ZeBu ® zManualPartitioner Application Note

Version V-2024.03-1, July 2024

SYNOPSYS®

Copyright and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.
All other product or company names may be trademarks of their respective owners.

Free and Open-Source Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

Feedback

Contents

Preface

This section contains the following topics:

About This Book

The *ZeBu® zManualPartitioner Application Note* describes the *zManualPartitioner* feature for ZeBu. It is a graphical interface that allows you to manually partition a design after a compilation with ZeBu.

Audience

This guide is written for experienced ZeBu users who are familiar with design partitioning.

Contents of This Book

The *ZeBu® zManualPartitioner Application Note* has the following chapters:

Chapter	Describes...
Design Partitioning With zManualPartitioner	*zManualPartitioner* GUI options
Managing Black-boxes	about managing black-boxes for design partitioning
Moving Instances Between Partitions	about shifting instances from one partition to another
Limitations of zManualPartitioner	the limitations of *zManualPartitioner*

Feedback

Related Documentation

Document Name	Description
ZeBu User Guide	Provides detailed information on using ZeBu.
ZeBu Debug Guide	Provides information on tools you can use for debugging.
ZeBu Debug Methodology Guide	Provides debug methodologies that you can use for debugging.
ZeBu Unified Command-Line User Guide	Provides the usage of Unified Command-Line Interface (UCLI) for debugging your design.
ZeBu UTF Reference Guide	Describes Unified Tcl Format (UTF) commands used with ZeBu.
ZeBu Power Aware Verification User Guide	Describes how to use Power Aware verification in ZeBu environment, from the source files to runtime.
ZeBu Functional Coverage User Guide	Describes collecting functional coverage in emulation.
Simulation Acceleration User Guide	Provides information on how to use Simulation Acceleration to enable cosimulating SystemVerilog testbenches with the DUT
ZeBu Verdi Integration Guide	Provides Verdi features that you can use with ZeBu. This document is available in the Verdi documentation set.
ZeBu Runtime Performance Analysis With zTune User Guide	Provides information about runtime emulation performance analysis with zTune.
ZeBu Custom DPI Based Transactors User Guide	Describes ZEMI-3 that enables writing transactors for functional testing of a design.
ZeBu LCA Features Guide	Provides a list of Limited Customer Availability (LCA) features available with ZeBu.
ZeBu Synthesis Verification User Guide	Provides a description of zFmCheck.
ZeBu Transactors Compilation Application Note	Provides detailed steps to instantiate and compile a ZeBu transactor.
ZeBu zManualPartitioner Application Note	Describes the zManualPartitioner feature for ZeBu. It is a graphical interface to manually partition a design.
ZeBu Hybrid Emulation Application Note	Provides an overview of the hybrid emulation solution and its components.

Typographical Convention

This document uses the following typographical conventions:

To indicate	Convention Used
Program code	`OUT <= IN;`
Object names	`OUT`
Variables representing objects names	`<sig-name>`
Message	Active low signal name '<sig-name>' must end with _X.
Message location	`OUT` <= IN;
Reworked example with message removed	`OUT_X` <= IN;
Important Information	**NOTE:** This rule...

The following table describes the syntax used in this document:

Syntax	Description
[ ] (Square brackets)	An optional entry
{ } (Curly braces)	An entry that can be specified once or multiple times
| (Vertical bar)	A list of choices out of which you can choose one
... (Horizontal ellipsis)	Other options that you can specify

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our

software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

1

Design Partitioning With zManualPartitioner

ZeBu provides a graphical user interface, `zManualPartitioner` that allows manual partitioning of specific parts of a design. The interface allows drag and drop to manipulate design instances after compilation.

The manual partitioning also exists in `zNetgen`, a Tcl-based ZeBu tool.

This section explains how to use the `zManualPartitioner` GUI. This section consists of the following subsections:

Launching zManualPartitioner

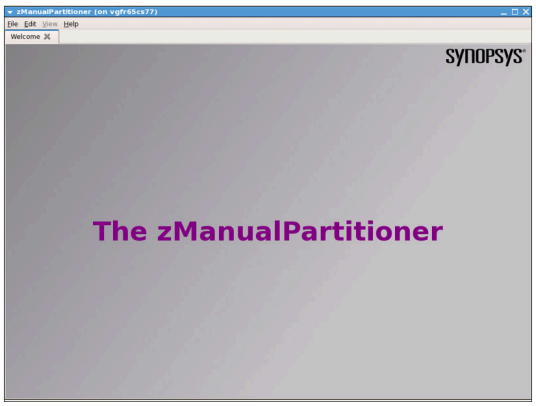
To launch the `zManualPartitioner` GUI, use the following command:

```
$ zManualPartitioner
```

Feedback

This command opens the following welcome screen:

Figure 1 ManualPartitioner Welcome Screen



Loading a Netlist File

You can load two types of netlist files into the `zManualPartitioner` GUI:

- EDIF
- Automatic Clustering Interconnection

This section consists of the following sub-sections:

- Loading an EDIF File
- Loading an Automatic Clustering Interconnection File

Loading an EDIF File

To load an EDIF file, perform the following steps:

1. Select **File** > **Open EDIF** file.
2. Browse to your EDIF file. Extensions accepted are `.edf`, `.edf.gz`, and `.edif`.
3. Click **Open**.

   Note:

   You can load only one EDIF file at a time in a `zManualPartitioner` session.

Feedback

Loading an Automatic Clustering Interconnection File

You can load an interconnection file in `zManualPartitioner` if you have compiled your design at zCore-Level with Automatic Clustering enabled.

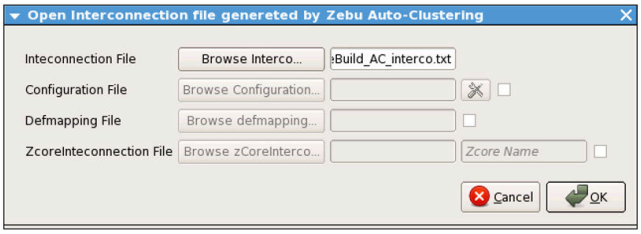
Note:

You can load multiple Automatic Clustering interconnection files in parallel in a `zManualPartitioner` session.

