EDI System Workshop 1

What you will learn

- Setting up the EDI System environment
- Importing the design
- Using the Design Browser
- Learning floorplanning and Editing floorplan objects
- Planning Power/Ground stripes and rings
- Routing Power/Ground to Power/Ground pins
- Running Placement
- Running Trial Route and viewing congestion
- Extracting RC and generating simulation files
- Calculating delays and generating SDF file
- Building timing graph and generating violation reports
- Fixing setup timing violations by Timing Optimization
- Running clock tree synthesis
- Building timing graph and generating violation reports
- Fixing hold timing violations by Timing Optimization
- Running timing and SI driven routing using NanoRoute

Table of Contents

Introduction

Workshop 1 is a tutorial to walk you through the basic steps in the implementation flow using Encounter Digital Implementation (EDI) System. This is a flat implementation flow which can be applied to chip level designs as well as blocks. If you are interested in performing a hierarchical design flow it is recommended you complete Workshop 2 after completing this tutorial. Lastly, if you're interested in floorplanning for Flip Chip designs it is recommended you complete Workshop 3.

The goal of this tutorial is to provide you a small example of using the EDI System software. It is very basic by design so we highly recommend users attend one of the several EDI System training classes provided by Cadence Educational Services. For more information on available training please visit www.cadence.com and click *Support & Training*.

Leon Design Information

Leon information - The design in this workshop is a Leon processor. The Leon design is a block level design with 35K instances, 4 memories, and 1500 IO pins. The library used is a Cadence Generic 45nm library using 9 routing layers.

Workshop overview - Virtual prototyping is used to quickly determine the feasibility of the Leon design. The design inputs are the netlist, floorplan, clock sources, and timing constraints. The process/technology inputs are the physical libraries, timing libraries, and process technology libraries. The design is:

- Imported,
- Floorplanned
- Placed
- Trial routed
- Wire parasitic extracted
- Setup timing analyzed
- Timing optimized for setup timing
- Clock tree synthesized
- Hold time analyzed
- Timing optimized for close timing
- NanoRoute routed
- Signal integrity analyzed
- Fix violating noisy nets

Setting up EDI System and the Work Directory

The following are the instructions to setup and use the EDI System design tool. You need to download and install the software from http://downloads.cadence.com. Then download the tutorial database from http://support.cadence.com/wps/mypoc/cos?uri=deeplinkmin:DocumentViewer;src=wp;q=Training/Digital_IC_Design/tutorialDatabaseEDI13.1.tgz.

Setting the EDI System environment –

In a Linux window (shell tool, xterm, or etc.), you must set a path to the installation directory and set an environment variable for the *cdslmd* license:

```
linux% set path=(<install_dir>/tools/bin $path)
linux% setenv LM_LICENSE_FILE 5280@lic_hostname
linux% which encounter (to verify the path is set
properly)
```

Note that the Linux window you invoke EDI System from becomes the EDI System Console where EDI System text commands are entered and EDI System messages are output.

Setting up your work directory for the workshop –

To setup the workshop directory extract the tutorial database as follows:

```
linux% tar xfz tutorialDatabaseEDI13.1.tgz
linux% cd edi13.1_tutorial
linux% ls
DBS/ DATA/ LIBS/
```

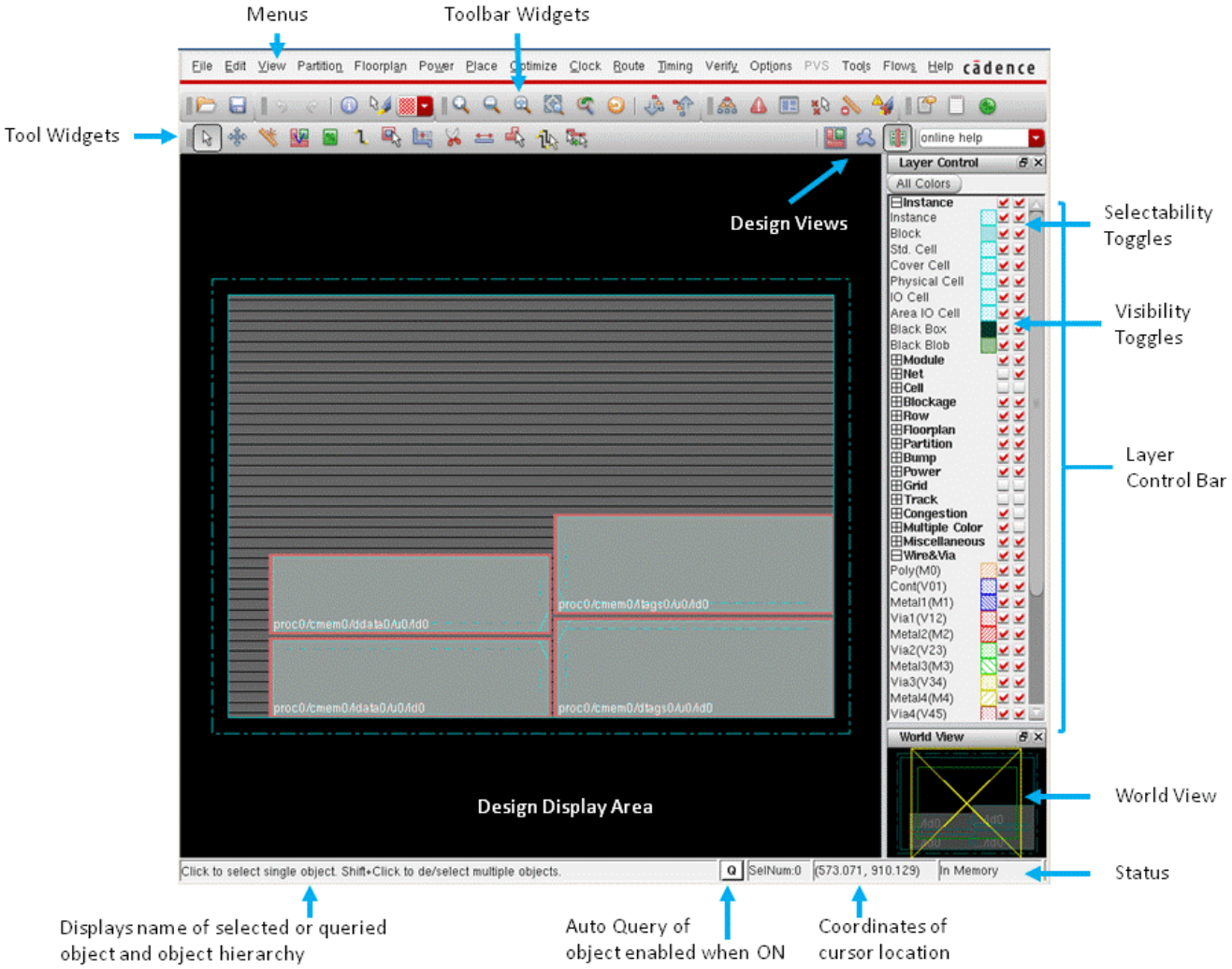
You are now ready to start the tutorial.

Starting an Encounter Session

Make sure you are in the *edi13.1_tutorial* work directory and invoke EDI System:

```
linux% encounter
```

Now the EDI System Main Window displays.



EDI System General Mouse Usage

Use the Left Mouse Button (LMB) to perform the action of the active icon. The **Select Object** icon  is active by default.

Each of the three design views allows certain objects to be selectable. The following table shows in general which objects can be selected for each view by default:

View	Selectable Objects
Floorplan	Modules, blocks, and floorplan objects.
Amoeba	Modules and blocks
Physical	Blocks, instances, and nets

Use the icons shown below to switch between the above views:

- Use the **Shift** key for multiple object selections, and to move multiple objects. You can also left-click and drag the mouse to select objects, such as partition pins and block pins.
- Use the **Space Bar** to change the highlighting focus on an object.
- Double-click the left mouse button on an object to view or change object attributes.

Right-click and drag the mouse to specify an area that you want to see in greater detail. When you release the mouse button, the display zooms in to the selected area.

Click the middle button of your mouse to pan the viewable window to the center point. This is equivalent to using the **panCenter** command.

Move the scroll wheel of your mouse to pan and zoom the design:

- To zoom in or out, simply move the wheel forward or backward.
- To pan up or down, press **Shift-key** and move the wheel forward or backward.
- To pan left or right, press **Ctrl-key** and move the wheel forward or backward.

EDI System General Binding Keys

EDI System provides keyboard shortcuts, called binding keys, to access commands in the design display area. A command's binding key is shown in the pull-down menus next to the command and also when hovering over an icon. To see the full list of predefined binding keys:

1. Select **Options - Set Preference**
2. On the **Design** tab select the **Binding Key** button.

   Here you can view, edit and add binding keys.
3. Close the **Binding Key** and **Preferences** windows.

