Astro Manual

An End User's Guide for
Synopsys Astro in Linux Environment

Application Specific Integrated Circuit Design Laboratory
Chung Yuan Christian University

Document Update History	Author	Date
V1.1	Yu-Kuen Lai	July 2, 2008
V1.0 Initial Release	Alfonso Cesar Albason 亞豐碩	Dec, 2007

Copyright Notice and Proprietary Information

Disclaimer

The Application Specific Design Laboratory makes 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.

Astro Manual

Contents

Chapter 1 Introduction

Synopsys Astro is the Automatic Place and Route (PnR) tool from Synopsys. Aside from PnR, it can also perform Verification and can also generate back-annotation scripts. For more details, see the Astro User Guide included in this CD.

The semantic <*text here*> is used in this document. Any text enclosed with the '<' and '>' signs are just place-holders and should be substituted with a text that has a relevant value in the design. So for example you see a reference to "<your_design>.v", you should substitute it with a filename for your design.

The Astro version used in this tutorial is Z-2007.03-SP3. An Astro license is needed before you can use Astro PnR tool. Also, before you begin with the tutorial, you need to have a netlist output from the synthesis tool (<synthesized_design>.v). For more information on how to synthesize a design, see the Design Vision Manual.

After the PnR process, you should be able to produce two files which can be used for simulation:

- <layout>.v – the hierarchical netlist output from Astro.
- <delay>.sdf – the standard delay file for the layout.

Chapter 2 Place and Route of an OpenRISC[1] Control Unit

This chapter guides you through the place and route process of an OpenRISC 32-bit microprocessor control unit. The Verilog source code can be found at http://www.opencores.org/projects.cgi/web/or1k/overview

Environment Variable

You have to ***source*** an important shell script file before running Astro properly. Please open a x-terminal and type commands shown as follows:

```
[ylai@ee107 ~]$ csh
[ylai@ee107 ~]$ source /usr/cad/synopsys/CIC/astro.cshrc
```

The system will prompt you a message "***Platform = linux***" once everything is set properly.

Starting Astro

Open a terminal and type the following:

```
[ylai@ee101 ~]$ Astro&
```

The Astro application should now be visible on the screen.

[1] http://www.opencores.org/projects.cgi/web/or1k/overview

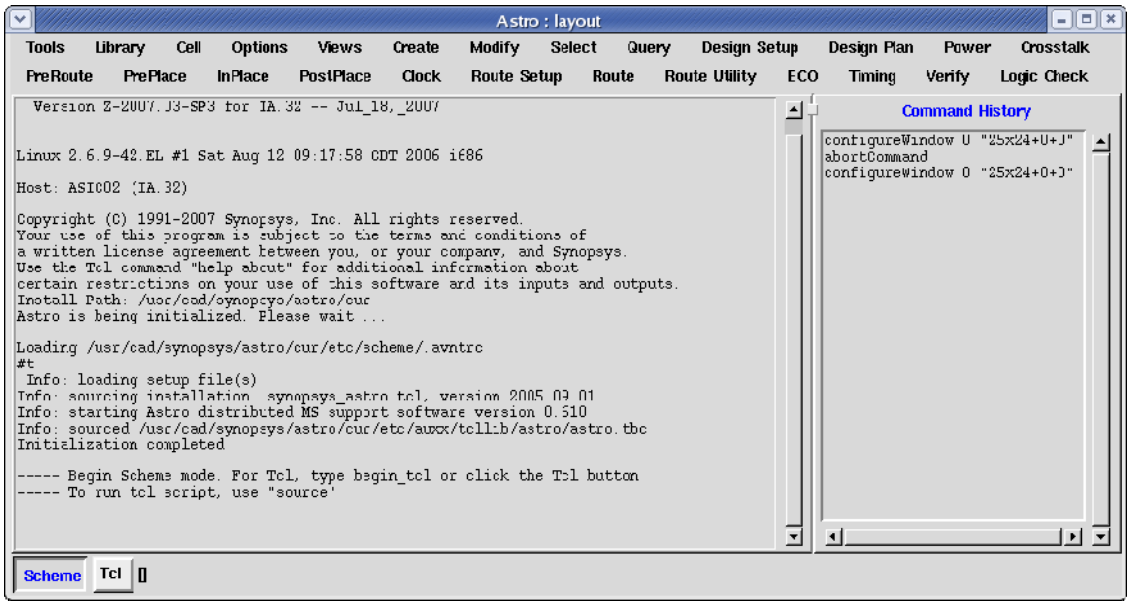


Figure 1 - Astro

Setting Up Reference Library

1. Click **Tools** then **Data Prep.**
2. Select **Cell Library** then **Library Preparation.**
3. Enter a Library Name, for example REFERENCE.
4. Click Prepare **Physical Library.**
5. Click **Select One Format.** You have **TF+GDSII+CLF**, **PLIB**, **PLIB+GDSII**, **LEF**, **TF+LEF**, **LEF+GDSII** as choices for building up the standard cell library. For example you choose, **TF+GDSII+CLF**.
6. Click **Create Library.**
7. Enter a **Library Name, Technology File** and toggle **Set Case Sensitive.**
8. Click **Set Bus Naming Style** and browse for your reference library.
9. Click **Stream In**, browse for GDS files of your standard cells and select your reference library. Enable **Overwrite Existing Cells** and **Confirm Overwrite of Existing Cells.** Enable **Use Layer for Bundary** and put in the GDS layer number for **Boundary Layer**. For **Cell Type Definition File** make a file called standard.map which contains a single command line **gdsStandardCell ***, this sets every streamed in GDS into standard cell type.
10. Click **Set Cell Port Type**. For **Scheme File Name** make a file called standard.port which contains the commands:

    dBSetCellPortTypes “REFERENCE” “*” ‘((VCC” “Inout” “Power”) (“VSS” “Inout” “Ground”)) #f

11. Click **Extract BPV**, then fill in the metal text layer number for the appropriate metal layer. This step creates the FRAM views of your reference library.

