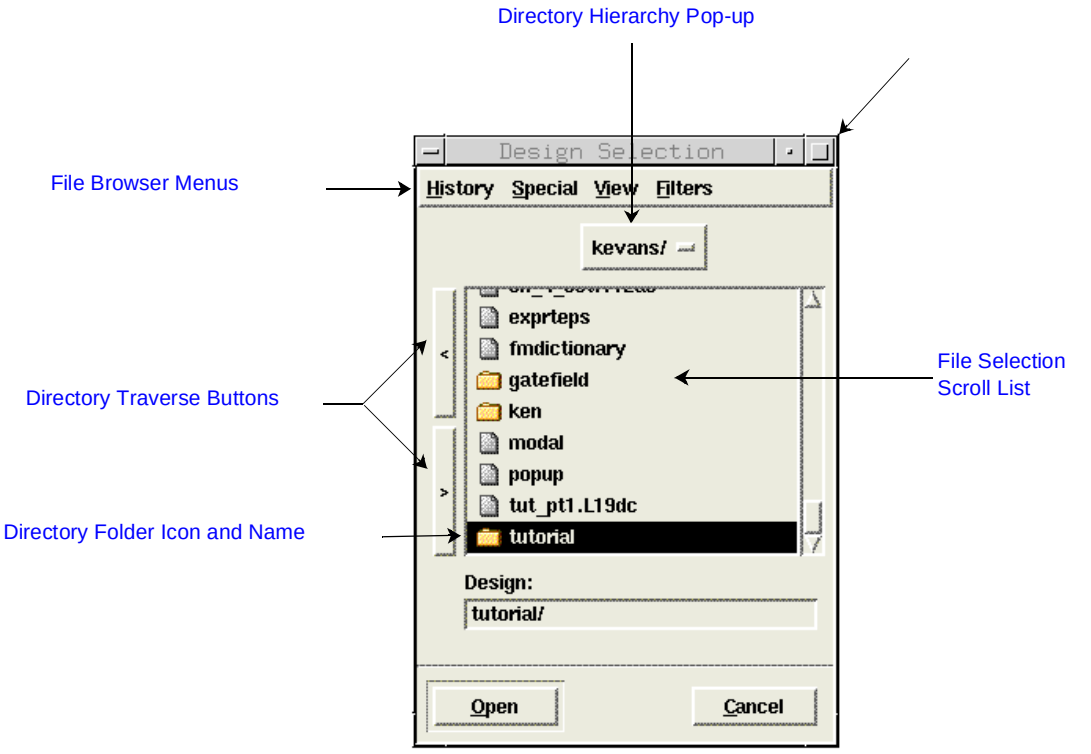


Figure B-2. Tools for Traversing Directories

To see the entire directory hierarchy above the current directory, click the directory hierarchy pull-down menu shown in Figure B-3 on page 121. The directory hierarchy menu is displayed and show the hierarchy of the root directory.

Select the name of any directory listed to display that directory's files in the file browser.

Figure B-3 shows the “users” directory two levels above the current directory named tutorial.

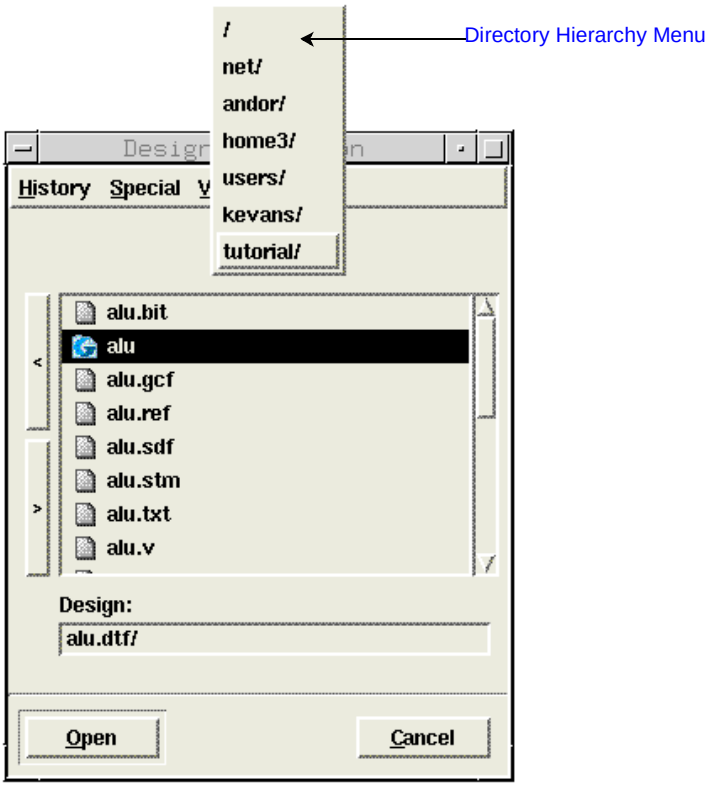


Figure B-3. Directory Hierarchy Pop-up Menu Selection

Viewing Contents of Multiple Directories

The file browser window can be expanded to view multiple file hierarchy levels. To view multiple directories, select a directory name, then horizontally drag the top-right corner or the bottom-right corner of the browser window. A maximum of five directory levels can be displayed. Figure B-4 shows the file browser expanded to three levels.