Encounter Auto Query *Q* Binding Keys

The main window includes an Auto Query (Q) button located at the bottom of the design display area that enables and disables the auto query. When you enable auto query, text information displays for any object that is under the cursor. You can use auto query for blockages, instance pins, special route wires and vias, and cell instances. When you use the **`loadViolationReport`** command to load a file with violation markers generated by external tools, auto query displays the rule names for the markers.

If there are overlapping objects under the cursor, text information displays for the object that is on top. Use the **n** key to get information on the next object, and the **p** key to get information on the previous object. You can also use the **F8** binding key to copy auto query text from the message bar in the EDI System main window to the EDI System console. The text is displayed in the EDI System console with the prefix *Message*.

EDI System User Guide and Command Help

EDI System help is available and is displayed by an internet browser. Click the left mouse button (LMB) on the **Help** button in the upper right corner of the EDI System Main Window. This displays the EDI System product documentation.

Context sensitive help is done by clicking (LMB) on the **Help** button in the individual forms. This displays the related *Encounter Digital Implementation System Menu Reference* section.

EDI System Command *help* is available by using the Man pages. Type:

`man <commandName>` or

`help <keyword>` and a list of commands displays pertaining to the keyword.

Importing the Design

Importing the design - The Design Import form is used to load the Verilog netlist, physical libraries, process technology libraries, timing libraries, and timing constraints. Import the design by performing the following steps.

1. Select **File - Import Design**

2. Complete the Design Import form with the values below. You can use the file widget '…' to browse and select the other file names for the LEF and view definition file entries. Do not click the **OK** button yet.

   In the Netlist section:

   - Select **Verilog**
   - **Files:** *DATA/leon.v.gz*
   - Select **Top Cell:** Auto Assign

   In the Technology/Physical Libraries section select **LEF Files** and specify the following files in order. Note the LEF containing the technology information must be specified first:

   LEF Files: LIBS/lef/gsclib045.fixed.lef
   LIBS/lef/MEM2_128X32.lef
   LIBS/lef/MEM1_256X32.lef
   LIBS/lef/pads.lef

   In the Power section:

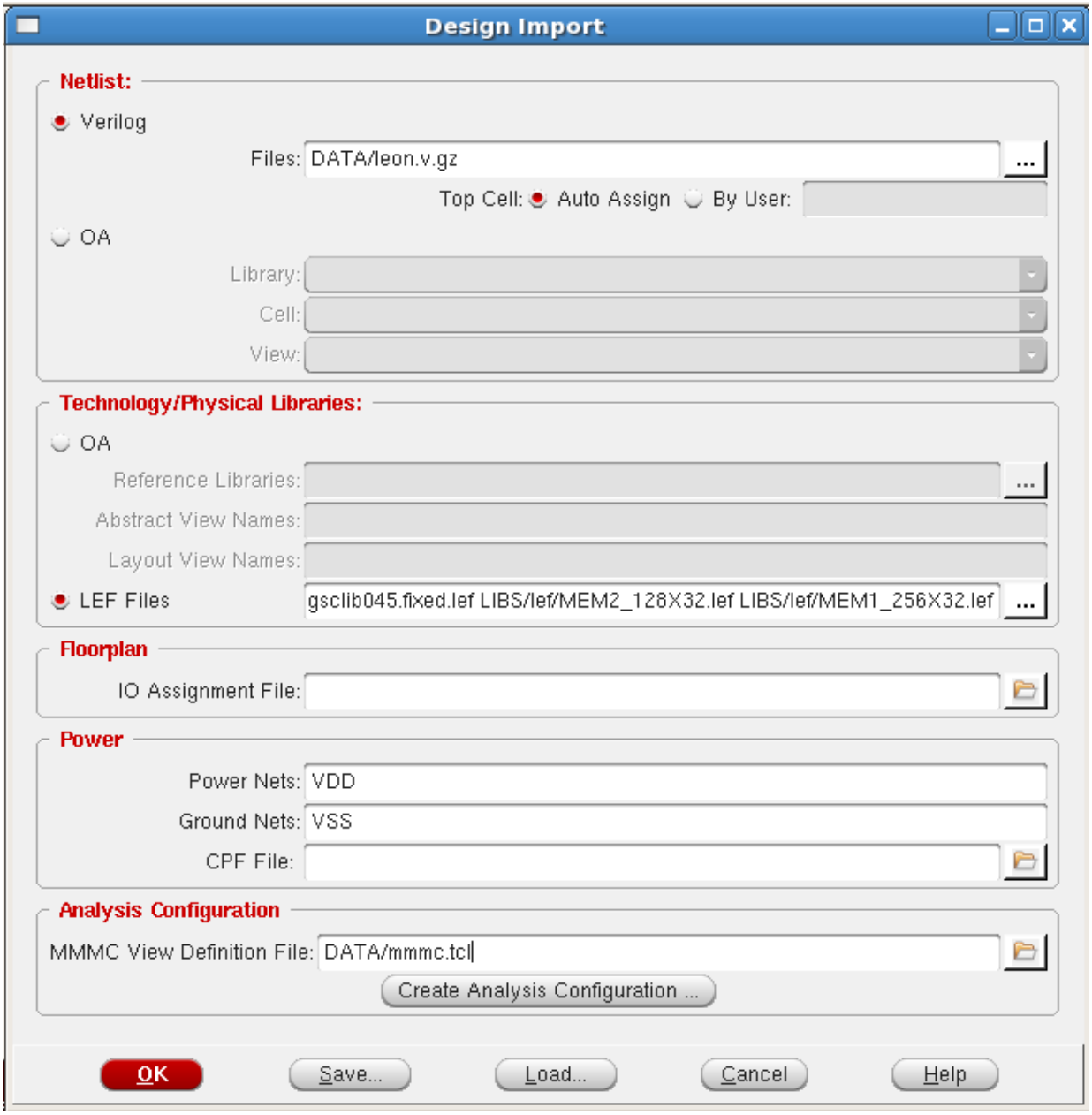
   - **Power Nets:** VDD
   - **Ground Nets:** VSS

   In the Analysis Configuration section:

   - **MMMC View Definition File**: DATA/mmmc.tcl

   The mmmc.tcl file contains the commands for defining the timing analysis views. Timing analysis views are made up of a delay corner and an operating mode. Delay corners define the timing libraries, RC corners and operating conditions to analyze. For details on defining timing analysis views see the section *Configuring the Setup for Multi-Mode Multi-Corner Analysis* in the *Encounter Digital Implementation System User Guide*.

   The completed form is shown on the next page.



If you want, you can click the **Load** button, and select the file *DATA/leon.globals* to completely setup the Design Import form.

3. Click **OK** to import the libraries and netlist.

4. Next set the design mode by entering the following command prompt. It is important to specify the process technology because it sets capacitance filters and extraction effort level based on the process node.

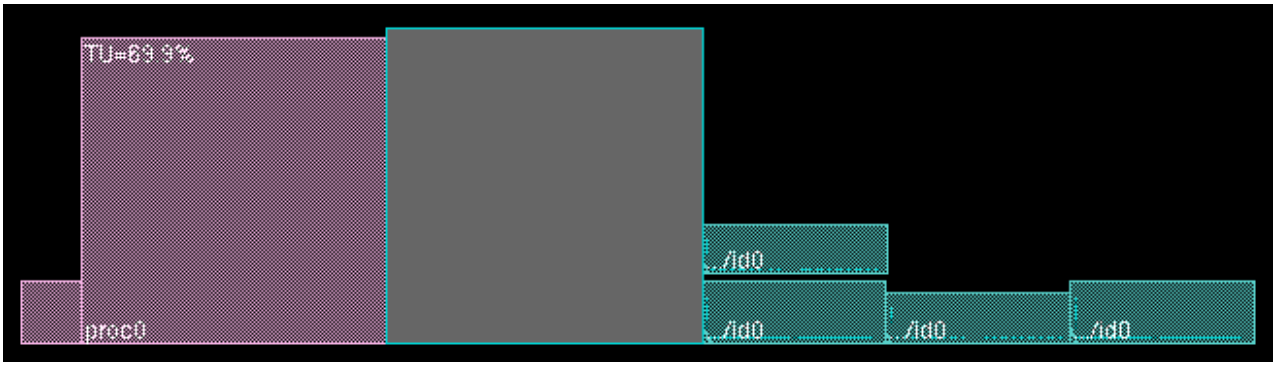
```
setDesignMode -process 45
```

Viewing the Imported Design – Now the design is imported. Before performing floorplanning you will familiarize yourself with the EDI System GUI using the instructions below.

First, get to know the Toolbar Widgets. The Toolbar is the row of widgets directly under the Menus.

To see what each widget is, slowly move the cursor over each widget to display their labels. For example, the folder widget on the far left side is the **Import Design** form.