12. Click **Set PR Boundary.** Fill in your reference library name and apply to all cells. Specify Unit Tile Width, which can be found in your technology file. Specify cell height, which can be measured in your reference library standard cell. Enable **shared P/G (double back).**
13. Click **Create PDB** then browse for your reference library. Click **Import Logic Model DB**, Click **Select DB,** then browse for your minimum, maximum and typical database. Enable **Overwrite**, this creates the LM view of your reference library to be used by Astro for timing and power parameters.

Setting Up Design Library

The next step is to create the design library. Basically, we will tell Astro that we want a certain file to be the source of our designs. From this file, Astro will then generate the standard cells needed from the Chartered 0.35um in able to implement your design.

1. Click **Tcl** found at the bottom-left of the screen. To start the creation of the design library, issue the following command in Astro shell:

```
Astro> auVerilogToCell
```

   A window similar to Figure 2 should pop up.

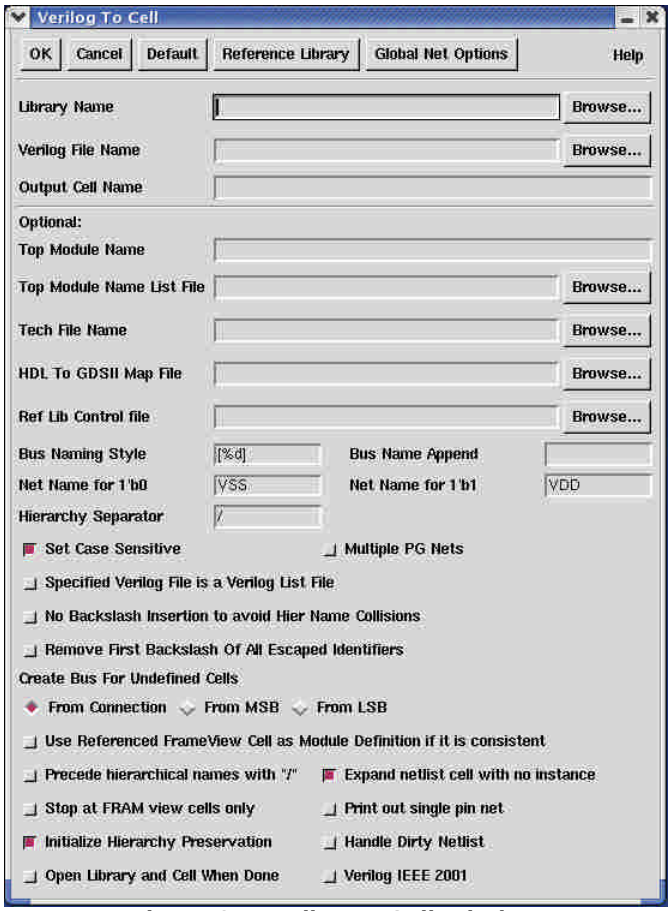


Figure 2 – Verilog to Cell Window

2. Fill in the following information in the corresponding fields:

   Click **Browse...** in the **Library Name** field and browse to the /astro/layout/design directory as shown in Figure 3. Click **design** and then click **Hide**. The path to the library will now appear on the **Library Name** field.

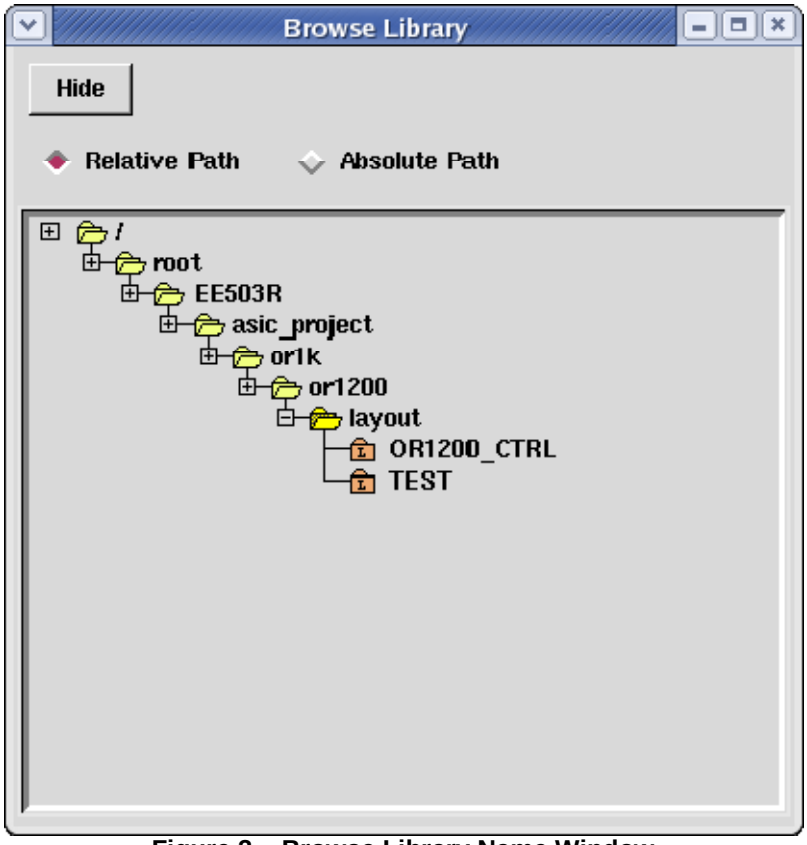


Figure 3 – Browse Library Name Window

Click **Browse...** in the **Verilog File Name** field and look for your saved netlist. Select the file and click **OK**.