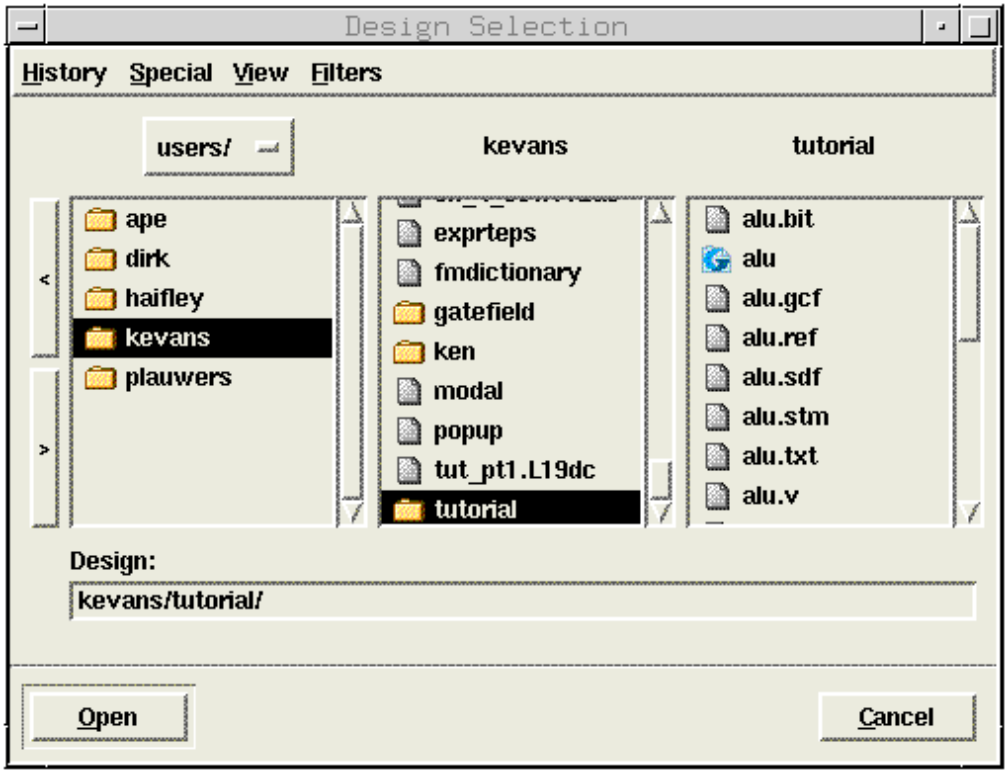


Figure B-4. Expanded Design Selection File Browser

Viewing the Directory History and Displaying Earlier Views

The History menu tracks the directories have been displayed in the file browser in reverse order of the actions taken. Click a directory name on the History menu to display that directory's contents in the current file browser.

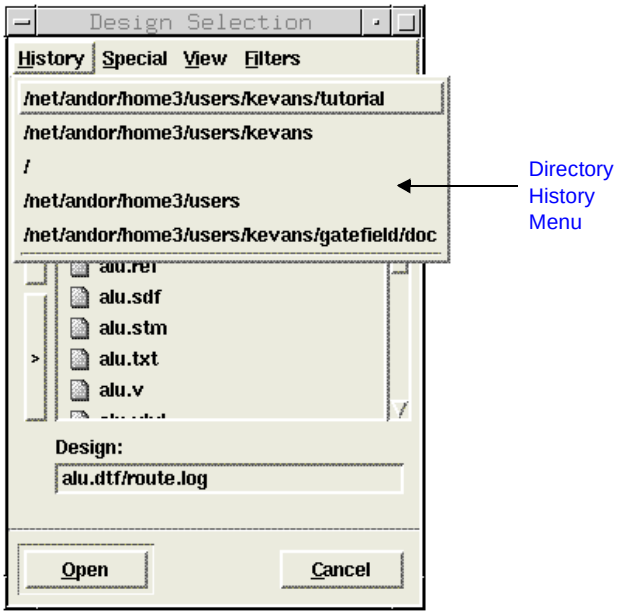


Figure B-5. History Menu on the Design Selection File Browser

Using Mnemonics and Keyboard Accelerators

A mnemonic is a keystroke or sequence of keystrokes assigned to execute a command. Mnemonics are used by combining the <Alt> key with the underlined letter to open a menu. Once opened, the commands in a menu can executed without pressing the <Alt> key. Keyboard accelerators key strokes are always combined with the <CTRL> or <Alt> key. Keyboard accelerators are displayed to the right of the mnemonic in pull-down menus, as shown in Figure B-6. Different windows in ASICmaster can assign the same accelerator keys to different functions.

Note: Although the mnemonic and accelerator keys appear in uppercase letters on menus, do not use the <Shift> key unless it is explicitly specified in the mnemonic or accelerator key sequence.