To load an Automatic Clustering Interconnection file, perform the following steps:

1. Click **File** > **Open AC file** to open the following dialog box:

Figure 2 Loading an Automatic Clustering Interconnection File



Note:

To retrieve an Interconnection file for a design, generated at **zCoreBuild** level, you must activate automatic clustering at the **zTopBuild** level.

2. Click **Browse Interco…** to select your Automatic Clustering Interconnection file.

In addition to the Automatic Clustering Interconnection file, you can also load a configuration file, a defmapping file, and a `ZcoreInterconnection` file to reproduce and modify partitioning results as needed. By default, these fields are inactive in the window. To activate these fields, click the check box next to them (see Figure 2). For more details on these files, see the following table:

Table 1 Additional Files to Modify Partitioning Results

Files	Description	Mandatory/Optional
Integration Files	This is the `zCoreBuild_AC_interco.txt` file that is generated by automatic clustering after compilation at zCore-level (`zCoreBuild`). It can also be generated by the system-level compiler (`zTopBuild`).This file does not require the EDIF file to perform partitioning.It is available in your zCore-level working directory, that is, `work.<core_name>`.	Mandatory

Feedback

Table 1 Additional Files to Modify Partitioning Results (Continued)

Files	Description	Mandatory/Optional
Configuration Files	This is a target configuration file that selects the ZeBu configuration in the `$ZEBU_ROOT/etc/configurations` directory.To load a configuration file, select the corresponding check box and click the **Browse Configuration** button.**Note:** If you click the  button, the **Configuration** window appears, which allows you to customize the configuration. For more information, see Configuration Window.	Optional
Defmapping File	This is the `zCoreBuild_AC_defmapping.tcl` file associated with the `zCoreBuild_AC_interco.txt` file. It contains the results of a previous partitioning.This file is available in your zCore-level working directory, that is, `work.<core_name>`.	Optional
zCoreInterconnection File	This is the `zTopBuild_AC_TOPNL_interco.txt` file that describes the interconnection information between all `zCores`. This file is generated by the system-level compiler (`zTopBuild`) with the cluster `enable -map_inter_core_io` command.	Optional

Main Window

The **zManualPartitioner** main window changes depending on the file you load, that is, an EDIF file or an Automatic Clustering Interconnection file.

Feedback

Figure 3 EDIF File loaded in zManualPartitioner

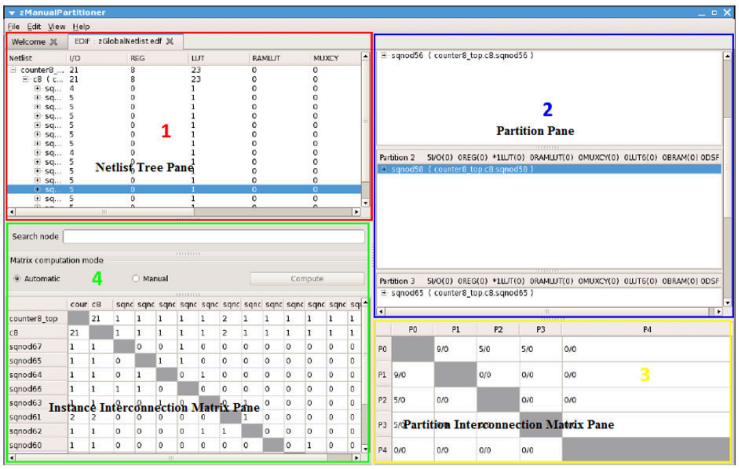
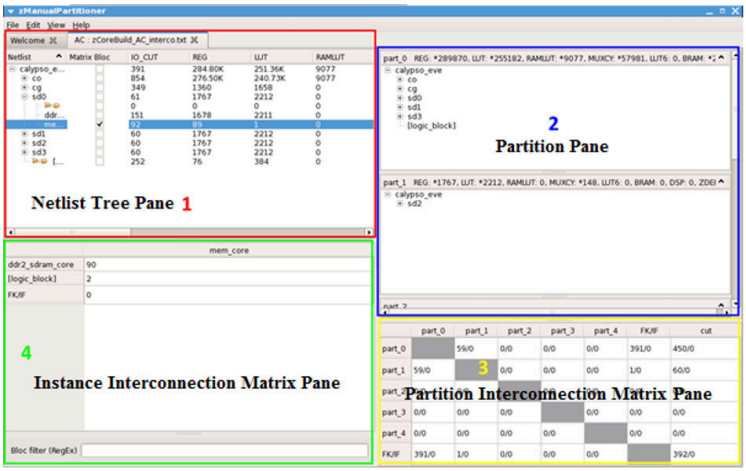


Figure 4 Automatic Clustering Interconnection File loaded in zManualPartitioner



The main window can be divided into the following four panes, (see Figure 3 and Figure 4).

- Netlist Tree Pane
- Partition Pane
- Partition Interconnection Matrix Pane
- Instance Interconnection Matrix Pane

The **Instance Interconnection Matrix** pane is replaced by the **Timing Analysis** pane when you launch the **Timing Analysis** feature. For more details, see Timing Analysis Pane.

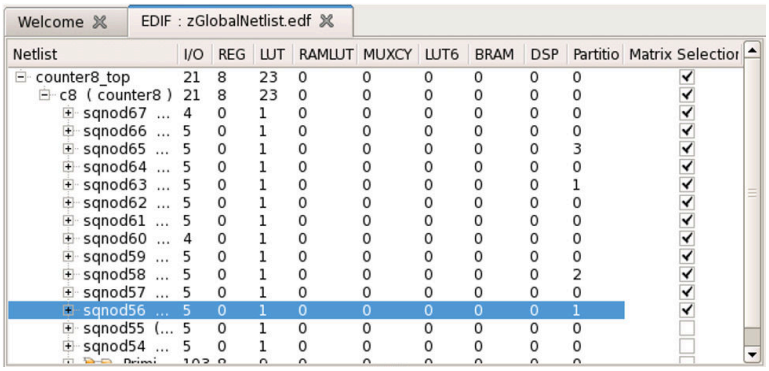
Netlist Tree Pane

The **Netlist Tree** pane allows you to view all the available instances in a netlist file. The pane changes depending on the file you load into `zManualPartitioner`.