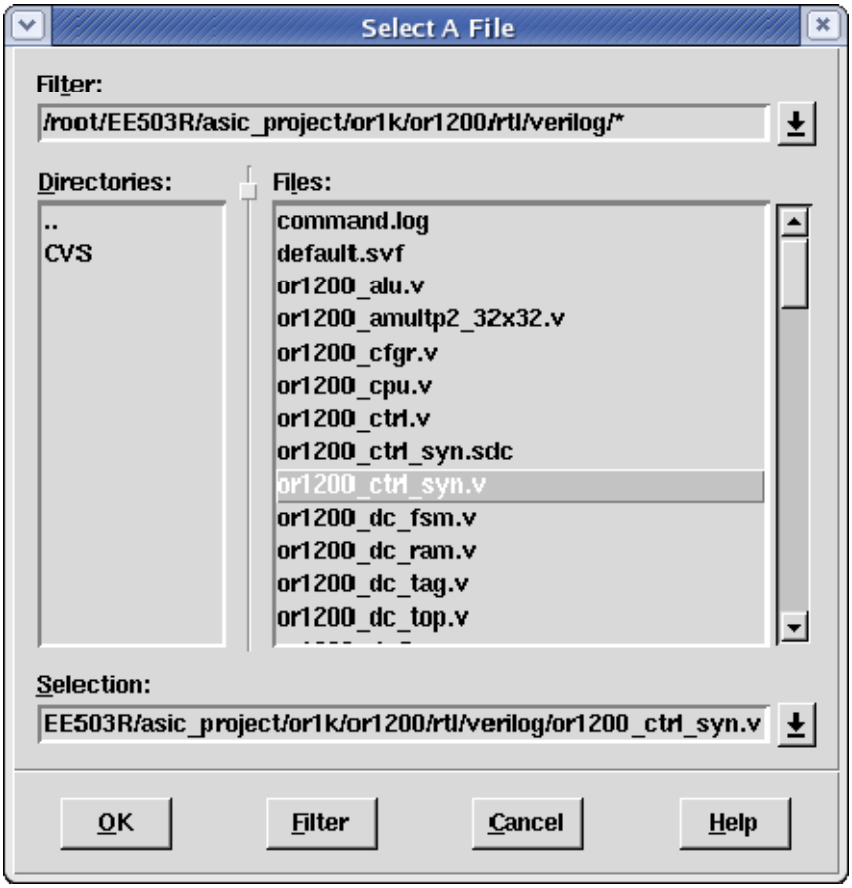


Figure 4 – Browse Verilog File Name Window

In the **Output Cell Name** field, type a label. In this example, we will use **or1200_ctrl_syn**.

In the **Top Module Name** field, enter the name of your topmost module. The topmost module's name of our example is **or1200_ctrl**.

Click **Browse…** in the **Tech File Name** field and browse to the /usr/cad/library/chartered35/tech/ directory. Select the file **charter35m3.tf** and click **OK**.

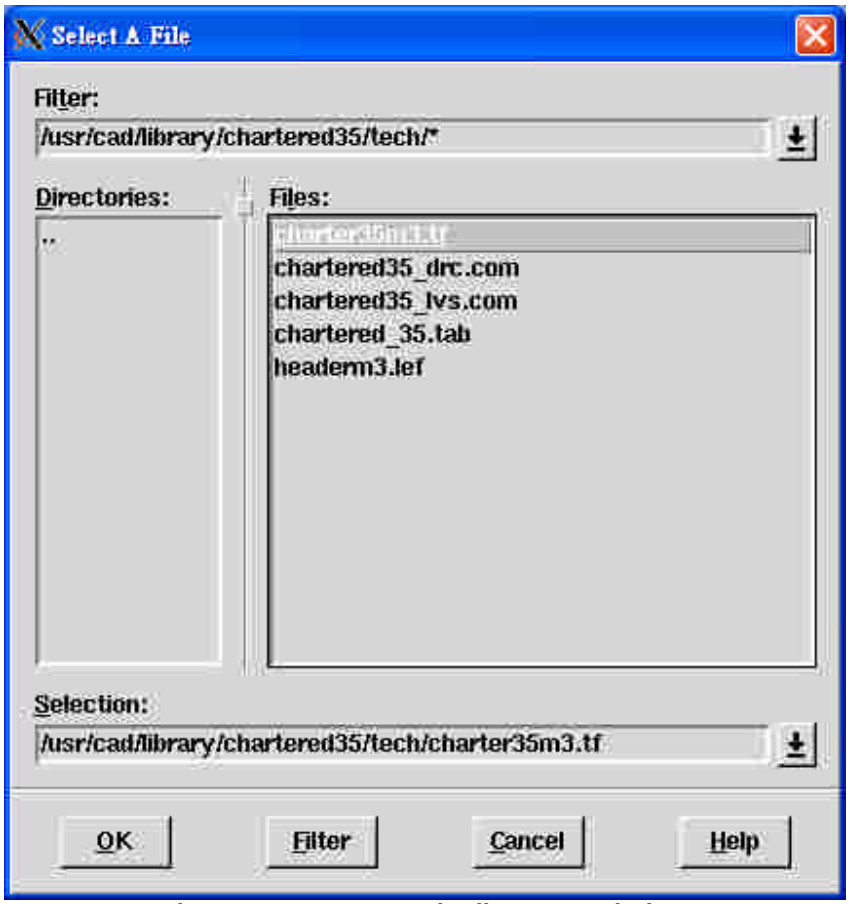


Figure 5 – Browse Tech File Name Window

3. Click the **Reference Library** button to specify the reference libraries for your designs and a window will pop up.

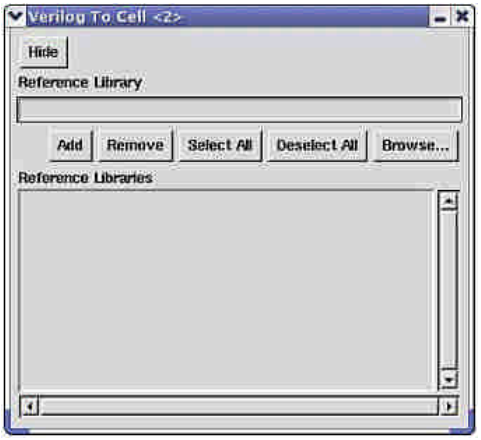


Figure 6 – Reference Library Window

Click **Browse…** and another window will pop up. Look for the 'testing' library.