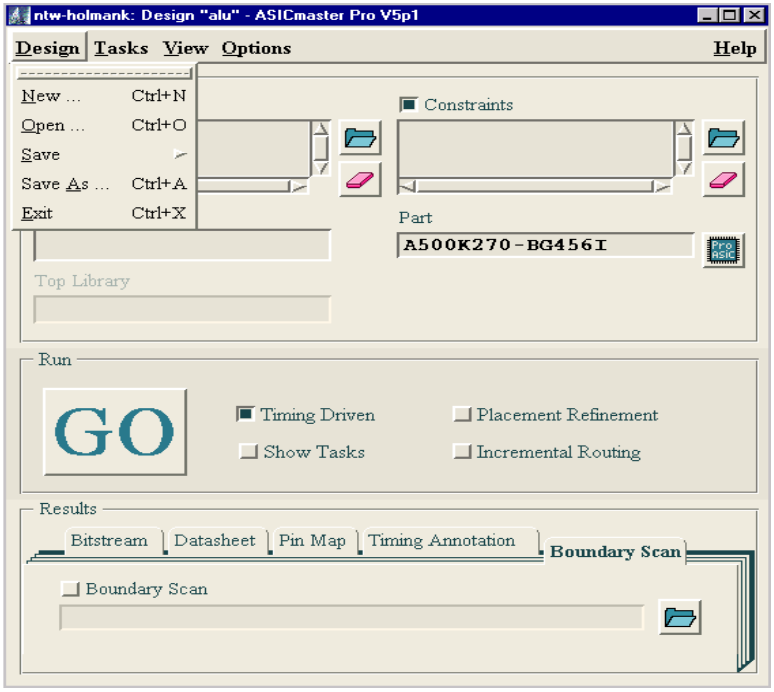


Figure B-6. Identifying Mnemonics and Keyboard Accelerators

Table B-2 lists the mnemonics and accelerator keys available in ASICmaster.

Table B-2. Menu Mnemonics and Accelerator Keys

Menu and Mnemonic	Command and Mnemonic	Accelerator Keys	Function
Design		<Alt+d>	Performs design-file administrative functions such as opening and closing design files.
	New	<Ctrl+n>	Opens the Design Selection file browser (configured for creating a new design project).

Table B-2. Menu Mnemonics and Accelerator Keys (Continued)

Menu and Mnemonic	Command and Mnemonic	Accelerator Keys	Function
	Open	<Ctrl+o>	Opens the Design Selection file browser (configured for opening an existing design project).
	Save	<Ctrl+s>	Saves the design state and associated settings.
	Save As	<Ctrl+a>	Opens the Save As Design Selection file browser (configured for specifying a new design name into which a copy of the current design and its settings is placed).
	Exit	<Ctrl+x>	Exits the Design Executive.
Tasks		<Alt+t>	Limits the task flow to the selected task and those tasks it requires; executes the tasks.
	Place & Route	<Ctrl+p>	Placement and routing.
	Run Script	<Ctrl+r>	Runs a batch script.
	Import Netlist	<Ctrl+i>	Imports the netlist and other data.
	Select Part	<Ctrl+e>	Selects a device.
	Check Input	<Ctrl+c>	Checks the input and generates parameter settings.
	Calculate Timing	<Ctrl+t>	Calculates post-layout delays.
	Generate Bitstream	<Ctrl+b>	Generates the bitstream device-programming file.
	Generate Datasheet	<Ctrl+d>	Generates the datasheet
	Generate Boundry Scan	<Ctrl+Shift+b>	Generates the boundry scan description
	Generate Pin Map	<Ctrl+m>	Generates the pin map.
	Edit Pin Map	<Ctrl+Shift+e>	Starts the pin map editor.
	Estimate Power	<Ctrl+Shift+p>	Starts the power estimation tool.

Table B-2. Menu Mnemonics and Accelerator Keys (Continued)

Menu and Mnemonic	Command and Mnemonic	Accelerator Keys	Function
View		<Alt+v>	Displays the Layout Viewer, logs, and output data files.
	Layout	<Alt+l>	Displays the physical layout in the Viewer.
	Command Log	<Alt+c>	Displays the Command Report.
	Session Log	<Alt+s>	Displays the Session Report.
	Netlist Log	<Alt+n>	Displays the Netlist Report.
	Check Log	<Alt+e>	Displays the Check Report.
	Place Log	<Alt+p>	Displays the Place Report.
	Route Log	<Alt+r>	Displays the Route Report.
	Viewer Log	<Ctrl+w>	Displays the Viewer Report.
	Timing Log	<Alt+i>	Displays the Timing Report.
	Bitgen Log	<Alt+b>	Displays the Bitstream Report.
	Files	<Alt+f>	Displays Netlist, Constraints, pin map, boundary scan, datasheet, and timing annotation files.
Options		<Alt+o>	Defines a set of potential device operating conditions and sets the default text editor.
	Design	<Alt+d>	Displays the Design Options window in which the default netlist and timing file format, and the temperature, voltage, and process operating conditions for post-layout timing calculation can be specified.
	User	<Alt+h>	Displays the User Options window in which you specify the default text editor.

Keeping Pull-Down Menus Displayed

If a pull-down menu is used frequently, it can be moved to remain on the screen instead of disappearing after a command is chosen. A menu can only be moved if it has tear-off perforations at the top, as shown in Figure B-7 on page 127.

Figure B-7. A Tear-Off Menu

To tear off a menu:

1. **Click the perforation line.**
2. **Drag the menu to a new location.** The torn-off menu becomes a window. Figure B-8 shows a torn-off menu in the MOTIF environment.

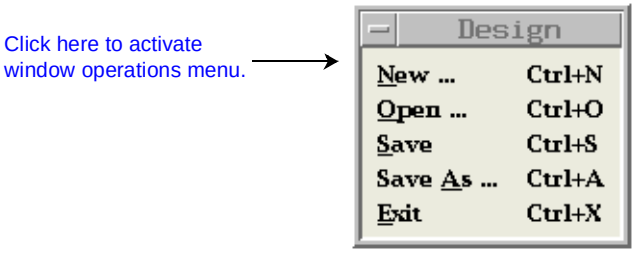


Figure B-8. A Torn-Off Menu

To close a torn-off menu:

Choose Close from the window operations menu. Tearing off a menu does not prevent it from being pulled down from the menu bar.