Using an EDIF File

The following figure displays the **Netlist Tree** pane, when you load an EDIF file in `zManualPartitioner`.

Figure 5 Netlist Tree Pane (EDIF File)



The following table describes the columns available in the **Netlist Tree** pane.

Table 2 Column Description in the Netlist Tree Pane (EDIF File)

Column	Description
Netlist	Displays the list of the following available items in the netlist in a tree view:The top of a design. Common instances followed by their module `nameLogical` gates. Black box in the original design or in `zManualPartitioner`.(For more information on black-box management in `zManualPartitioner`, see Managing Black-boxes).Primitives containing logical gates and black-boxes at the same level.
I/O	Displays the number of inputs/outputs.
REG	Displays the number of registers.
LUT	Displays the number of lookup tables.
RAMLUT	Displays the number of RAMLUTs.
MUXCY	Displays the number of multiplexers for Carry Logic.

Feedback

Table 2 Column Description in the Netlist Tree Pane (EDIF File) (Continued)

Column	Description
LUT6	Displays the number of LUT6s.
BRAM	Displays the number of BRAMs.
DSP	Displays the number of DSPs.
Partition	Partition number where an instance is currently located. It is dynamically updated when you move an instance from one partition to another.
Matrix Selection	A Check box that allows you to select as many items as necessary to display in the **Instance Interconnection Matrix** pane.

The following table displays the icons available in the **Netlist Tree** pane.

Table 3 Available Icons in the Netlist Tree Pane (EDIF File)

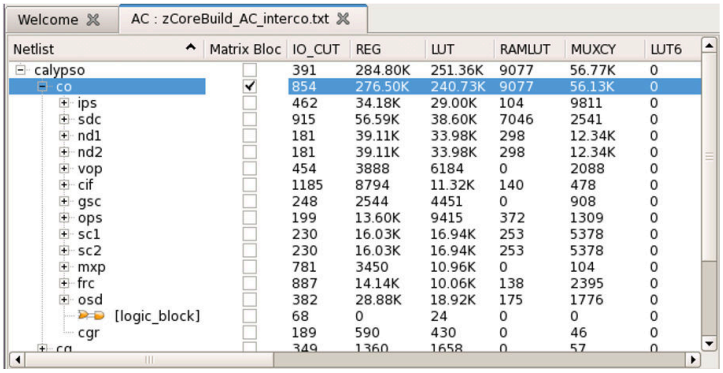
Column	Description
	Black box is already set in the netlist. For more information on black-box management in `zManualPartitioner`, see Managing Black-boxes.
	Black box set in `zManualPartitioner`.For more information on black-box management in `zManualPartitioner`, see Managing Black-boxes.
	Primitives.

Using an Automatic Clustering Interconnection File

The following figure displays the Netlist Tree pane when you load an `Automatic Clustering Interconnection` file in `zManualPartitioner`.

Feedback

Figure 6 Netlist Tree Pane (Automatic Clustering Interconnection File)



The following table describes the columns available in the Netlist Pane when you load `Automatic Clustering Interconnection` file in `zManualPartitioner`.

Table 4 Netlist Tree Pane Column Description (Automatic Clustering Interconnection File)

Column	Description
Netlist	Displays the list of the following available items in the netlist organized in a tree view:Top of a `designLogic blocksCommon` instances followed by their module name
Matrix Block	A check box that allows you to select an item to display in the **Instance Interconnection Matrix** pane. Only one item can be selected at a time.
IO_CUT	Displays the number of inputs/outputs.
REG	Displays the number of registers.
LUT	Displays the number of LUTs.
RAMLUT	Displays the number of RAMLUTs.
MUXCY	Displays the number of MUXCYs.
LUT6	Displays the number of LUT6s.
BRAM	Displays the number of BRAMs.
DSP	Displays the number of DSPs.
ZDELAY	Displays the number of zDelays.
ZRM	Displays the number of ZRMs.

Feedback

Partition Pane

When you load a netlist file in `zManualPartitioner`, both the **Netlist Tree** pane and the **Partition** pane display the file.

The **Partition** pane allows you to view the available partitions and the instance placement. The information in the header displayed in this pane is the same as the information in the columns of the **Netlist Tree** pane (see Figure 7 and Figure 8).

Figure 7 Partition Pane (An EDIF File)

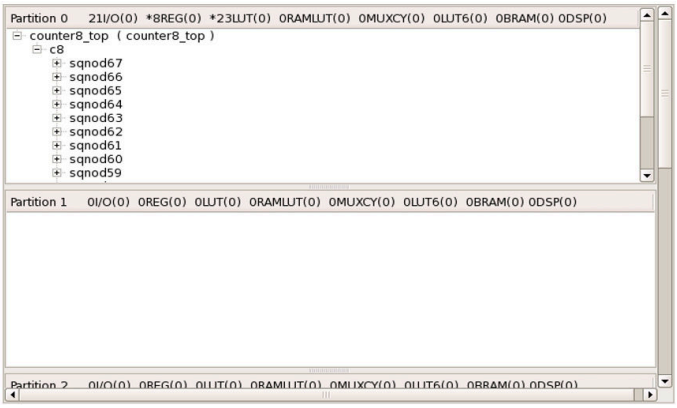


Figure 8 Partition Pane (An Automatic Clustering Interconnection File)

	part_0	part_1	part_2	part_3	part_4	FK/IF	cut
part_0		58/0	57/0	57/0	57/0	391/0	620/0
part_1	58/0		2/0	2/0	2/0	1/0	65/0
part_2	57/0	2/0		0/0	0/0	1/0	60/0
part_3	57/0	2/0	0/0		0/0	1/0	60/0
part_4	57/0	2/0	0/0	0/0		1/0	60/0
FK/IF	391/0	1/0	1/0	1/0	1/0		395/0

A netlist is loaded in Partition 0 by default. However, you may drag and drop instances from one partition to another in the **Partition** pane.