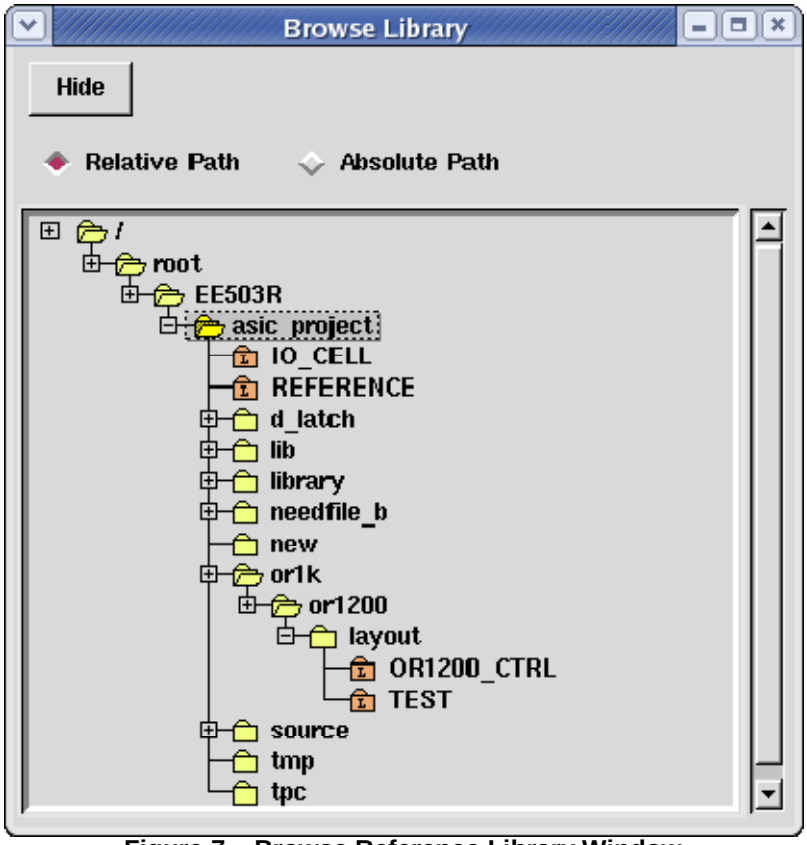


Figure 7 – Browse Reference Library Window

Click the **testing** library then click **Hide**. The window will close.

Click **Add** in order to add the reference library to the list, then click **Hide** to close the window.

4. Click **Global Net Options** to specify the wire names for VDD and VSS/GND connections.

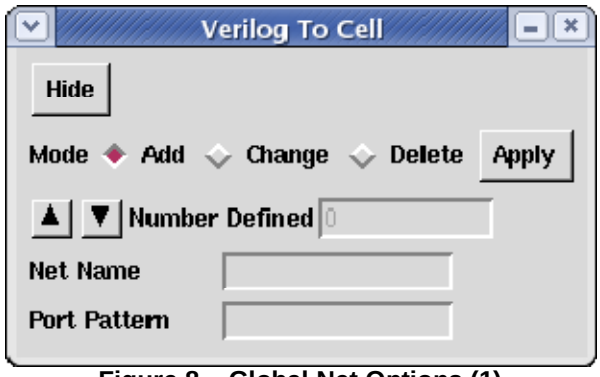


Figure 8 – Global Net Options (1)

Type the following in the corresponding fields:

Net Name: VCC
Port Pattern: VCC

Click **Apply**.

5. Change the VDD fields to VSS, and click **Apply** again.

Figure 9 – Global Net Options (2)

The **Number Defined** field should now have a value of 2 since you have defined two wire names. Click the **Hide** button to return to the previous window.

6. Click **OK** to start the creation of the Design Library. Astro should output a message saying that the conversion was a success.

Notes:

- *Tech file is short for technology file. This file contains information of standard cells of a specific fabrication process. Information in the tech file includes (but not limited to) delay information of cells, area information of cells, input and output capacitance models of cells.*
- *A reference library is created from a tech file, and they contain information from the tech file converted in such a way that other Synopsys programs can use it. So for example, we want out design to be fabricated at ABC Company, then we must create a reference library based on the technology file provided by the ABC Company. In this tutorial, our reference library is created from the technology file of the Chartered Semiconductor Manufacturing Company 0.35um CMOS process, or more commonly known as charter 0.35um process.*
- *The purpose of creating a design library is to let Astro determine all the standard cell (from the reference library) needed in your design.*

Floor Planning

1. We will now open the CEL view of your design. Go to **Library → Open** and a small window similar to the window on the left on 錯誤! 找不到參照來源。 will appear.

   Click **Browse…** and another window will appear (right window on 錯誤! 找不到參照來源。7. Browse to the **design** library. Click **design**, and then click **Hide**. For most of the time, the library is already open so you can skip this step. Nonetheless, you should do this step just to check if the library is already open. Click **OK**.

2. Go to **Cell → Open**, then **Browse…** on the window that appears and another window will pop up. Click the name of the cell that you converted. In this tutorial, we used **adder_synth** as the name of our CEL view. Click **Hide**.

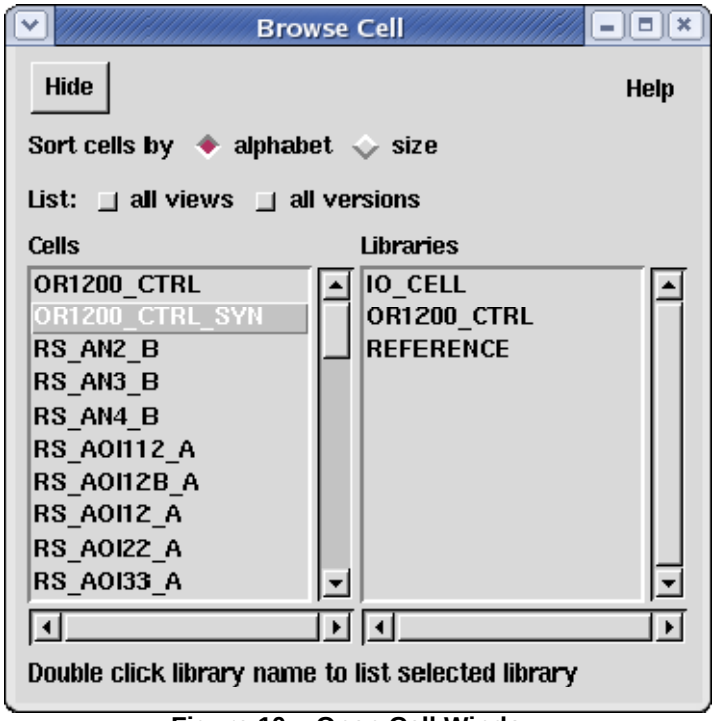


Figure 10 – Open Cell Window

After clicking **OK**, a window similar to Figure 11 should appear.