Using the Escape Key to Cancel Displays

The Escape key <Esc> closes the active window or subwindow. It does not backtrack through previous tasks. It works only in terms of the open windows. The <Esc> key can be pressed successively to work backwards through display actions. The <Esc> key does not close the Design Executive window.

C

Supported Netlist Formats and Constructs

This appendix provides information on the netlist formats and constructs that ASICmaster supports.

Netlist Formats Accepted by ASICmaster

ASICmaster accepts netlists in Verilog, VHDL and EDIF 2.0.0 formats. If importing multiple netlists for a design, all must have the same format. Full descriptions of Verilog, VHDL and EDIF 2.0.0 formats can be found in the reference manuals for these languages.

Because ASICmaster converts netlists to physical designs and maps them to ProASIC devices, ASICmaster accepts only structural entities instead of higher-level, more abstract entities. Each netlist input to ASICmaster must be entirely structural (with the lowest level of the structure being the leaf cells supported by ASICmaster). The netlist itself can be hierarchical, but the lowest hierarchical level must contain the basic design objects that ASICmaster recognizes.

Although behavioral statements have no meaning to ASICmaster, it fully supports the use of Synopsys and Exemplar synthesis tools. Users can describe a design in HDL and then synthesize down to the structural level, using the Synopsys Design Compiler or FPGA Compiler and the Design Kit for the Synopsys Synthesis Environment. Likewise it's possible to use Exemplar's Leonardo and the Design Kit for the Exemplar synthesis environment. The resulting output can have either the Verilog, the VHDL or the EDIF 2.0.0 format accepted by ASICmaster.

Before design layout, ASICmaster performs a detailed syntax and structure check of the imported netlist and reports errors.

Whenever possible (even for small designs), simulate the structural netlist as well as the HDL description to check for accuracy. For this purpose, ASICmaster offers Verilog and VHDL simulation models and timing models for use with Static Timing Analyzers.

Netlist Constructs Supported

The ASICmaster netlist reader supports the following constructs:

CAUTION: The constructs listed for each language comprise a *subset* of the complete description found in the reference manual for each language.

Verilog Constructs

The ASICmaster Verilog reader reads in Verilog files and creates a Netlist Data Model. A parser does the following:

- Operates on the Verilog files.
- Checks for any syntax and semantic errors in the description.
- Processes only structural Verilog.

If the parser finds any behavioral constructs, it reports errors.

Verilog Names

Verilog names are maintained throughout ASICmaster.

Assign Statements

The simple assign statements in Verilog (one signal assigned to another; no expressions allowed) are accepted and kept as aliases. Such aliases are automatically resolved across the design.

Primitives and Connections

The Verilog description should *not* contain instances of Verilog standard primitives (or gates) or user-defined primitives (UDP's).

Only instances of ASICmaster library (A500K.v) primitives are permitted in the netlist. Connections between instances can only be net, bit-select of a net, part-select of a net, or a concatenation of the previous three. Expressions are not permitted.

EDIF 2.0.0

The EDIF netlist reader creates a data model for EDIF 2.0.0 NETLIST views only and ignores all other EDIF view types. The reader handles the constructs listed in Table C-1.

Table C-1. EDIF Constructs Recognized by ASICmaster

Construct	Used In
array	*netNameDef, portNameDef*
cell	*external, library —> cellNameDef*
cellRef	*design, viewRef —> cellNameRef*
cellType	Generic | Tie | Ripper (No support for tie cells)
contents	view
design	*edif —> designNameDef*
direction	port
edif	the *top —> edifFileNameDef, edifLevel, edifVersion*
external	*edif —> libraryNameDef, edifLevel, technology*
instance	*contents, net —> instanceNameRef*
instanceNameRef	instanceRef, netRef, portRef —> instanceNameRef
interface	view
joined	interface, net
library	edif —> libraryNameDef, edifLevel, technology
libraryRef	cellRef —> libraryNameRef
listOfNets	netBundle
listOfPorts	portBundle
member	netNameRef, portNameRef
net	contents, listOfNets, net —> netNameDef

Table C-1. EDIF Constructs Recognized by ASICmaster (Continued)

Construct	Used In
netBundle	*contents —> netNameDef* (Support for only one level of hierarchy)
port	interface, listOfPorts —> portNameDef
portBundle	*interface, listOfPorts —> portNameDef* (Support for only one level of hierarchy)
portList	joined
portRef	joined, portList —> portRef, portNameRef
property	cell, design, instance, net, port, view —> properTyNameDef
rename	nameDef
view	cell —> viewNameDef
viewRef	instance, instanceRef, netRef, portRef viewList
viewType	view

The following constructs are understood but ignored:

author	comment
criticality	dataOrigin
nonPermutable	numberDefinition
owner	permutable
program	scale
status	technology
timeStamp	userData
version	write

EDIF Names

The EDIF rename string, if present, is used for identifier names instead of EDIF names.

Port and Net Arrays

Only one-dimensional port and net arrays are supported.

Instance Arrays

Only one-dimensional instance arrays are supported. It is also assumed that all the member elements refer to the same cellView, so that the viewList construct is not needed.

Port and Net Bundles

Port and net bundles are supported by flattening them.

For example, all of a port bundle's member ports are treated as if they occurred inside the interface form. In addition, all of a net bundle's member nets are treated as if they occurred inside the contents form. It is also assumed that a bundle does not contain additional bundles.