1. Click the widget  to display the Floorplan view. The Floorplan view displays the floorplan specific objects such as hard macros, module constraints and power routing.
2. Zoom out by clicking (LMB) the **Zoom Out** widget  in the Toolbar (or enter the **Shift+Z** keys). Zoom out until you can see the pink module guides on the left side of the floorplan and the hard macros to the right size of the floorplan as shown below. Click and drag a box with the RMB to zoom to an exact area:



The pink colored objects on the left side of the core area are the modules instantiated in the top level of the imported Verilog netlist, and the 4 green-grayish colored objects on the right side are all the blocks (hard macros) in the design. The core area appears in the Design Display Area after a design is imported.

3. Click (LMB) on the large pink colored top-level module, *proc0*, and this selects the module. Note the blue color connection flight lines and white color highlighted blocks. The blue flight lines display the number of connections between the selected module and other instances such as other modules and

blocks. The blocks are highlighted white because they are child instances of *proc0*.

4. Now, click (LMB) the **Hierarchy Down** widget  in the Toolbar or enter the **Shift-G** keys (ungroup) to ungroup the *proc0* module. Note that there are four submodules displayed.

   Module display - By default only modules with 100 or more instances are displayed. To display modules with less than 100 instances select **Options - Set Preference.** On the Preferences form select the **Display** tab. Change the value of **Min. Floorplan Module Size** to **1** and click **OK**.

5. To traverse further down the hierarchy of a submodule select module *proc0/cmem0* and enter the **Shift-G** keys. To traverse back up the hierarchy, click the **Hierarchy Up** widget  in the Toolbar or enter the **g** key (group).

   Floorplanning a submodule information – At times there is a need to preplace submodules in the floorplan rather than the top-level module. In the above exercise, it demonstrates graphically how to traverse down a module's hierarchy so the submodule can be displayed and then floorplanned (guided into the core area).

6. Now double click (LMB) on one of the blocks to the right of floorplan. This will open the Attribute Editor displaying the attributes of this block. Remember that each floorplan object has properties and these properties can be changed in the Attribute Editor form. The Attribute Editor can also be opened by using the **q** key after selecting the object, or clicking the right mouse button (RMB) after selecting the object.

   Block orientation information – When a block is selected / highlighted, the flight lines are displayed to each pin of the block instead of the center of the block. This feature helps in determining the proper orientation of a block. For example, it helps with route congestion and timing when most of the block pins face the center of the chip or adjacent block pins of face each other.

Using the Design Browser – The **Design Browser** allows you to browse the design and search design objects hierarchically. Open the Design Browser using one of the following methods:

1. Select **Tools - Design Browser** or
2. Select the **Design Browser** toolbar widget 

The following are some of the features of the Design Browser:

- Observe the design statistics for the *leon* module.
- Expand objects by clicking the **(+)** sign next to the object.
- Place your cursor over the icons at the top to observe what each does.
- The bottom controls which colors are used when highlighting.
- The search box at the top searches for specified objects.
    - Use the pull-down menu to select the type of object to search for.
    - Wildcards can be used in search strings using an asterisk "*".

Take some time to familiarize yourself with the Design Browser. You will also use it later in the tutorial. Once you are done close the Design Browser.

Learning Floorplanning

Changing the core, IO, or die size is done with the Specify Floorplan form.

Specifying die size - Use the Specify Floorplan form to set the core box, IO box, and die box sizes following the instructions below:

1. Select **Floorplan - Specify Floorplan**

2. Use the default setting for:

   Core Size by: Aspect Ratio
   Ratio H/W: 1
   Core Utilization: 0.6

   Under the section of Core Margins by: Core to IO Boundary, make the following entries:

   Core to Left: 25
   Core to Right: 25
   Core to Top: 25
   Core to Bottom: 25

3. Click the **Apply** button.

   The floorplan is resized so the core area has the specified utilization and there are 25 microns between the core box and design boundary. Due to snapping you may see the 25um distance be slightly more by a fractional amount.

4. For this block we want the width to be greater than the height. To modify the core's aspect ratio of height/width (H/W), change the value for **Ratio (H/W)** to **0.7** and click the **OK** button. Now the core shape has been changed to the rectangular shape we want for this block.

Learning floorplan widgets/tools – In this section you will familarize yourself with the floorplan Tool Widgets. You will clear the edits you make at the end of this section so feel free to experiment with the widgets as much as you like. The floorplan Tool Widgets are the row of widgets directly above the art work window where the design is displayed. Identify each widget by slowly moving the cursor over each widget to display their labels. Note that some labels display a letter in parentheses (). The letter represents the binding key.

Moving objects – The Move/Resize/Reshape widget is used to enable the movement of objects. Use this to manually move objects into the floorplan. In this exercise you will move a submodule of *leon* into the core area.

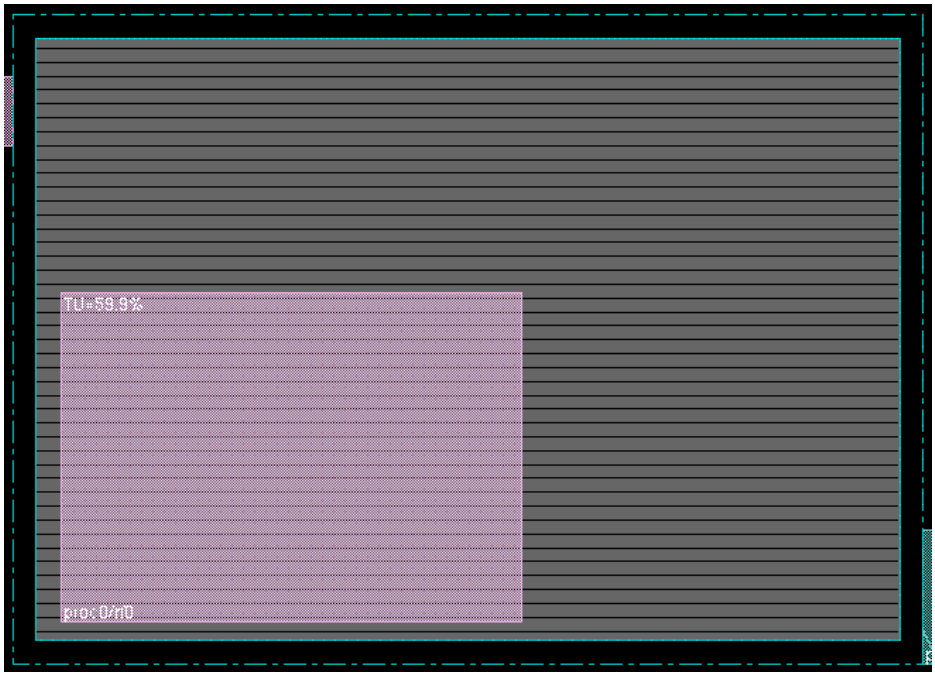
5. Ungroup the *proc0* module if it is not already by selecting it and pressing **Shift-G**.

6. Click (LMB) on the **Move/Resize/Reshape (R)** Tool Widget  .

7. Moving is done by using two clicks (LMB). The first click selects the objects to move. Then move the object to the desired location. The second click places the object at the desired location.

   Click (LMB) on one of the module guides to the left of the floorplan. The object becomes selected. As you move the mouse, the object moves with it. As the module is moved, a ghost and flight lines are displayed. At this point you can zoom-in or out, or pan the design.

8. Click (LMB) anywhere in the core area to place the module guide inside the core area. The floorplan should like something like the following:



Reshaping objects – The Move/Resize/Reshape widget is also used to reshape a module guide, say to a rectangular shape.

9. To reshape a module guide and to keep the area constant, Select **Options - Set Preference** to open the **Preferences** form.

10. Select the **Edit** tab.

11. On the **Edit** tab select the **Maintain Area** option for **Box Stretch Restrictions** then click **OK**.

12. Resize the module you placed in the core area by moving the cursor over one of its edges or corners. The cursor will change to a double arrow when positioned over the edge to move.

13. Click (LMB) to start the resize. Move the mouse to the desired location and click (LMB) again to complete the resize. Note the guide maintains its area as it is resized.

Moving multiple objects information – If you desire to move multiple objects at once select the Select widget then select (LMB) each object while holding down the **Shift** key. Then use the Move/Resize/Reshape tool widget to click on one of the selected objects to start the move. To deselect an object, hold down the **Shift** key and then click on the object to be deselected.

Attribute Editor form information – Each floorplan object has properties and these properties can be viewed and edited. The Attribute Editor can be opened several ways.

- Double-click an object with the (LMB) or
- Select/highlight the object and enter the **q** key or
- Select/highlight the object and click the (RMB). Then select **Attribute Editor** on the menu which appears.

To open the Attribute Editor on a nested floorplan object, first click the (LMB) directly over the object, then use the **Space Bar** key to traverse the objects, once the desired object is selected, enter the **q** key.