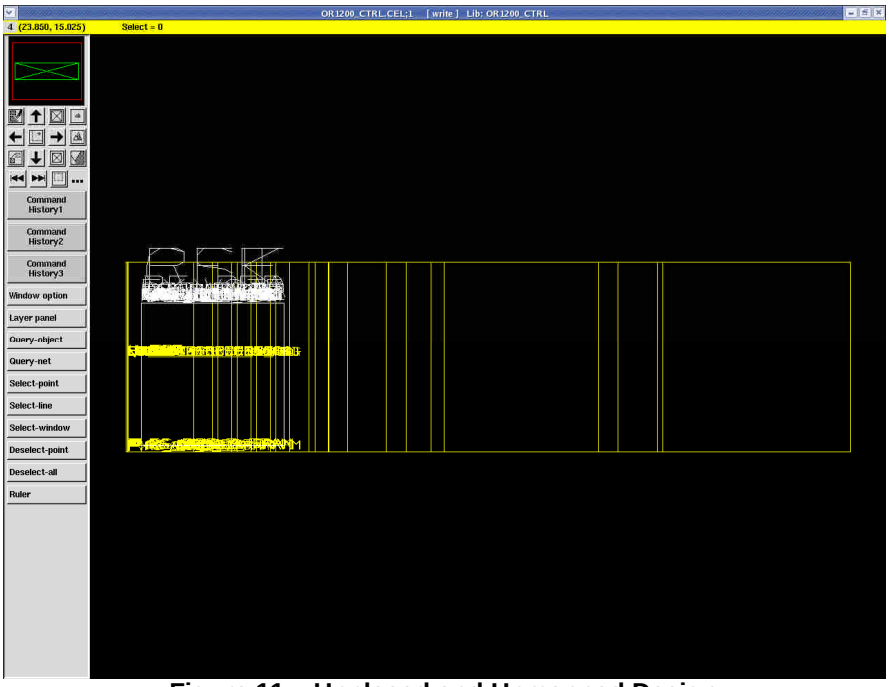


Figure 11 – Unplaced and Unmapped Design

As you can see, the design is a little chaotic because all of the standard cells are still overlapping since Astro still does not have any information on how we want our design to be routed.

3. The Floor Planning process will now segregate the standard cells and will also allow you to automatically allocate an area that will fit nicely with your design. To invoke the Floor Planner tool of Astro, issue the following command in Astro shell:

```
Astro> axgPlanner
```

In the window that appears, fill in the following fields with the corresponding values:

Core Utilization: 0.7
Row/Core Ration: 1
Core Aspect Ratio (H/W): 1
Core to Left/Right/Top/Bottom: 70
Horizontal Row, Double Back, Start First Row, Flip First Row: Enabled
Pin Snap: Enabled

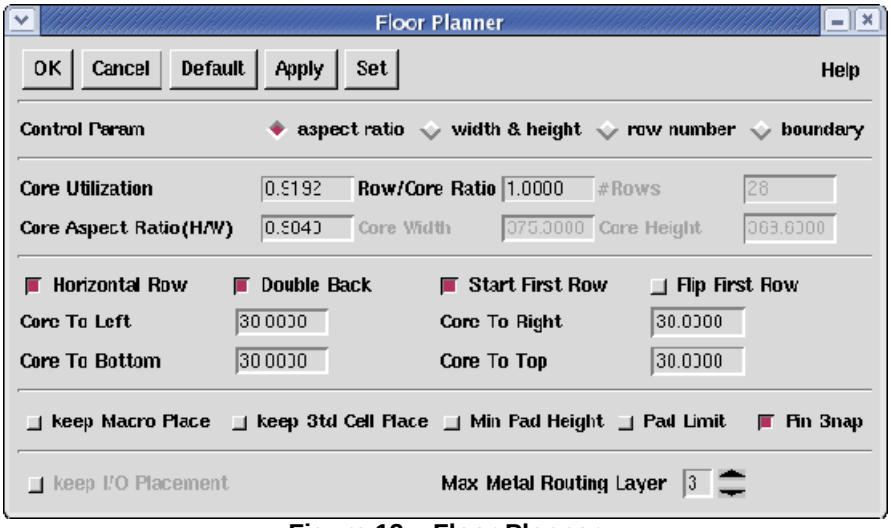


Figure 12 – Floor Planner

Click **OK**.

If you look at the CEL view of your design, you should see that it is starting to get organized – the core area of the chip (the red rectangle) has been defined, and the standard cells are segregated (the small yellow rectangles). If you try to zoom in at an individual standard cell (one yellow rectangle), you will see what part of your design it is, and also what kind of cell.