Ripper Cells

Instances of ripper cells are supported. It is assumed that all the ripper cells have only two ports that are fully connected to each other. The size of the ports can be 1 or greater.

Tie Cells

Tie cells usually do not occur in netlist views and are not supported.

Properties

EDIF properties are attached to the owner object. All types of values (whether integer, real, or string) are converted to string. If the integer and real values have min:typ:max values, these become part of the converted string.

Leaf Cells

In EDIF, leaf cells are described by omitting the contents form from their description. Therefore, any cell that does not have a contents form is considered a leaf cell.

VHDL

The ASICmaster VHDL reader conforms to the IEEE standard 1076-1987 VHDL. The reader accepts VHDL files, analyzes them, and creates a Netlist Data Model. It also checks the description for any syntax and semantic errors and processes only structural VHDL.

A built-in IEEE 1164 standard logic package “std_logic_1164” in library “IEEE” is available, which the VHDL netlist can refer to in a Use clause: “USE IEEE.std_logic_1164.ALL;”.

If the parser finds any behavioral constructs, it gives errors. Also configuration statements are not supported. Table C-2 lists supported VHDL constructs. Table C-3 on page 135 lists VHDL constructs that are understood but not used.

Table C-2. Supported VHDL Constructs

Construct	Relies On
Design units	Entity declaration, Architecture declaration
Entity declaration	Generic clause, Port clause, Use clause
Architecture declaration	Component declaration, Signal declaration, Use clause for library and package, Concurrent statement
Concurrent statement	Concurrent signal assignment statement, Component instantiation statement
Expression	Indexed names, Slice names, Selected names, Literals (‘0’ and ‘1’), Aggregates

Table C-3. Understood but Not Used VHDL Constructs

Construct	Relies On
Design units	Package declaration
Package declaration	Type declaration, Subtype declaration, Constant declaration, Component declaration, Use clause, Function declaration
Type declaration	Predefined enumeration type, Enumeration type declaration, Constrained array definition, Unconstrained array definition
Attribute declaration	
Attribute specification	
Predefined attributes	
Resolution functions	
Expression	Type conversions

VHDL Names

VHDL is a case-insensitive language. All names are retained by ASICmaster.

Assignment Statements

Concurrent signal assignment statements involving simple names (scalar or array), indexed names, slice names, aggregates and literals ('0' and '1') are supported. No operators are supported. Signals occurring on the LHS are made into aliases of the corresponding signals on the RHS. Such aliases are automatically resolved across the design.

Leaf Cells

All component instantiations statements which are not finally bound to any entity-architecture pair are treated as instantiations of leaf cells.

D

The Design (*.dtf) Directory

The database for a specific design resides in an ASICmaster design directory, which contains information ASICmaster uses for processing. The directory name always has the suffix .dtf, which stands for design tasks filing.

The design database contains several files, which are described in the sections that follow. All file formats are internal and can be changed by ASICmaster.

CAUTION: Do not edit, move, rename, or otherwise change the .dtf directory or the files it contains. These files form a consistent set of data that reflects the state of the design. Never merge two design directories or try to "repair" design directory files by editing their contents. Instead, use ASICmaster to regenerate the data from the input source.

Files Created by ASICmaster

The following files are created by ASICmaster as part of its normal operation.

Table D-1. Internal ASICmaster Design Files

File	Contains	Created by	Used by
abs_placement	Placement data	Placer	Router
blocks.in	Cell instances	Netlister	Checker
blocks	Cell instances	Checker	Placer
constraints	Physical and timing constraints	Checker	Placer, Router, Viewer
deleted_blocks	Deleted cell instances	Checker	Timing generator
designState	Design state parameters	Design Executive	Design Executive
hier.in	Cell hierarchy	Netlister	Checker

Table D-1. Internal ASICmaster Design Files

File	Contains	Created by	Used by
inverted_ports	List of inverted ports per net	Checker	Router
last_placement.gcf	Data for placement refinement	Placer	Checker
mem_plmt.gcf	Memory placement constraints	Checker	Checker
nets.in	Nets	Netlister	Checker
nets	Nets	Checker	Placer, Router, Viewer, Timing generator, Bitstream generator
placement	Placement data	Placer	Viewer, Timing generator, Bitstream generator
portrefs	Cross-reference between port names and I/O instance names	Netlister	Checker, Viewer, Datasheet generator
routes	Geometric routing data	Router	Viewer
swloc	Switch locations	Router	Timing generator, Bitstream generator
tera*	Internal placer files	Placer	Placer

designState File Contents

The designState file contains parameters, associated values, and last-modification timestamp data. All contribute to the intelligent behavior of the Design Executive as it tracks design status and task requirements. The design state parameters cannot be directly edited, but are influenced by changes made when the graphical user interface is used or batch mode is

run. The designState file contains the parameters described in Table D-2, as well as additional parameters, which are internal to the system.