14. Select the module which you placed in the core and open the Attribute Editor using one of the methods above.

15. Change the **Contraint Type** to **Fence** and click **OK**.

Observe the module changes from pink to orange indicating it is a fence. Below is an explanation of the different placement constraint types:

None - The module is not pre-placed in the core design area. The contents of the module are placed without any constraints.

Guide -The module is preplaced in the core design area. A module guide represents the logical module structure of the netlist. The purpose of a module guide is to guide placement to place the cells of the module in the vicinity of the guide's location. The preplaced guide is a soft constraint. After the design is imported, but before floorplanning, you can locate module guides on the left side of the core area, which appear as pink objects (by default) in the Floorplan view.

Fence -The module is a hard constraint in the core design area. After specifying a hierarchical instance as a partition, the constraint type status of a module guide is automatically changed to a fence. Instances belonging to a module of type fence must be placed inside the fence boundary.

Region -This constraint is the same as a fence constraint except that instances from other modules can be placed within its physical outline by placement.

Soft Guide -This constraint is similar to a guide constraint except there are no fixed locations. This provides stronger grouping for the instances under the same soft guide. The soft guide constraint is not as restrictive as a fence or a region constraint, so some instances might be placed further away if they have connections to other modules.

Designing a rectilinear module, fence, or region – Next you will change the shape of the submodule to a rectlinear shape using the **Cut Rectilinear** widget. Make sure you still have a submodule placed in the core area.

16. Click (LMB) on the **Cut Rectilinear** Tool Widget  .

17. To create a corner cut, place the cursor at the corner of the module. The cursor will change to a double arrow. Now, the 1st click (LMB) starts the cut and the 2nd click (LMB) ends the cut. The box you draw over the module area is the area which is cut out.

18. Create a slot cut by moving the cursor to an edge so the cursor changes to a double arrow. Now, do the 2-step clicking. Again the area of the box you draw is cut away.

TU and EU values information – You may have noticed the abbreviation TU in the upper left corner of the module guides followed by a percentage. TU (Target Utilization) value represents the physical design size (area of the module, fence, or region) and is a rough estimation, since only the module's child standard cells and blocks are calculated.

The use of the TU value is to judge the area size while resizing or reshaping a module. The initial TU value is calculated during design import. Resizing or reshaping a module changes the TU value. This new calculated value is displayed immediately.

After placement, an EU value will also be shown. The EU (Effective Utilization) value represents placement utilization for the all standard cells and blocks plus all floorplan objects, such as placement blockage, routing blockage, density screen, and partition objects. EU values also include non-child standard cells and blocks preplaced inside a fence or region. The EU value must never be greater than 100%, since greater than 100% means the fence or region is physically too small and the design cannot fit.

Creating floorplan objects – This section shows how to use the Tool widgets to create floorplan objects such as Placement Blockages, Partial Placement Blockage, and Route Blockages. Create a placement blockage using the following steps:

1. Select the **Create Placement Blockage** widget  or press **Shift+Y**.

2. Draw a rectangle in the core by clicking once (LMB) for the first point, move the mouse and clicking again (LMB) to complete the rectangle.

   A red rectangle should appear displaying the placement blockage you drew. By default, a placement blockage prevents the standard cell placer from placing any standard cells in this area.

3. Right-click (RMB) on the placement blockage you just drew and select **Attribute Editor** on the pop-up menu. This will display the attributes of the blockage.

4. Note the **Type** value is set to **Hard**. Change the **Type** to **Soft** and click **OK**.

   The valid placement blockage types are:

   - **Hard** - The area cannot be used to place blocks or cells. This is the default.
   - **Partial** - Sets a percentage of the area that is unavailable for placement. Use the **Blockage Percentage** pull-down menu to select a percentage.
   - **Soft** - The area cannot be used to place blocks or cells during standard cell placement, but can be used during in-place optimization, clock tree synthesis, ECO placement or placement legalization (`refinePlace`).

- **Macro-Only** - Enables `planDesign` to keep macros out of the placement blockage; however, it enables standard cells to be placed inside the box if no blockage is present.

5. Create as many placement blockages of different sizes and types as you like. You will clear all these blockages later. Note you can also specify the placement blockage type to create by pressing the **F3** key while the **Create Placement Blockage** widget is active.

6. Next, create routing blockages in a similar manner. First, select the **Create Routing Blockage** widget  .

7. Draw a rectangle in the core by clicking once (LMB) for the first point, move the mouse and clicking again (LMB) to complete the rectangle.

   A yellow rectangle should appear displaying the routing blockage you drew.

8. Right-click (RMB) on the routing blockage you just drew and select **Attribute Editor** on the pop-up menu. This will display the attributes of the blockage.

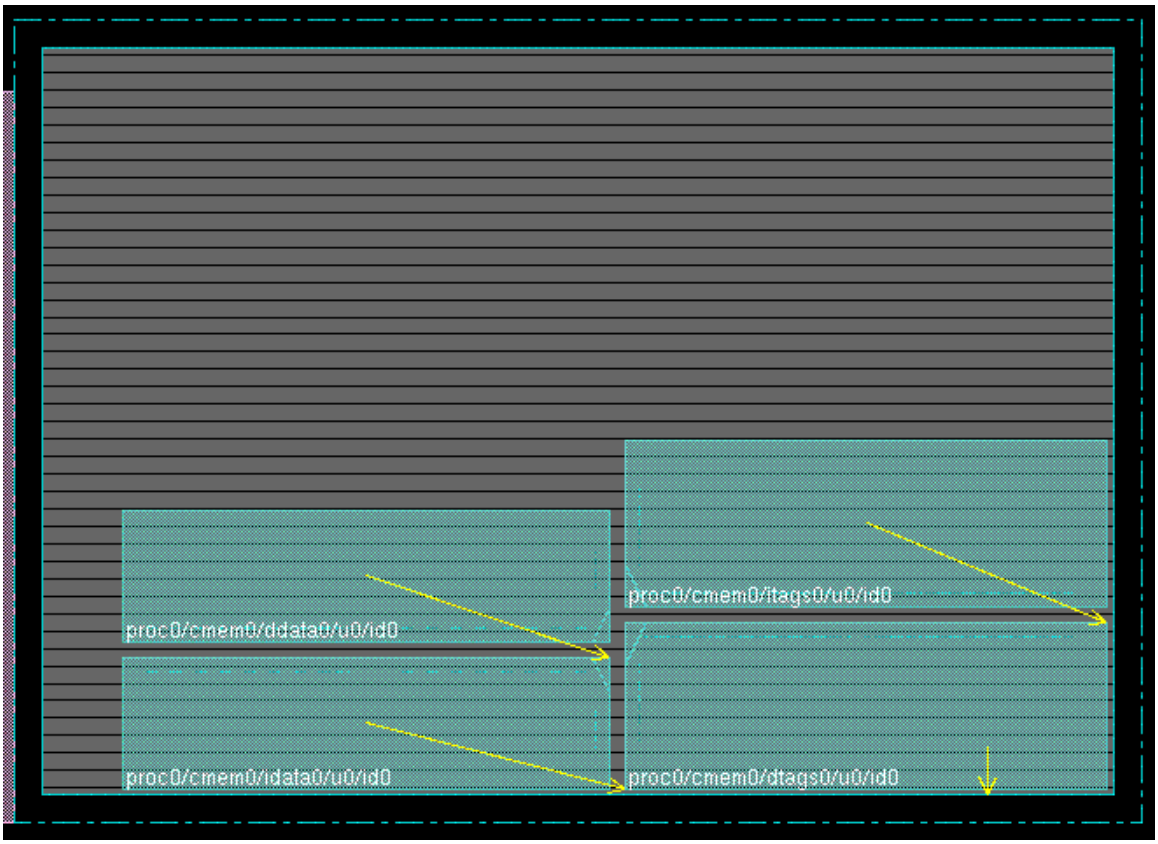
9. Observe you can specify the routing and cut layers for the blockage. The router is prevented from routing on the specified layers within the blockage area. Select as many layers as you like and click **OK**.

10. Create as many routing blockages as you like. You will clear all these blockages later.

11. Lastly, you can reshape or move blockages using the **Move/Resize/Reshape** widget and make them rectilinear using the **Cut Rectlinear** widget. Edit some blockages using these widgets.

12. Clear all the floorplan objects you created by selecting **Floorplan - Clear Floorplan**. Make sure **All Floorplan Objects** is selected and click **OK**. All the floorplan objects should now be removed from the core.