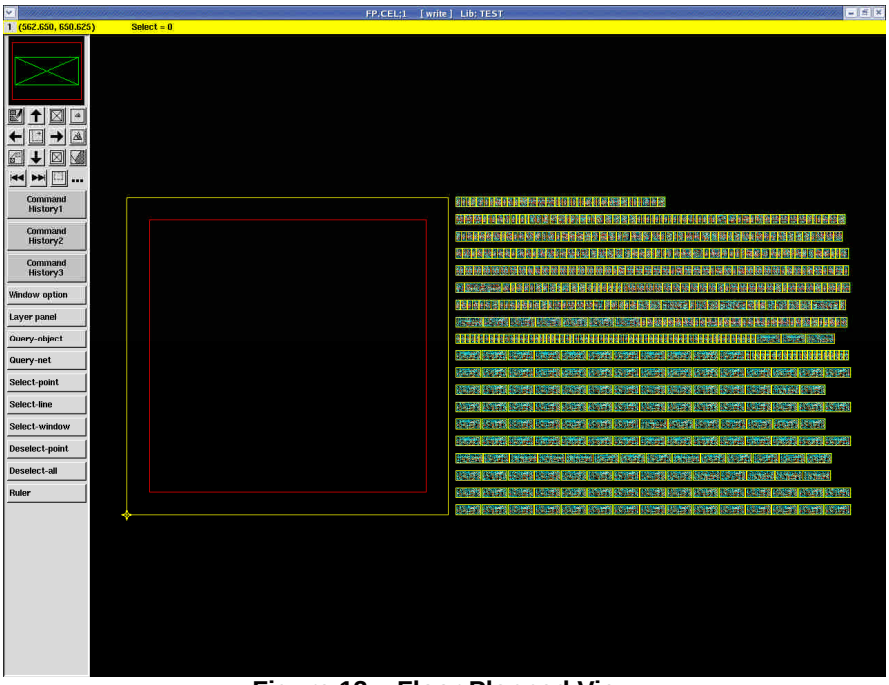


Figure 13 – Floor Planned View

4. Try to play with your design – zoom in and zoom out – you can even see the actual pins of your chip by zooming in at the boundaries of the core area of your chip.

Creating Rectangular Rings

Now that the design has been floor planned and the chip area has been defined, we will now create the Power Rings. These rings are where the standard cells get their VDD and VSS supplies.

1. Issue the following command in Astro shell and a window will appear:

```
Astro> axgCreateRectangularRings
```

2. Enter the following parameters in the corresponding fields:

   Around: Core
   Net Name(s): VCC, VSS
   L-Width, R-Width, B-Width, T-Width: 10
   L-Layer, R-Layer: 16
   B-Layer, T-Layer: 14
   Offsets: Applied After Automatic Adjustments
   Extend: Check None

   The window should now look like Figure 14.

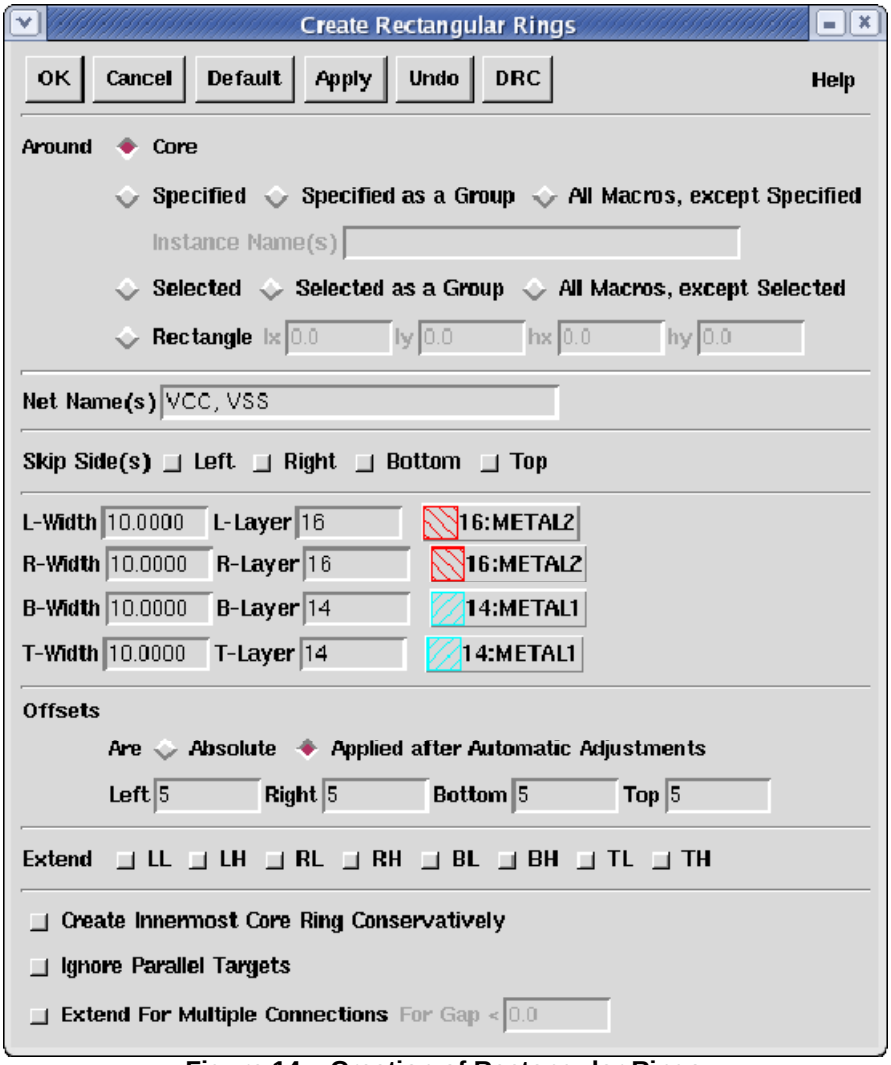


Figure 14 – Creation of Rectangular Rings

Click **OK**.

You should now see two rectangular rings (corresponding to VDD and VSS) around your chip area. Usually, Metal 1 layer (blue-green) is used in horizontal routings while Metal 2 layer (red) is used in vertical routings. You can check the layer coloring by clicking the **Layer panel** button in the CEL view.