For more information about moving instances between partitions, see Moving Instances Between Partitions.

Adding or Removing Partitions From the Partition Pane

You can add or remove as many partitions as necessary in the **Partition Pane** from the **Properties** dialog box.

To add or remove partitions in the **Partition** Pane, perform the following steps:

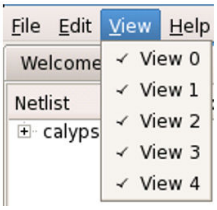
1. From the menu bar, click **Edit** > **Preferences mapping** to open the **Properties** dialog box.
2. Add or remove partitions using the **Number of partitions** field on the **Properties** dialog box as required.

Hiding or Displaying Partitions

You can hide or display as many partitions as necessary in the **Partition** pane by selecting **View** > **View N**.

A view is a partition in the **Partition** pane and a check symbol next to a view means its corresponding partition is displayed, see the following figure:

Figure 9 View Menu



Partition Interconnection Matrix Pane

The **Partition Interconnection Matrix** pane displays information about the number of cuts between partitions in a table. The table (Figure 10) displays one column and one row for each partition.

Using an EDIF File

The following figure displays the **Partition Interconnection Matrix** pane when you load an EDIF file.

Figure 10 Partition Interconnection Matrix Pane (EDIF File)

	part_0	part_1	part_2	part_3	part_4	FK/IF	cut
part_0		58/0	57/0	57/0	57/0	391/0	620/0
part_1	58/0		2/0	2/0	2/0	1/0	65/0
part_2	57/0	2/0		0/0	0/0	1/0	60/0
part_3	57/0	2/0	0/0		0/0	1/0	60/0
part_4	57/0	2/0	0/0	0/0		1/0	60/0
FK/IF	391/0	1/0	1/0	1/0	1/0		395/0

Each cell displays information such as, for instance, 4/0 or 9/0. These figures have the following meaning:

- The first figure is the number of cuts between partitions.
- The second figure is the maximum number of cuts allowed by the target configuration between partitions.

Using an Automatic Clustering Interconnection File

The following figure displays the **Partition Interconnection Matrix** pane when you load an *Automatic Clustering Interconnection* file.

Figure 11 Partition Interconnection Matrix Pane (Automatic Clustering Interconnection File)

	part_0	part_1	part_2	part_3	part_4	FK/IF	cut
part_0		58/0	57/0	57/0	57/0	391/0	620/0
part_1	58/0		2/0	2/0	2/0	1/0	65/0
part_2	57/0	2/0		0/0	0/0	1/0	60/0
part_3	57/0	2/0	0/0		0/0	1/0	60/0
part_4	57/0	2/0	0/0	0/0		1/0	60/0
FK/IF	391/0	1/0	1/0	1/0	1/0		395/0

Each cell displays information such as, for instance, 57/0 or 58/0. These figures have the following meaning:

- The first figure is the number of cuts between partitions.
- The second figure is the maximum number of cuts allowed by the target configuration between partitions.

The **Cut** column displays the cut between the current partition and the others.

Depending on the results of the automatic clustering, the column FK/IF may be displayed to give information on the interconnection between the current partition and the external interface IF.

Instance Interconnection Matrix Pane

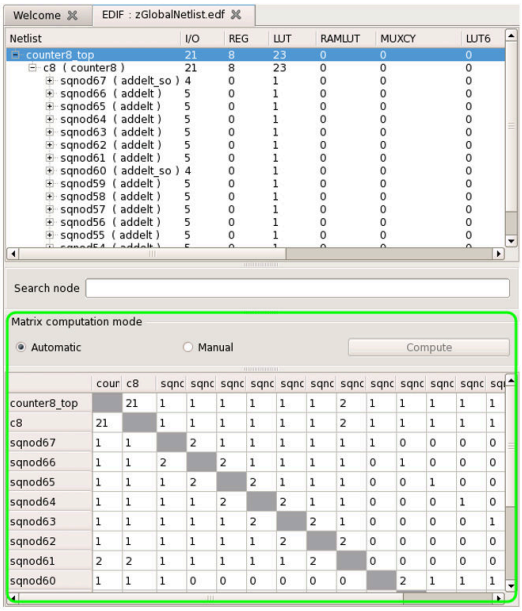
The **Instance Interconnection Matrix** pane computes the interconnection between adjacent instances.

Feedback

Using an EDIF File

The matrix consists of one column and one row for each instance selected, using the check box in the **Matrix Selection** column in the **Netlist Tree** pane (see Figure 12). You can select as many instances, as needed.

Figure 12 Instance Interconnection Matrix Pane (An EDIF File)



The **Matrix Computation Mode** allows you to control how the instance interconnection is calculated:

- **Automatic**: It is automatically calculated every time you must select an instance in the **Netlist Tree** pane.
- **Manual**: You first select all instances from the **Netlist Tree** pane and then click **Compute** for manual calculation.

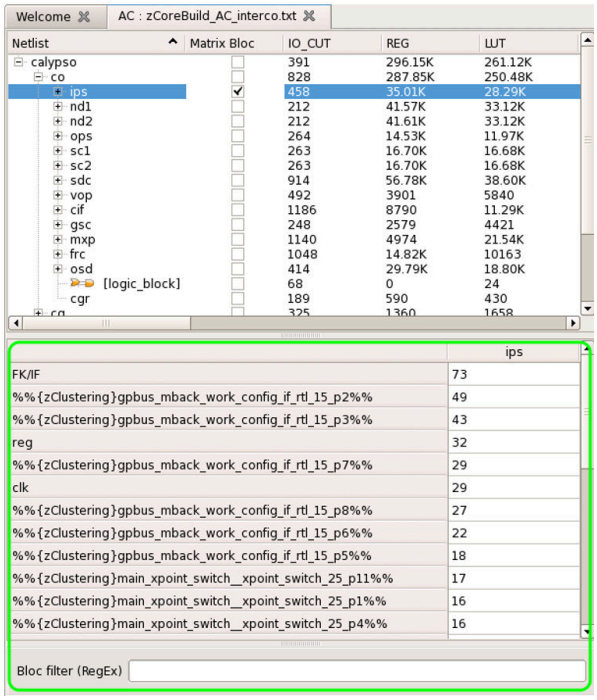
Using an Automatic Clustering Interconnection File

When you load an *Automatic Clustering Interconnection* file, only the interconnection of the selected instance (through the **Matrix Block** check box) and its neighbors can be displayed in the **Instance Interconnection Matrix** pane.