Table D-2. designState Parameters

Task Parameter	Type	Value Definition
SetTaskParameters(overrulethesystem'sdefaultforthevaluedefinitionsinthethird column.)		
BitstreamTask	Boolean	Generate device-programming bitstream task
BoundaryScanTask	Boolean	Generate boundary scan task
CheckTask	Boolean	Check design task
DatasheetTask	Boolean	Generate datasheet task
NetlistTask	Boolean	Import netlist task
PinmapTask	Boolean	Generate pin map task
PlaceTask	Boolean	Place task
RouteTask	Boolean	Route task
TimingTask	Boolean	Generate timing annotation task
Set Check Parameters (overrule the system's default for the value definitions in the third column.)		
BitstreamCheck	Boolean	Generating the bitstream device-programming file
BoundaryScanCheck	Boolean	Generating a boundary scan
ConstraintsCheck	Boolean	Import the specified constraints files
DataSheetCheck	Boolean	Generating a datasheet
PinmapCheck	Boolean	Generating the pin map file
TimingCheck	Boolean	Generating a timing annotation file
Netlist Parameters		

Table D-2. designState Parameters (Continued)

Task Parameter	Type	Value Definition
NetlistFormat	String	One of: "Verilog" | "VHDL" | "EDIF"
TopCell	String	Name of the selected top cell in the netlist (required for VHDL)
TopLib	String	Name of the top cell's library (applicable in EDIF)
Filename Parameters		
BitstreamFile	Filename	The filename for the bitstream device-programming file
BoundaryScanFile	Filename	The filename for the boundary scan
DataSheetFile	Filename	The filename for the datasheet
PinmapFile	Filename	The filename for the pin map
TimingFile	Filename	The filename for the timing annotation file
Physical Parameters		
Package	String	The Programmable ASIC part name
PlacementRefinement	Boolean	Placement Refinement
IncrementalRouting	Boolean	Incremental Routing.
TimingDriven	Boolean	Timing Driven Place and Route
Operating Condition Parameters		
Default_Pad_Load	Number	Default pad load used in timing calculations, in pF
Min_Temperature	Number	Minimum junction temperature used in timing calculations
Typ_Temperature	Number	Typical junction temperature used in timing calculations

Table D-2. designState Parameters (Continued)

Task Parameter	Type	Value Definition
Max_Temperature	Number	Maximum junction temperature used in timing calculations
Min_Voltage	Number	Minimum voltage used in timing calculations
Typ_Voltage	Number	Typical voltage used in timing calculations
Max_Voltage	Number	Maximum voltage used in timing calculations
Min_Process	String[a]	Minimum process used in timing calculations
Typ_Process	String	Typical process used in timing calculations
Max_Process	String	Maximum process used in timing calculations
Report	String[b]	Values reported in the timing log.

a. One of: "Best" | "Typical" | "Worst"

b. One of: "Fastest" | "Typical" | "Slowest"

E

Batch Script Syntax

A batch script file consists of commands, written one per line. Users specify the batch script file when starting the ASICmaster in batch mode. At that time, commands from the batch script file are executed in sequence.

The language of the batch script file is identical to the language used in interactive TTY mode at the ASICmaster prompt and in the Command Log window's command text field. Refer to "Using ASICmaster in TTY and Batch Modes" on page 87 for information about creating a batch file and an example of a batch script file.

A batch script can be executed in two ways:

- Include the batch option on the command line.
- Choose the Run Script command from the Tasks menu in the Design Executive window.

The syntax for a batch script file is as follows:

script_file ::= *script_file_statements*

script_file_statements ::= *script_file_statement* end_of_line

| *script_file_statements script_file_statement* end_of_line

script_file_statement ::= {

// comments

| # comments

| set *designState_parameter value_till_end_of_line*

| show [*designState_parameter*]

| remove *designState_list_parameter*

| add *designState_list_parameter value_till_end_of_line*

| initialize | go | reset

| place and route

| import netlist

| check | timing

| datasheet | pinmap

| boundaryscan

| bitstream

| graphics

| help

| quit | exit

}

designState_list_parameter ::= {NetListFiles | ConstraintsFiles }

designState_parameter ::= { TopCell | TopLib | Package

| BitstreamFile | BoundaryscanFile | DataSheetFile | PinmapFile

| TimingFile

| PlacementRefinement | IncrementalRouting

| UseJTAG | UseExtendedJTAG

| BitstreamCheck | BoundaryScanCheck | ConstraintsCheck

| DataSheetCheck | PinmapCheck | TimingCheck

| NetlistFormat

| Min_Temperature | Typ_Temperature | Max_Temperature

| Min_Voltage | Typ_Voltage | Max_Voltage

| Min_Process | Typ_Process | Max_Process | Report

| NetlistTask | CheckTask | PlaceTask | RouteTask | TimingTask

| DatasheetTask | BoundaryScanTask | PinmapTask | BitstreamTask

}

value_till_end_of_line ::= {

filename | *string* | *boolean* | *temperature* | *voltage* | *netlistformat* | *process* | *report*

}

filename ::= a valid filename

string ::= whatever is there till the end of the line, may contain embedded blanks.

boolean ::= {true | false }

temperature ::= number ranging from -55 to 125.

voltage ::= number ranging from 2.00 to 4.00 and using increments of 0.05

netlistformat ::= {"Verilog" | "EDIF" | "VHDL" }

process ::= { "Best" | "Typical" | "Worst" }

report ::= { "Slowest" | "Typical" | "Fastest" }

F

The .amrc Startup File

At system startup, the .amrc file initializes ASICmaster configuration. When ASICmaster is launched, the Design Executive, Netlist Importer, and Viewer read the .amrc file from the home directory and initialize variables:

By entering or editing parameters in the file, users can tailor settings to suit their design project. Settings in the ASICmaster user interfaces override settings in the .amrc file.

Users can also reset some of these parameters through ASICmaster's graphic user interface. When users exit the Design Executive window or click OK in the User Options window, ASICmaster updates the .amrc file to match new settings made in ASICmaster windows.