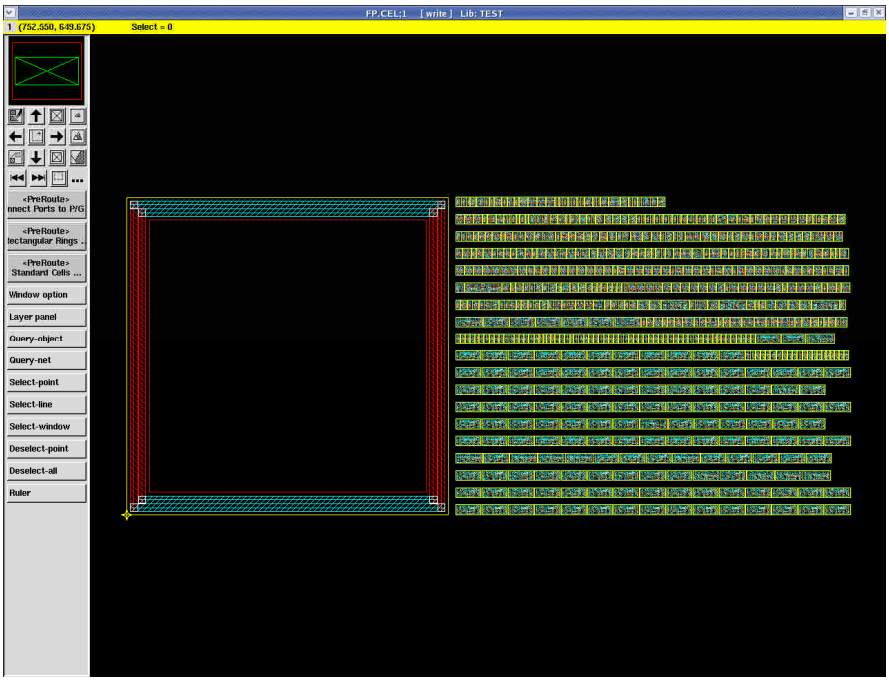


Figure 15 – Rectangular Rings Created

Caution: The yellow color for the boundary of the standard cells has nothing to do with the Metal layer so you should never generalize that standard cells are created using Metal 2 layers.

Automatic Placement

Now that we have created rectangular rings, we are now ready to place the standard cells inside the chip area. Astro provides automatic placement of standard cells.

1. Issue the following command in Astro shell:

```
Astro> astAutoPlace
```

2. Click **OK** in the window that appears. If your design is big, it should take a while to place the standard cells inside the chip area. After the placement process is finished, you should now see that the cells are placed inside the chip area. Again, feel free to play with your design – try to see which module is on the center of the chip and things like that.

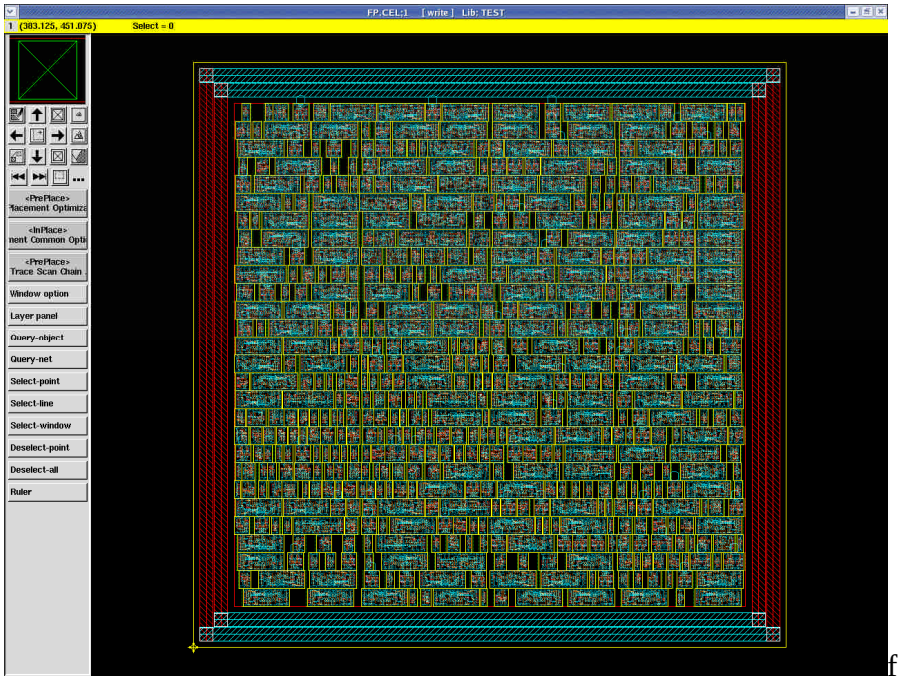


Figure 16 – Automatic Placement Done

Tools > Astro
InPlace > Placement Common Options

Optimization Mode	Congestion, Timing
No Cells under Preroute	M1, M2, M3
All other options	Default Value

Pre-Place > Pre-Placement Optimization
Default. OK.

PreRoute > Connect Ports to P/G

Net Name	VDD
Port Pattern	VDD
Cell Master Pattern	.*
Cell Instance Pattern	.*
Net Type	Power
Net SubType	Core
Update Tie Up/Down	Disable
Mode	Connect
Create Missing Ports	Enable

Apply

Prerouting

This step is performed before global routing so that the global router can recognize potential routing obstructions. You should not skip this step or else your final design will yield DRC and/or LVS errors.