To display the interconnection of an instance, go to the **Netlist Tree** pane and select the check box in the **Matrix Block** column (see Figure 13).

Feedback

Figure 13 Instance Interconnection Matrix Pane (Automatic Clustering Interconnection File)



When an instance has too many adjacent instances, you can filter them using the **Bloc Filter** (**RegEx**) field.

Timing Analysis Pane

The **Timing Analysis** pane allows you to view the structure of combinational paths, that is, the set of successive instances in the combinational path for each partition.

Note:

The timing analysis feature is only available when you use an Automatic Clustering Interconnection file in `zManualPartitioner`.

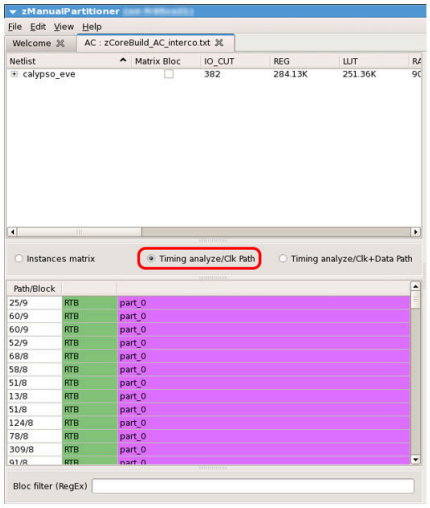
With a clear view of these paths, you can enhance performance by selecting long combinational paths across a minimum number of partitions.

Feedback

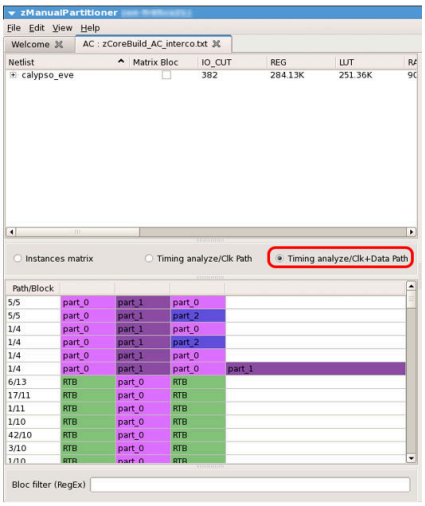
To display the **Timing Analysis** pane, perform the following steps:

1. Open an `Automatic Clustering Interconnection` file in `zManualPartitioner`.
2. Click **Edit** > **Timing Analyze** to replace the **Instance Interconnection Matrix** pane with the **Timing Analysis** pane. By default, it is set on the **Timing Analyze/Clk Path view** (see Figure 14):

Figure 14 Timing Analysis/Clock Path View



3. The **Timing Analyze/Clk+Data Path view** is another view available that displays combinational paths for the data and clock, see the following figure:



The following table displays options available in the **Timing Analysis** pane.

Feedback

Table 5 Available Options in the Timing Analysis Pane

Option	Description
Instances Matrix	Reverts to the Instance Interconnection Matrix pane.
Clock Path View	Displays the combinational path for the clock.
Clock + Data Path View	Displays the combinational path for the data and clock.
Path/Block	The Path is the number of paths represented by the set of path elements.The Block is the number of blocks in the path.

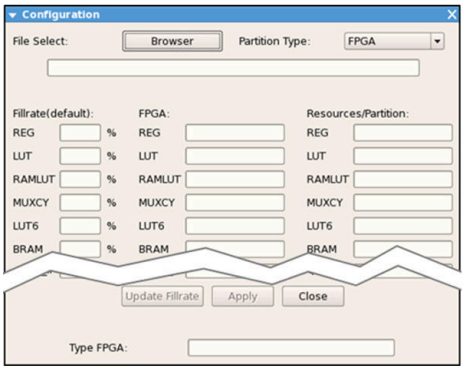
For more information about optimizing timing, see Performing Timing Analysis.

Configuration Window

The `zManualPartitioner` GUI also allows you to select a sample configuration file as a target to map a design with your customized parameters.

From the menu bar, click **Edit** > **Configuration** to open the **Configuration** window as follows:

Figure 15 Configuration Window



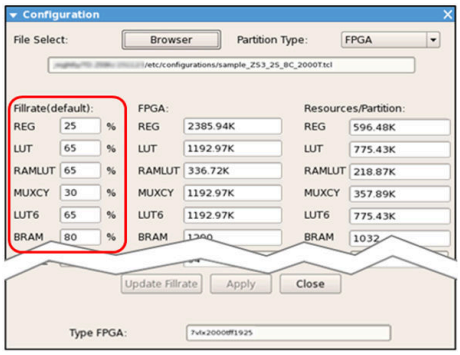
Loading a Sample Configuration File

From the **Configuration** window, you can select a specific configuration file from the `$ZEBU_ROOT/etc/configurations` directory.

To select a target, click **Browser** and select your configuration file. The following figure displays the **Configuration** window loaded for an example configuration file.

Feedback

Figure 16 Example of Sample Configuration File Loaded



Once a sample configuration file is loaded, you can perform the following actions:

- View various parameters at once.
- Modify filling rates for each resource type.
- See the immediate impact on FPGAs and partitions.