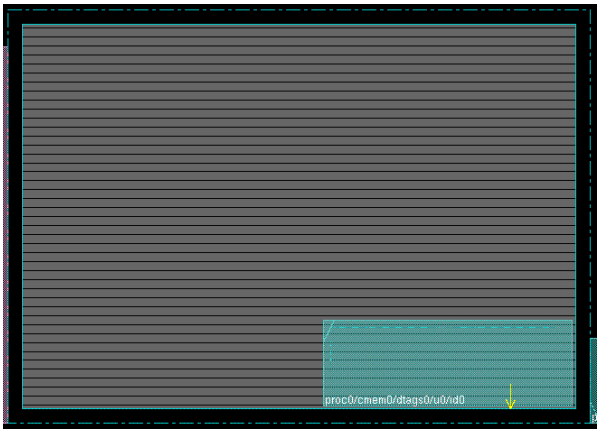
Creating a floorplan with Relative Floorplan – Designing a floorplan by accurately moving blocks into the core area is important. Accurate pre-placing of instances can be done with the Relative Floorplan form. This form allows placing an instance inside the core area, inside a module, or relative to another preplaced instance. By defining relative floorplan constraints, object placements are automatically updated if an object is moved which they are relative to. For this exercise, we will place the 4 blocks as shown below:



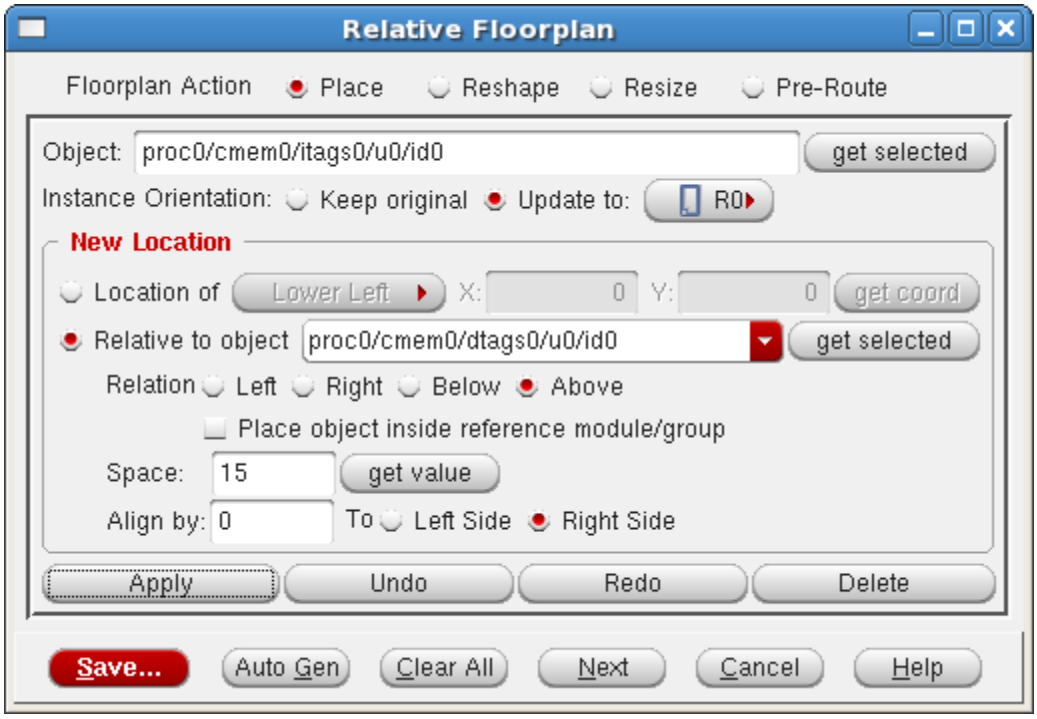
1. Select **Floorplan - Relative Floorplan - Edit Constraint**

2. Complete the **Relative Floorplan** form as shown below. This constraint says to place instance *proc0/cmem0/dtags0/u0/id0* 5um above the bottom core boundary and -5um from the right core boundary with an orientation of **MX**. Populate the Object field by selecting the block and clicking **get selected**. Note if you type in the instance name it will report an error "Object not given yet". This will be corrected in a future release. Until then, use **get selected** to populate the Object field with an instance name. Note when you select an object its instance name is displayed in the lower left corner of the GUI.

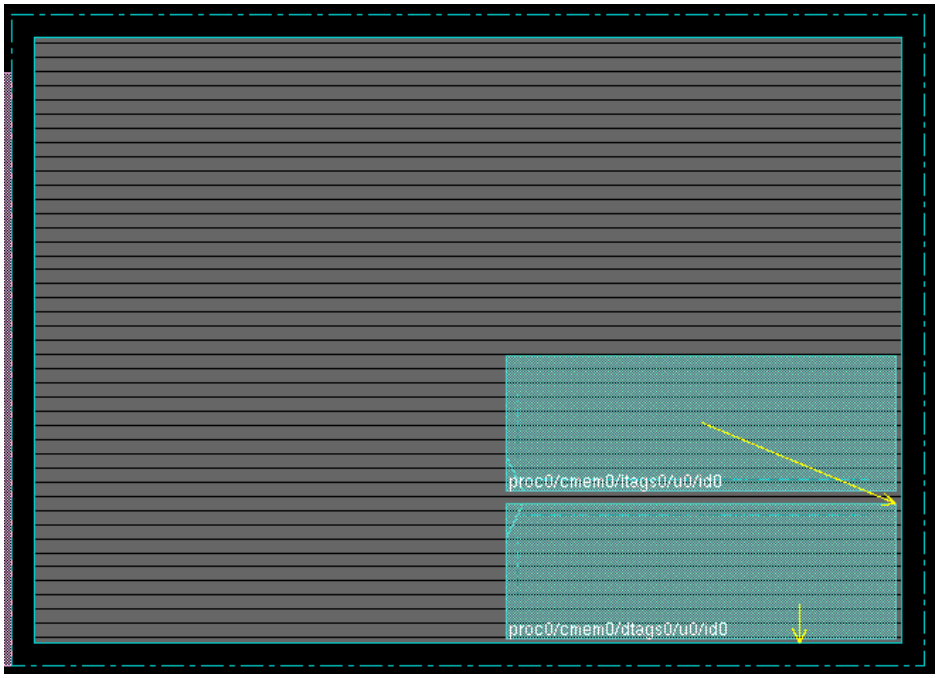
3. Click ***Apply***. The block should be placed as shown below. If the block placement is not as shown below, click ***Undo*** on the **Relative Floorplan** window, correct the settings and click ***Apply*** again.



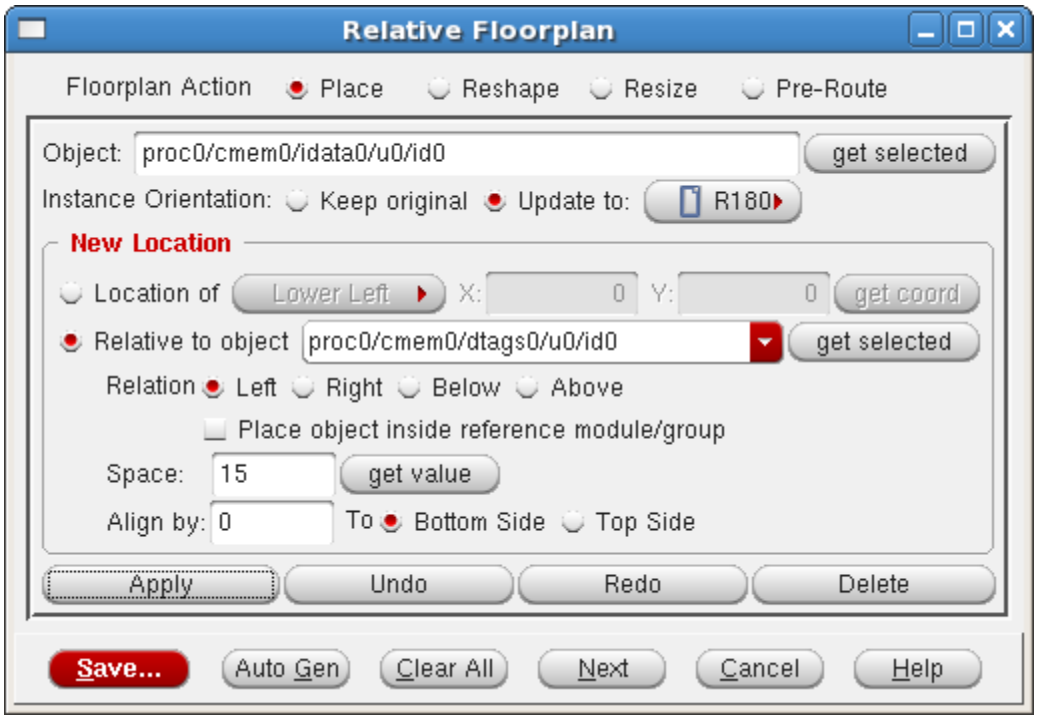
4. Place instance *proc0/cmem0/itags0/u0/id0* by completing the **Relative Floorplan** constraint form as follows and clicking **Apply**. This places the block 15um above the previous block we placed with an orientation of R0.