1. Type the following in Astro:

```
Astro> axgPrerouteStandardCells
```

2. Click **OK** in the window that appears (Figure 17). You should see a message saying that prerouting was a success.

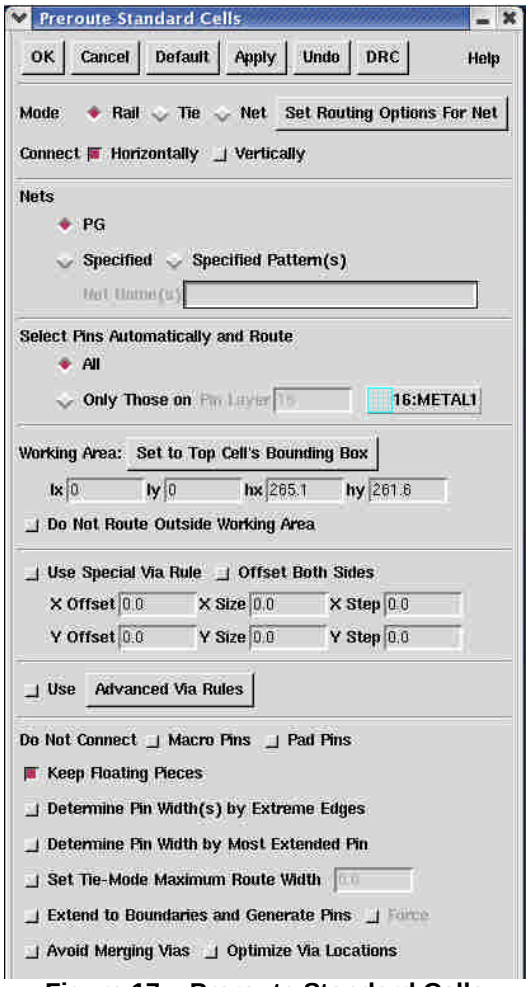
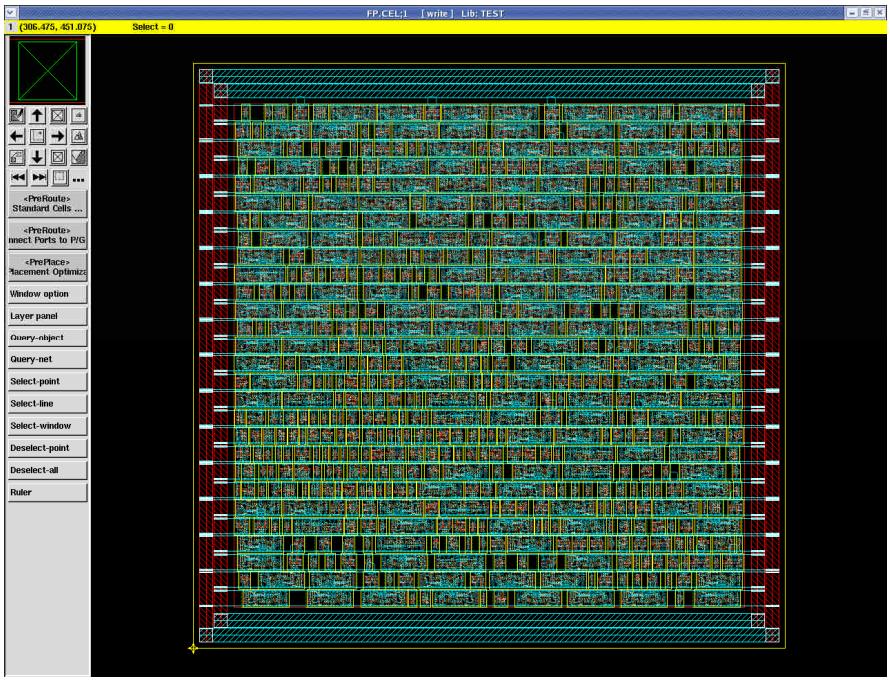


Figure 17 – Preroute Standard Cells



Global and Detailed Routing

The design is now ready to be routed. During routing, Astro estimates the wire tracks needed for each cell so that necessary counter measures can be made when detailed routing is done.

1. Issue the following command in Astro:

```
Astro> axgAutoRoute
```

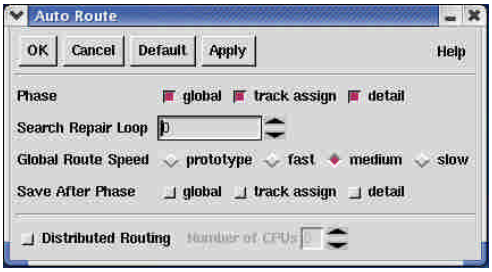


Figure 18 – Auto Route Window

2. Click **OK** to proceed. While your design is being routed, you should see status messages scrolling down in the Astro shell. This messages can provide useful information like the total wire length used in your design.

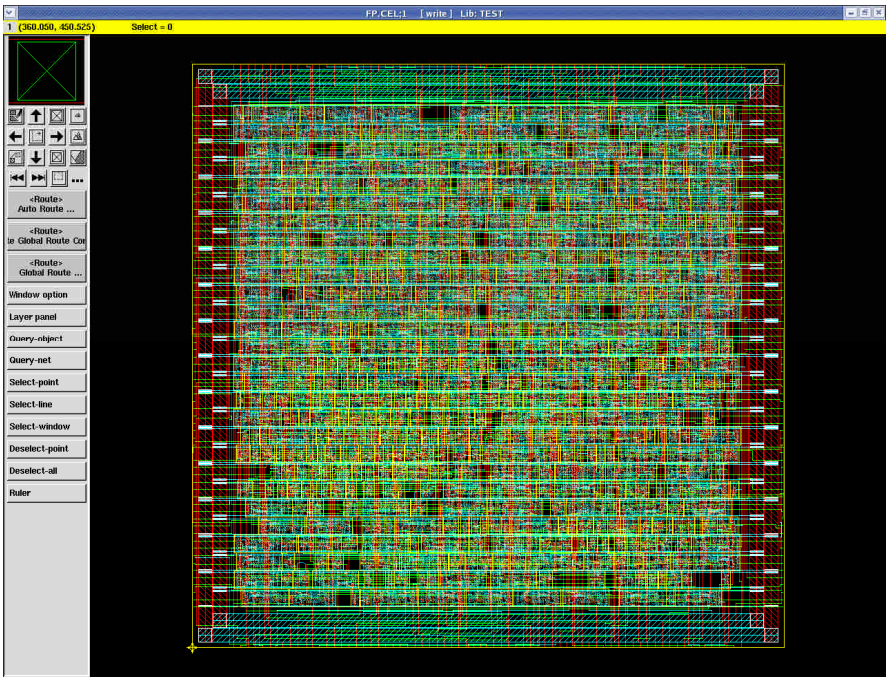


Figure 19 – Routing Complete