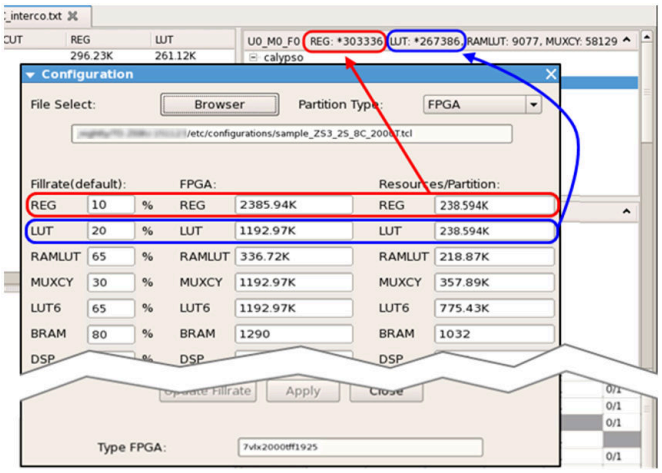
Modifying Filling Rates

To modify filling rates in the Configuration window (see Figure 17), perform the following steps:

1. Modify the resource percentage as required in the **Fillrate (default)** column.
2. Click **Update Fillrate** to modify the values in FPGA and **Resources/Partition** columns.
3. Click **Apply** to validate your parameters.

Feedback

Figure 17 Modifying Filling Rates in Configuration Window



Note:

If your set of parameters are not sufficient to map your design, an asterisk is displayed next to the impacted resource, as highlighted in Figure 17 .

Properties Window

The **Properties** window allows you to set various parameters of your `zManualPartitioner` environment.

To open the **Properties** window, click **Edit** > **Preferences mapping**.

The following table displays available options in the **Properties** window.

Table 6 Available Options in the Properties Window

Option	Description
Number of partitions	Specifies the number of partitions to display in the **Partition** pane.
Generate log file	Defines the path for the log file containing all user actions.
Information	Defines the columns to display/hide in the **Netlist Tree** pane.

Loading and Generating Mapping Files

This section consists of the following sub-sections:

Generating a Mapping File

To retrieve partitioning results into a mapping file, perform the following steps:

1. Click **Edit** > **Generate mapping**.
2. Type a name for the mapping file and click **Save**.

The generation of the mapping file can also be performed with a Tcl command.

Loading a Mapping File

You can load a previously saved mapping file into `zManualPartitioner`. To do so, perform the following steps:

1. Click **Edit** > **Load mapping**.
2. Browse to your mapping file and click **Open**.

   Note:

   You can load a mapping file only with an Automatic Clustering Interconnection file.

2

Managing Black-boxes

In `zManualPartitioner`, you can manage black boxes in the following ways:

- View black boxes originally defined in the netlist (viewed as )
- Add or remove the black box attribute to any item in the netlist (viewed as )

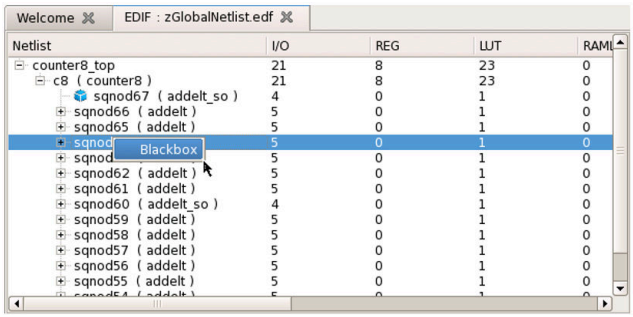
  Note:

  You can manage black boxes only with EDIF files.

Applying a Black Box Attribute

To apply a black box attribute on an item in the netlist, right-click on it and select **Blackbox**.

Figure 18 Applying a Black Box Condition

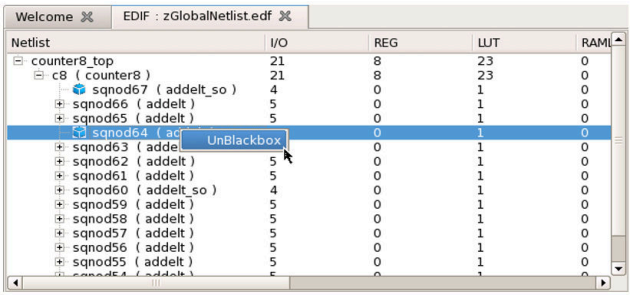


Removing Black Boxing Attribute

To remove a black box attribute and revert an item to its original state, right-click the black box attribute and click **UnBlackbox**.

Feedback

Figure 19 Removing a Black Box Condition



3

Moving Instances Between Partitions

When you load a netlist file in `zManualPartitioner`, both the **Netlist Tree** pane and the **Partition** pane display the file. By default, the netlist is loaded in Partition 0. However, you can drag and drop instances from one partition to another in the **Partition** pane.

This section consists of the following sub-sections:

Using an EDIF File

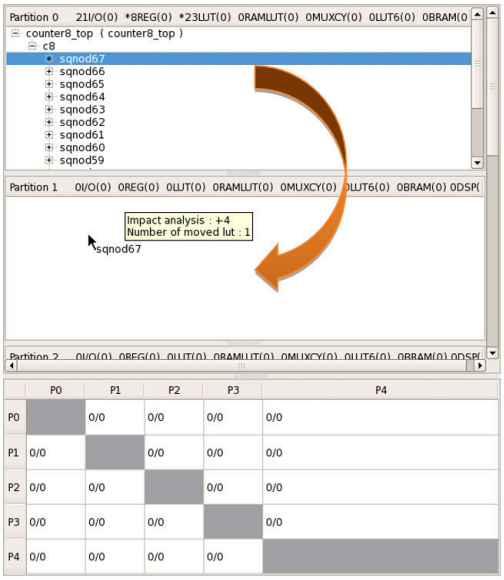
To move an instance to another partition when you load an EDIF file, perform the following steps:

1. In the **Partition** pane, browse to your netlist and select the instance you want to move.
2. Drag this instance into another partition (Partition 1, see Figure 20).