Each parameter in the .amrc file consists of a name, followed by a colon, followed by a parameter value. For example:

```
gfexec.editor: vi
```

where gfexec.editor: is the parameter name and vi is the value.

The file with the pathname $AMHOME/etc/gfdata/amrc contains a template copy of the .amrc file. Users can copy this .amrc file to $HOME/.amrc. Figure F-1 on page 148 shows the parameters and settings contained in the .amrc startup file.

Design Executive Reads

```
gfexec.editor: vi
gfexec.show_tasks: FALSE
gfexec.nl_format: Verilog
gfexec.Verilog_extension: v
gfexec.VHDL_extension: vhd
gfexec.edif_extension: edif
```

Netlist Importer Reads

```
gfnetlister.fanout_threshold: 8
```

Viewer Reads

```
gfviewer.color: Background 65535 65535 65535 1
gfviewer.color: Cells 12106 53427 52744 1
gfviewer.color: Instances 6772 1697 63836 1
gfviewer.color: Instance_Names 0 0 0 0
gfviewer.color: I/O 65535 0 0 0
gfviewer.color: I/O_Names 0 0 0 0
gfviewer.color: Highlight 65535 0 0 1
gfviewer.color: Global_Routes 5549 54533 59982 1
gfviewer.color: Macro_Routes 0 65535 0 1
gfviewer.color: V_External_Routes 0 65535 0 1
gfviewer.color: H_External_Routes 0 0 65535 1
gfviewer.color: V_Internal_Routes 0 65535 0 1
gfviewer.color: H_Internal_Routes 0 0 65535 1
gfviewer.color: Selected 65534 42011 0 0
gfviewer.color: Tile_Boundaries 18654 46879 24747 1
gfviewer.color: Unrouted 65535 0 0 1
gfviewer.printer: hpcolor
```

Figure F-1. Default .amrc File

Parameters for the Design Executive

The Design Executive reads the .amrc file to set the following interface parameters:

- The default text editor.
- The show_tasks setting.

- The default netlist format ASICmaster uses for imported netlist files.
- Extensions for EDIF, VHDL and Verilog files.

Setting the Default Text Editor

When ASICmaster requires a text editor, it uses the editor set using the editor parameter in the .amrc startup file. Valid file editor settings are:

- vi
- emacs
- A string that is an executable command in the path.
- A string that performs a file viewing function that supports editing.

The system default text editor is the vi (UNIX) or Notepad (PC). Users can change the default either by editing the .amrc file or by changing the setting in the Design Executive's User Options window.

Example parameter:

```
gfexec.editor: textedit
```

Enabling Display of the Tasks Selection List

The show_tasks setting determines whether or not the Tasks selection list appears when clicking the GO button in the Design Executive window. The system default value is FALSE. Users can change this setting either by editing the .amrc file or by checking the Show Tasks check box in the Design Executive window.

Example parameter:

```
gfexec.show_tasks: FALSE
```

Setting the Default Netlist Format

ASICmaster uses the nl_format parameter in the .amrc file as the default filter for the Netlist Selection file browser when creating a new design. Valid settings for the netlist format are:

- edif
- vhdl
- verilog

Users can set the netlist format either by editing the .amrc file or by using the Design command in the Options menu in the Design Executive Window.

Example parameter:

```
gfexec.nl_format: EDIF
```

Identifying Extensions for EDIF, VHDL and Verilog Files

ASICmaster uses the edif_extension parameter in the .amrc file to identify file extensions to appended to netlist files. When the Filters command on file browsers is used ASICmaster can restrict the file listing to those files with the specified extensions.

When editing the extension parameter value, do not include the period (.) that precedes actual file extensions.

Example parameters:

```
gfexec.edif_extension: edif2
gfexec.vhdl_extension: vhdl
gfexec.Verilog_extension: v
```

Parameters for the Netlist Importer

The netlist importer reads the fanout_threshold parameter in the .amrc file to set the default fanout threshold per net. When users set a default maximum net fanout in the .amrc file, the netlist importer reports all nets in a design that include a fanout higher than the threshold. The information is available in the Netlist report.

Example parameter:

```
gfnetlister.fanout_threshold: 8
```

Parameters for the Viewer

The Viewer reads the .amrc startup file to set parameters for colors assigned to layout objects such as cells, global nets, and unrouted nets. It also registers your last printer setting.

Parameters can be changed by editing them in the .amrc file or by using the Viewer in the ASICmaster graphic user interface. Actel recommends using the graphical user interface. Refer to "Customizing Viewer Setup" on page 55 for additional information

Example parameters:

```
gfviewer.color: Background 65535 65535 65535 1
gfviewer.color: Cells 12106 53427 52744 1
gfviewer.color: Instances 6772 1697 63836 1
gfviewer.color: Instance_Names 0 0 0 0
gfviewer.color: I/O 65535 0 0 0
gfviewer.color: I/O_Names 0 0 0 0
gfviewer.color: Highlight 65535 0 0 1
gfviewer.color: Global_Routes 5549 54533 59982 1
gfviewer.color: Macro_Routes 0 65535 0 1
gfviewer.color: V_External_Routes 0 65535 0 1
gfviewer.color: H_External_Routes 0 0 65535 1
gfviewer.color: V_Internal_Routes 0 65535 0 1
gfviewer.color: H_Internal_Routes 0 0 65535 1
gfviewer.color: Selected 65534 42011 0 0
gfviewer.color: Tile_Boundaries 18654 46879 24747 1
gfviewer.color: Unrouted 65535 0 0 1
```

The numbers in the parameter values represent RGB colors. The number 1 at the end of a parameter indicates "ON," and the number 0 indicates "OFF."