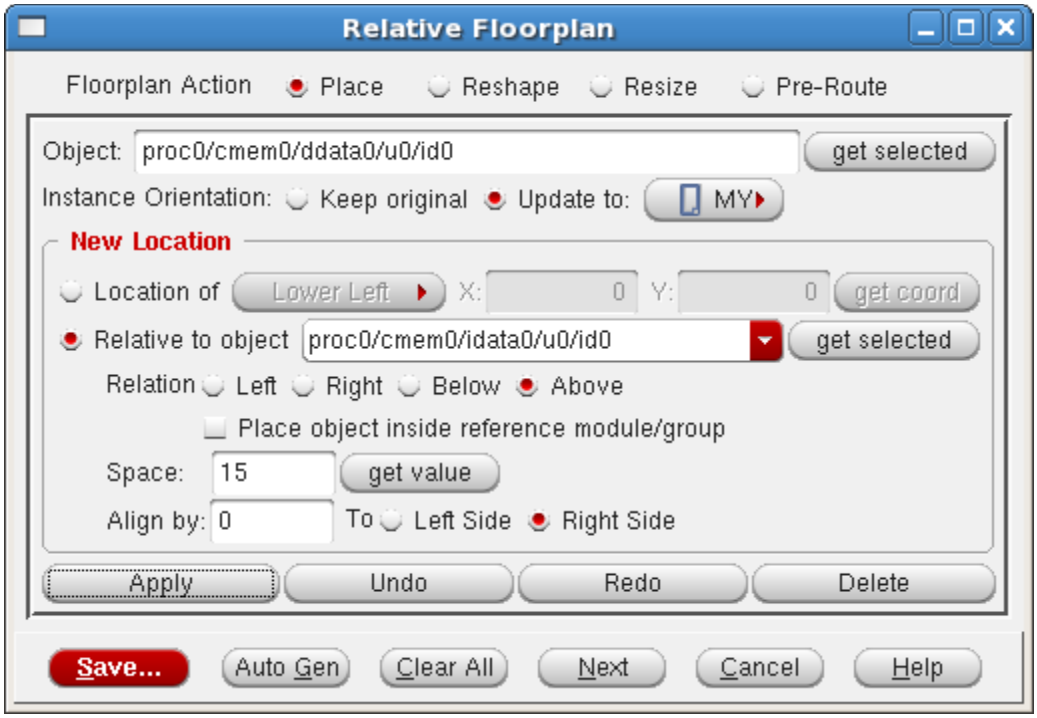
Result:



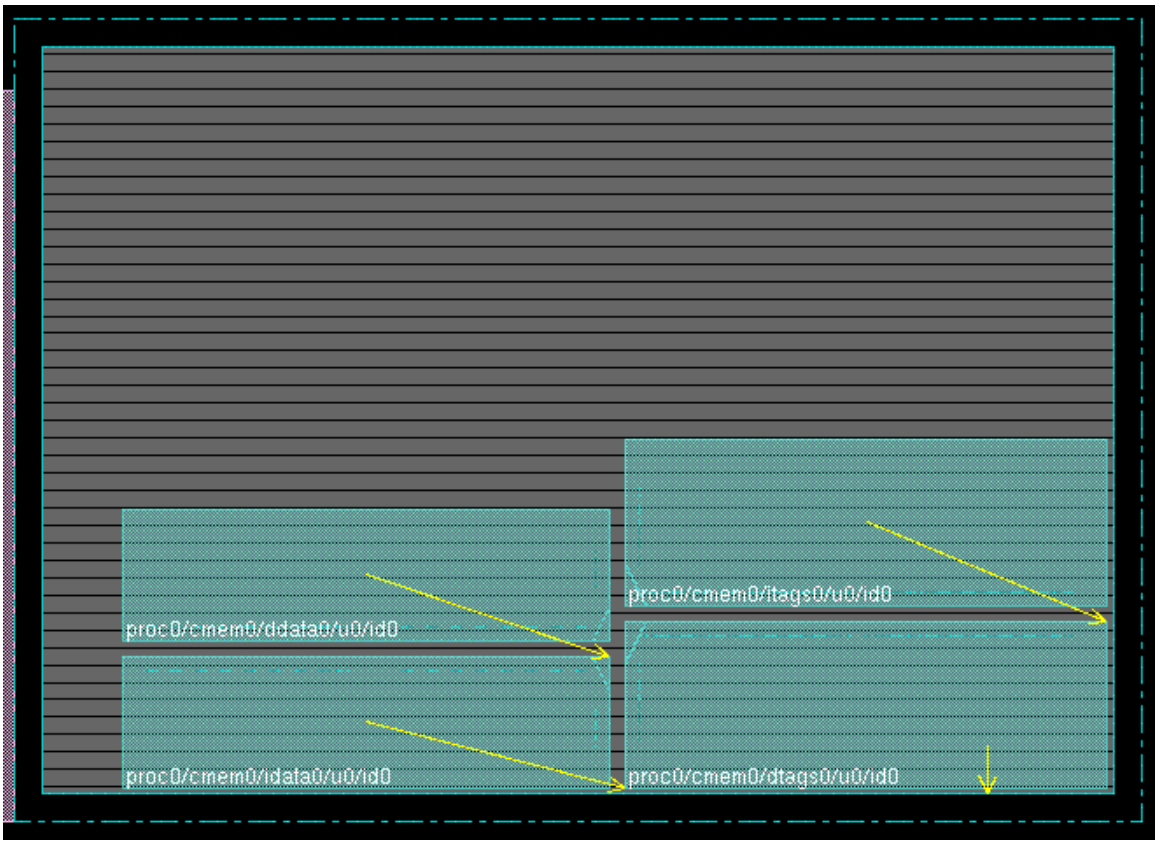
5. Place instance *proc0/cmem0/idata0/u0/id0* by completing the **Relative Floorplan** constraint form as follows and clicking **Apply**.



6. Place instance *proc0/cmem0/ddata0/u0/id0* by completing the **Relative Floorplan** constraint form as follows and clicking **Apply**.



7. The placement of the blocks should appear as follows. If they do, close the **Relative Floorplan** form. If you cannot pre-place the 4 blocks as in the sample floorplan, you can load the floorplan file, *leon.blocks.fp* by using the **File - Load - Floorplan** form.

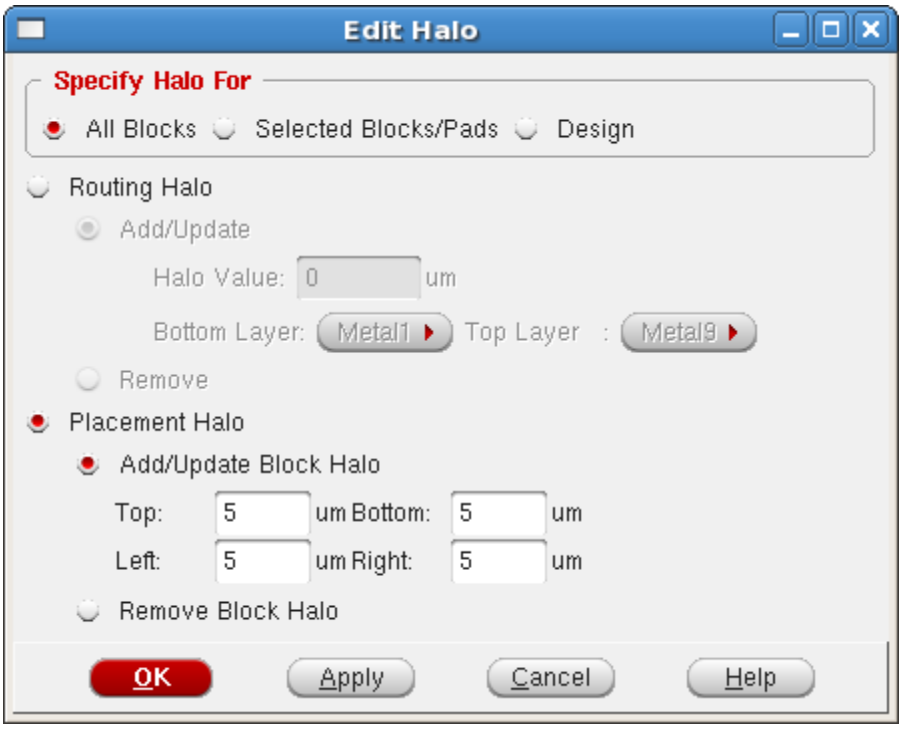


Note the display of the yellow color arrows. Each arrow shows the reference point of the placed block. The display of the yellow arrows can be turned off by unselecting the visibility toggle of the **Rel. FPlan.** This toggle is located on the right side of the EDI System window under Floorplan (Click the + next to **Floorplan** to reveal it).