The information about **Impact Analysis** and **Number of Moved LUTs** is dynamically displayed in a tool tip.

Feedback

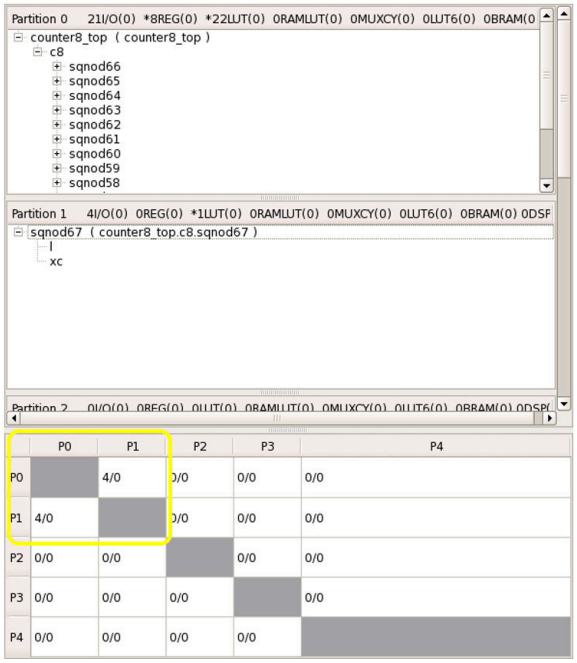
Figure 20 Moving Instances in the Partition Pane Using an EDIF File



1. Drop the instance into the new partition (Partition 1).

The result of moving the instance is displayed in the **Partition Interconnection Matrix** window (see Figure 23).

Figure 21 Result of Moving Instances in the Partition Pane Using an EDIF File



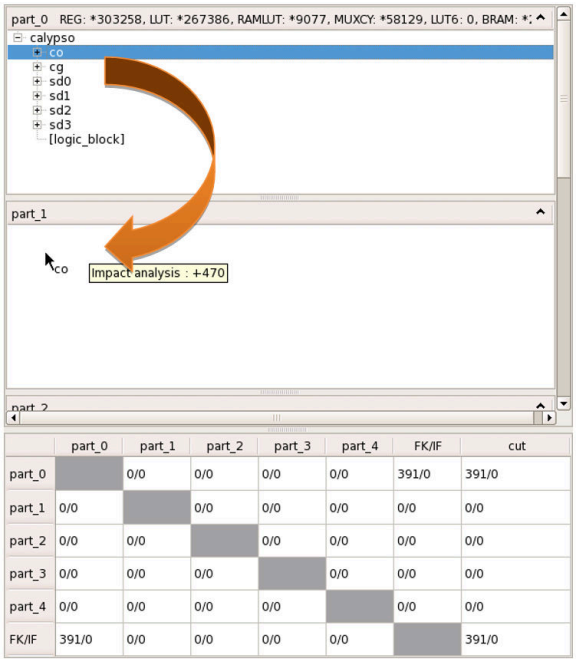
Feedback

For more information on the **Partition Interconnection Matrix**, see Partition Interconnection Matrix Pane.

Using an Automatic Clustering Interconnection File

The process for moving an instance to another partition using an `Automatic Clustering Interconnection` file is same as that for an EDIF file, see Using an EDIF File.

Figure 22 Moving Instances in the Partition Pane Using an Automatic Clustering Interconnection File

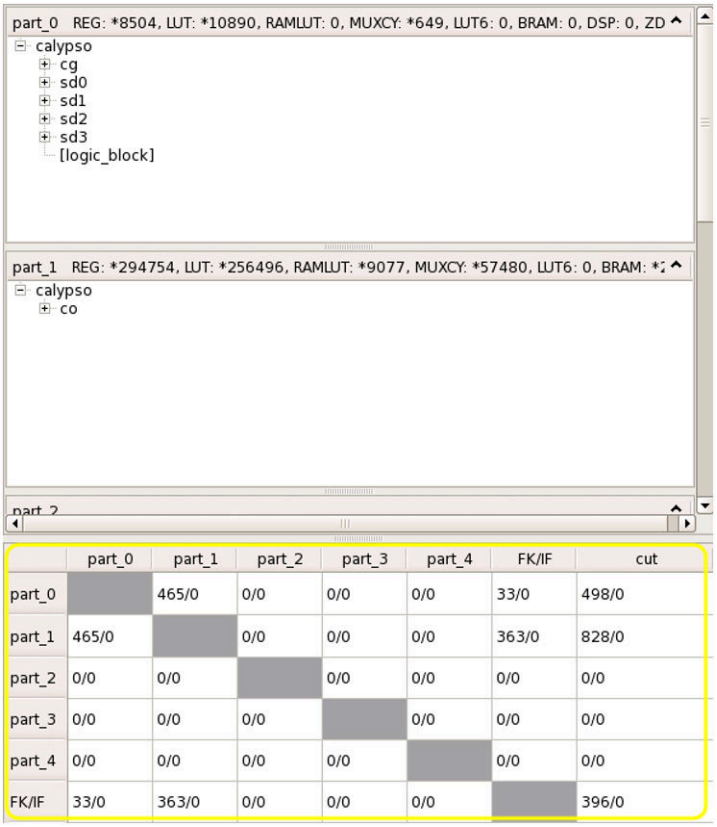


	part_0	part_1	part_2	part_3	part_4	FK/IF	cut
part_0		0/0	0/0	0/0	0/0	391/0	391/0
part_1	0/0		0/0	0/0	0/0	0/0	0/0
part_2	0/0	0/0		0/0	0/0	0/0	0/0
part_3	0/0	0/0	0/0		0/0	0/0	0/0
part_4	0/0	0/0	0/0	0/0		0/0	0/0
FK/IF	391/0	0/0	0/0	0/0	0/0		391/0

The following figure displays result of moving instances in the **Partition** pane using an *Automatic Clustering Interconnection* file.