G

Product Support

Actel backs its products with various support services including Customer Service, a Customer Applications Center, a Web and FTP site, electronic mail, and worldwide sales offices. This appendix contains information about using these services and contacting Actel for service and support.

Actel U.S. Toll-Free Line

Use the Actel toll-free line to contact Actel for sales information, technical support, requests for literature about Actel and Actel products, Customer Service, investor information, and using the Action Facts service.

The Actel Toll-Free Line is (888) 99-ACTEL.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call (408) 522-4480.
From Southeast and Southwest U.S.A., call (408) 522-4480.
From South Central U.S.A., call (408) 522-4434.
From Northwest U.S.A., call (408) 522-4434.
From Canada, call (408) 522-4480.
From Europe, call (408) 522-4252 or +44 (0) 1256 305600.
From Japan, call (408) 522-4743.
From the rest of the world, call (408) 522-4743.
Fax, from anywhere in the world (408) 522-8044.

Customer Applications Center

The Customer Applications Center is staffed by applications engineers who can answer your hardware, software, and design questions.

All calls are answered by our Technical Message Center. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:30 a.m. to 5 p.m., Pacific Standard Time, Monday through Friday.

The Customer Applications Center number is (800) 262-1060.

European customers can call +44 (0) 1256 305600.

Guru Automated Technical Support

Guru is a Web based automated technical support system accessible through the Actel home page (**http://www.actel.com/guru/**). Guru provides answers to technical questions about Actel products. Many answers include diagrams, illustrations and links to other resources on the Actel Web site. Guru is available 24 hours a day, seven days a week.

Web Site

Actel has a World Wide Web home page where you can browse a variety of technical and non-technical information. Use a Net browser (Netscape recommended) to access Actel's home page.

The URL is **http://www.actel.com**. You are welcome to share the resources we have provided on the net.

Be sure to visit the "Actel User Area" on our Web site, which contains information regarding: products, technical services, current manuals, and release notes.

FTP Site

Actel has an anonymous FTP site located at **ftp://ftp.actel.com**. You can directly obtain library updates, software patches, design files, and data sheets.

Electronic Mail

You can communicate your technical questions to our e-mail address and receive answers back by e-mail, fax, or phone. Also, if you have design problems, you can e-mail your design files to receive assistance. The e-mail account is monitored several times per day.

The technical support e-mail address is **tech@actel.com**.

Worldwide Sales Offices

Headquarters

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
Toll Free: 888.99.ACTEL

Tel: 408.739.1010
Fax: 408.739.1540

US Sales Offices

California

Bay Area
Tel: 408.328.2200
Fax: 408.328.2358

Irvine
Tel: 949.727.0470
Fax: 949.727.0476

San Diego
Tel: 619.938.9860
Fax: 619.938.9887

Thousand Oaks
Tel: 805.375.5769
Fax: 805.375.5749

Colorado

Tel: 303.420.4335
Fax: 303.420.4336

Florida

Tel: 407.677.6661
Fax: 407.677.1030

Georgia

Tel: 770.831.9090
Fax: 770.831.0055

Illinois

Tel: 847.259.1501
Fax: 847.259.1572

Maryland

Tel: 410.381.3289
Fax: 410.290.3291

Massachusetts

Tel: 978.244.3800
Fax: 978.244.3820

Minnesota

Tel: 612.854.8162
Fax: 612.854.8120

North Carolina

Tel: 919.376.5419
Fax: 919.376.5421

Pennsylvania

Tel: 215.830.1458
Fax: 215.706.0680

Texas

Tel: 972.235.8944
Fax: 972.235.965

International Sales Offices

Canada
Suite 203
135 Michael Cowpland Dr,
Kanata, Ontario K2M 2E9

Tel: 613.591.2074
Fax: 613.591.0348

France
361 Avenue General de Gaulle
92147 Clamart Cedex

Tel: +33 (0)1.40.83.11.00
Fax: +33 (0)1.40.94.11.04

Germany
Bahnhofstrasse 15
85375 Neufahrn

Tel: +49 (0)8165.9584.0
Fax: +49 (0)8165.9584.1

Hong Kong
Suite 2206,
Parkside Pacific Place,
88 Queensway

Tel: +011.852.2877.6226
Fax: +011.852.2918.9693

Italy
Via Giovanni da Udine No. 34
20156 Milano

Tel: +39 (0)2.3809.3259
Fax: +39 (0)2.3809.3260

Japan
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150

Tel: +81 (0)3.3445.7671
Fax: +81 (0)3.3445.7668

Korea
135-090, 18th Floor,
Kyoung Am Building
157-27 Samsung-dong
Kangnam-ku, Seoul

Tel: +82 (0)2.555.7425
Fax: +82 (0)2.555.5779

Taiwan
4F-3, No. 75, Sec. 1,
Hsin-Tai-Wu Road,
Hsi-chih, Taipei, 221

Tel: +886 (0)2.698.2525
Fax: +886 (0)2.698.2548

United Kingdom
Daneshill House,
Lutyens Close
Basingstoke,
Hampshire RG24 8AG

Tel: +44 (0)1256.305600
Fax: +44 (0)1256.355420

Index

D

E

F

G

H

I

R

S

T

U

V

W

X

Z