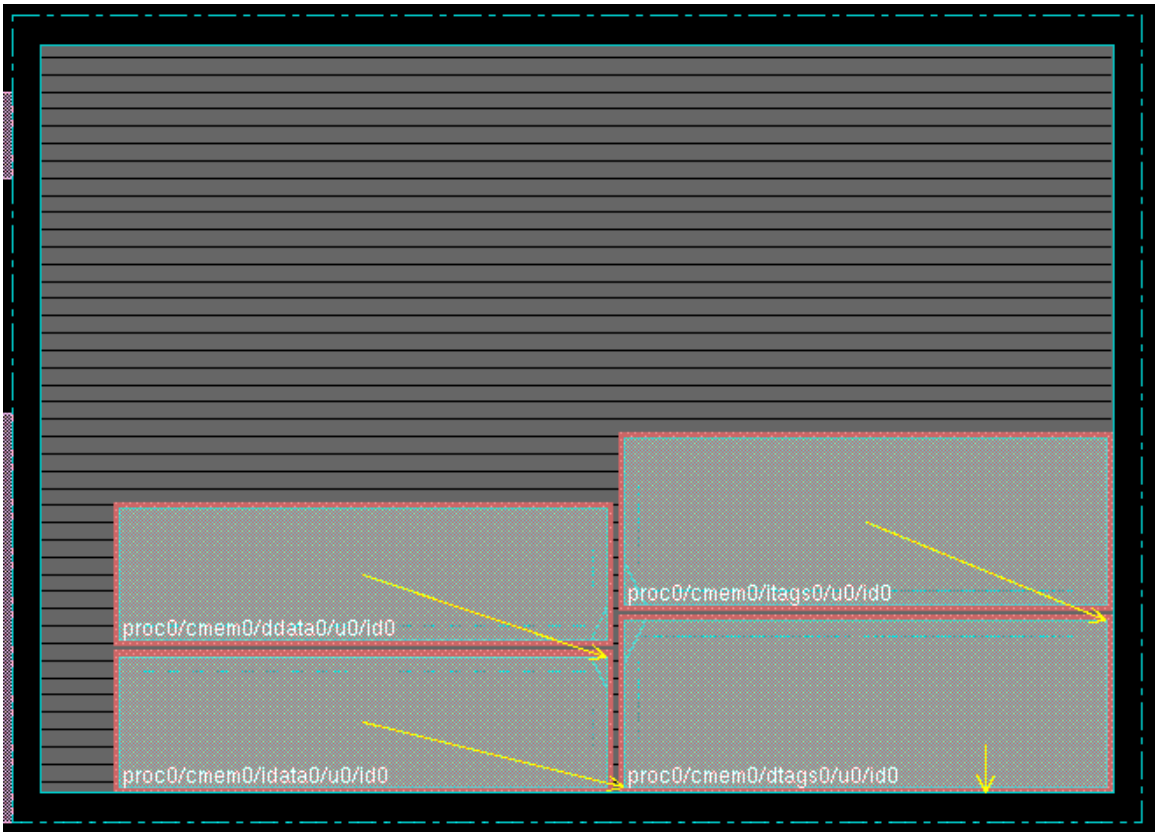
Creating block halos – Once you have all the blocks preplaced, block halos can be created for each block by using the **Edit Halo** form. A block halo prevents the placement program from placing standard cells in the halo area and away from the block. Placement halos are similar to a hard placement blockage. The advantage of using a block halo is it is associated with the block and therefore moves with the block when the block is moved.

1. Select **Floorplan - Edit Floorplan - Edit Halo**. Notice this form is used for adding and removing block halos.

2. Complete the **Edit Halo** form as shown below and click **OK**:



The floorplan should now have halos around each of the blocks as shown below.



Saving Your Design Work

Saving the Design - It is a good idea to save the design work as you progress and the most notable steps are after floorplanning, running placement, running route. To save the design using the **Save Design** form:

1. Select **File - Save Design**
2. Select **Data Type: Encounter**
3. Specify **File Name: *DBS/floorplan.enc***
4. Click **OK**.

When you save a database using the Encounter Data Type it will create a **.enc* file which is a pointer to the database directory which has the same prefix and a **.enc.dat* suffix. Under the *.enc.dat directory you'll see all the files which make up design database. Now you can restore your work in a future EDI System session using **File - Restore Design**.

Creating Power and Ground Rings and Stripes

Preparing to create power/ground rings – Before designing any power and ground rings or stripes, the global nets for power and ground must be assigned for the entire design. This is done with the Global Net Connections form or `globalNetConnect` command. From the netlist, the power pins, tie high pins, and tie low pins need to be connected to power and ground nets. There are 4 sets of entries required and they are shown in the table below.

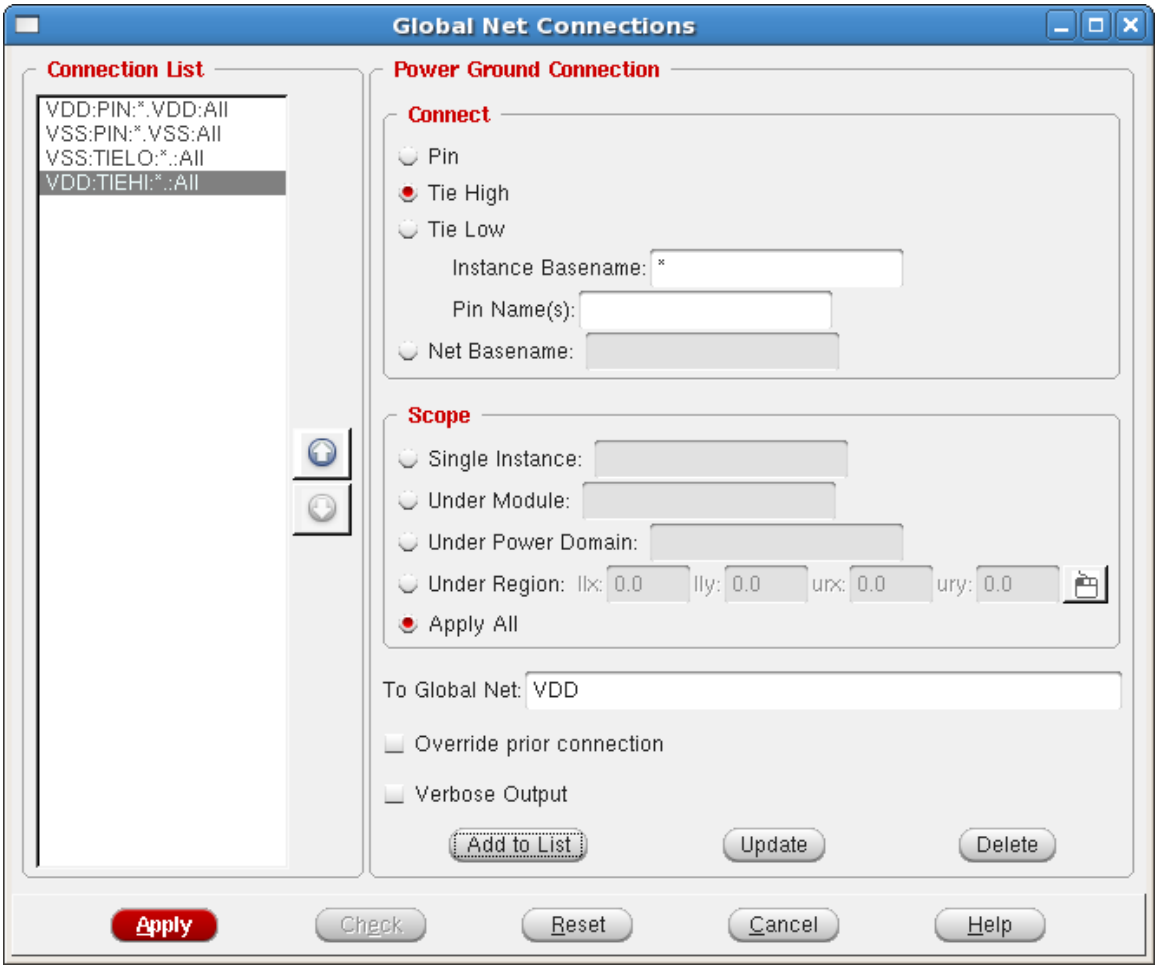
Entry	Set 1	Set 2	Set 3	Set 4
Pin Name(s)	VDD	VSS		
Instance Basename	*	*	*	*
Tie High	n/s	n/s	n/s	Selected
Tie Low	n/s	n/s	Selected	n/s
Apply All	Selected	Selected	Selected	Selected
To Global Net	VDD	VSS	VSS	VDD

n/s = not selected

1. You can specify the above connections using the `globalNetConnect` command or the GUI through **Power - Connect Global Nets**. To use text commands enter the following in the EDI System console:

```
globalNetConnect VDD -type pgpin -pin VDD -all
globalNetConnect VSS -type pgpin -pin VSS -all
globalNetConnect VDD -type tiehi
globalNetConnect VSS -type tielo
```

2. If you prefer to use the GUI select **Power – Connect Global Nets**. Enter the data for Set 1 above and click **Add to List**. Continue for Sets 2 through 4 and when done, the completed form looks like the picture below.