Feedback

Figure 23 Result of Moving Instances in the Partition Pane Using an Automatic Clustering Interconnection File



	part_0	part_1	part_2	part_3	part_4	FK/IF	cut
part_0		465/0	0/0	0/0	0/0	33/0	498/0
part_1	465/0		0/0	0/0	0/0	363/0	828/0
part_2	0/0	0/0		0/0	0/0	0/0	0/0
part_3	0/0	0/0	0/0		0/0	0/0	0/0
part_4	0/0	0/0	0/0	0/0		0/0	0/0
FK/IF	33/0	363/0	0/0	0/0	0/0		396/0

Performing Timing Analysis

After moving instances across partitions (see Using an Automatic Clustering Interconnection File), you can modify elements of the combinational paths (that is, a set of successive instances that are in the same partition) and their timing to enhance performance.

Note:

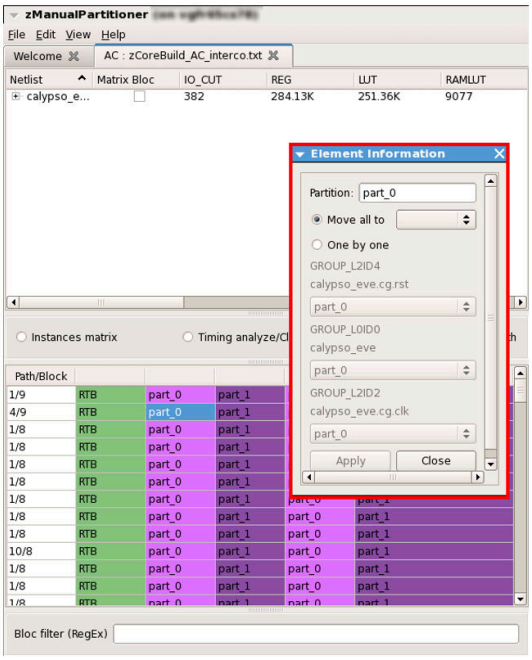
The timing analysis feature is only available when you use an Automatic Clustering Interconnection file in `zManualPartitioner`.

To modify the content of a combinational path, perform the following steps:

1. Click **Edit** > **Timing Analyze** to replace the **Instance Interconnection Matrix** pane with the **Timing Analysis** pane.
2. Select your view by selecting **Timing analyze/Clk Path** or **Timing analyze/Clk + Data Path**.

3. Double-click on the path element you want to move to display the **Element Information** window (see Figure 24).

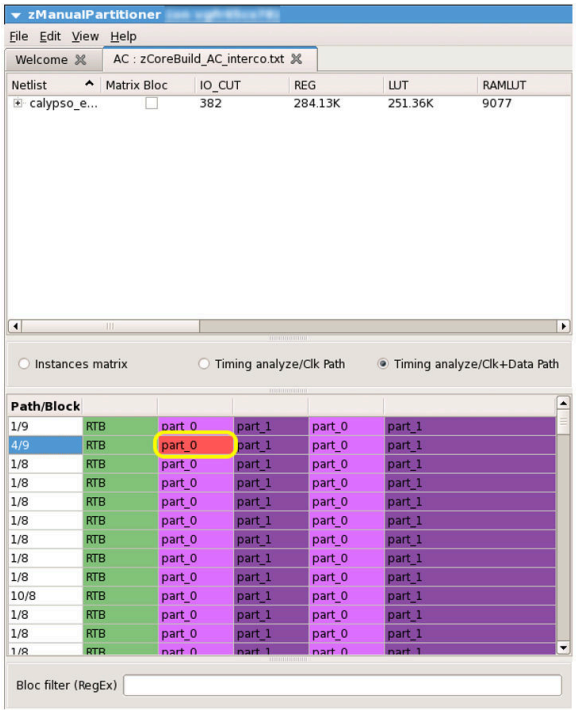
Figure 24 Element Information Window



4. In the **Element Information** window, select the method to move by selecting one the following options:
   - **Move all to**: Moves the entire combinational path content of a partition to another partition. Selects the destination partition from the drop-down next to it.
   - **One by one**: Moves an individual element of a partition to another partition. Selects the destination partition from the drop-down below it.
5. Click **Apply** and then **Close**. The modified element is now in red. It means that it is required to launch timing analysis again (see Figure 25).

Feedback

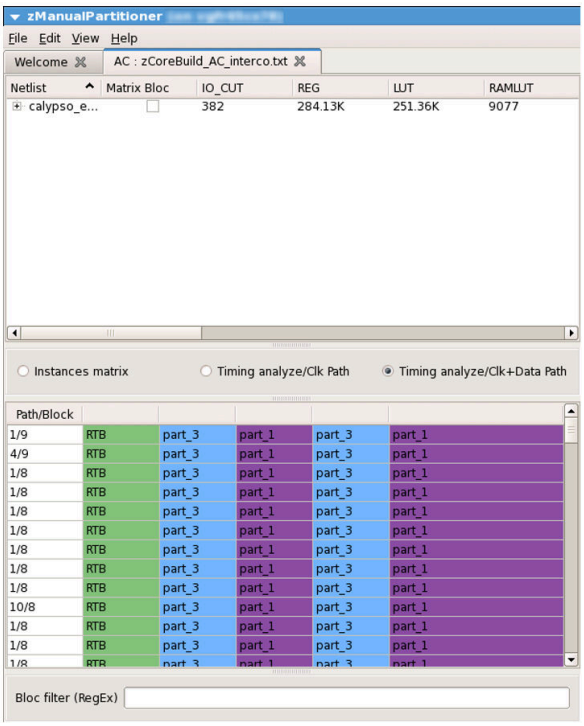
Figure 25 Element Modified in the Timing Analysis Pane



Feedback

6. Select **Edit** > **Timing Analyze** to refresh the **Instance Interconnection Matrix** pane according to your modifications (see Figure 26).

Figure 26 Timing Analysis Pane Refreshed



4

Limitations of zManualPartitioner

The limitation of `zManualPartitioner` is as follows:

- `zManualPartitioner` is not supported by other partitions.
- It may be challenging to partition large EDIF files using `zManualPartitioner`.