3. Click the **Apply** button when all 4 sets appear correct in the **Connection List**. This will apply the connections to the netlist.

4. Click **Cancel** to close the **Global Net Connections** form.

Creating power/ground rings – Now with the power and ground nets logically assigned, power planning can be done. In this design, power and ground rings are added around the core area. Power and ground stripes are also added to the top level design. The Add Ring and Add Stripe commands are used to create the power grid.

1. Make instance pins visible. This is done on the right side of the GUI. First, click the "+" next to **Cell**. Then select the box in the checkbox next to **Pin Shapes**. You should now see pins displayed on the blocks.

Add Power Rings Around Core:

2. Select **Power - Power Planning - Add Ring**

3. Add power rings around the core by completing the **Add Rings** form as follows and click **Apply**:

   Net(s): ***VDD VSS***

   In the Ring Type section:
   - Select – **Core ring(s) contouring**
   - Select – **Around core boundary**

   In the *Ring Configuration* section:
   - Set **Top** and **Bottom** layers to **Metal9 H**
   - Set **Left** and **Right** layers to **Metal8 V**
   - Set the **Width** for each side to **10**.
   - Set **Spacing** to **1.25** for each side or click the **Update** button to automatically set the minimum spacing for this width of wire. Note this button automatically updates the values in the *Spacing* fields. If spacing values in the *Spacing* fields are less than the values from the LEF file for a particular width, the values in the *Spacing* fields are increased to prevent minimum spacing violations.
   - Select – **Offset: Center in channel**

   Click **OK** and you should see two power rings going around the core on Metal8 and Metal9 with a width of 10um.

Add Power Stripes:

4. Select **Power - Power Planning - Add Stripe**.

5. Complete the **Basic** tab as shown below. Do not click **OK** yet.

   Net(s): ***VDD VSS***
   Layer: **Metal8**
   Direction: **Vertical**
   Width: **8**
   Spacing: **1.25**

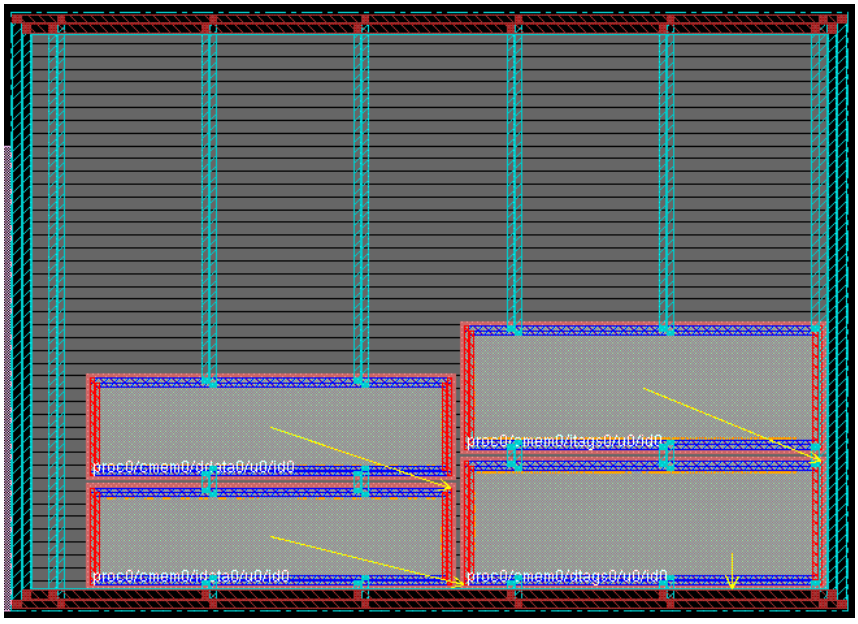
   In the Set Pattern section:
   - Number of sets: **6**

   In the Stripe Boundary section:
   - Select – **Core ring**

   In the First/Last Stripe section:
   - Select – **Start from**: **left**
   - Select – **Relative from core or selected area**
   - Specify – **X from left: 20**

6. On the **Advanced** tab select **Omit stripes inside block rings**. This will break stripes so they do not go over the blocks.

7. Click **OK**. Now you should see the power and ground rings and stripes created for the design as shown below:



Routing the power/ground structures – To route the remaining power and ground structure, the SRoute form is used. The SRoute program routes the block pins, pad pins, pad rings, and standard cell pins. For our exercise, power and ground needs to be routed to the standard cell pins. Make the following entries:

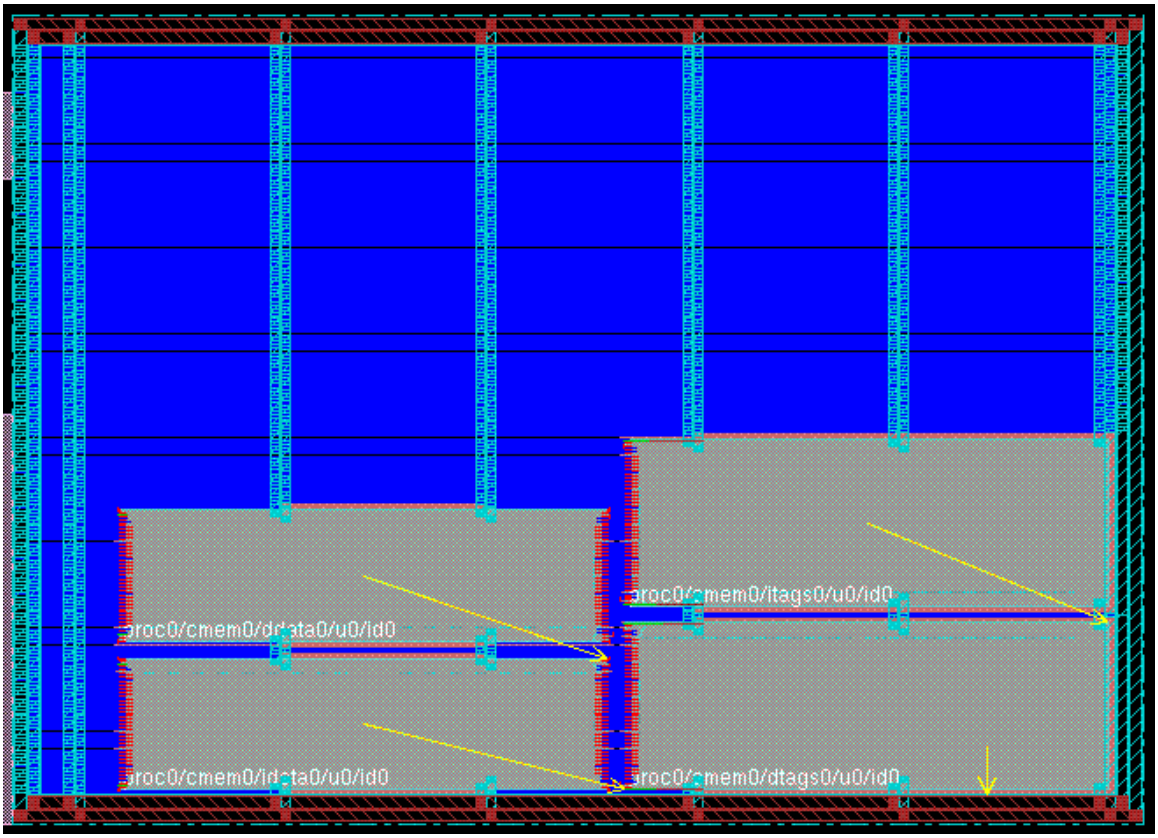
8. Select **Route - Special Route**

9. In the **Net(s)** field specify: *VDD VSS*

10. On the **Basic** tab deselect **Block Pins**, **Pad Pins**, **Pad Rings** and **Floating Stripes**. Keep everything else as the default.

11. Click **OK**.

This creates the standard cell power rails on Metal1 horizontally across the core. Your floorplan should now look like the following:



Running Placement

Supplied floorplan – This is optional. You can use your designed floorplan but as an option there is a workshop supplied floorplan file. To use your floorplan, skip following steps 1 through 3.

1. Select **File - Load - Floorplan**
2. Select the file *DATA/leon.power.fp*.
3. Click **Open**.

Before proceding to placement save the design by typing the following at the EDI System command prompt:

```
saveDesign DBS/power.enc
```

Running Placement – Use the Place form to run placement program and make the following entries:

4. Select **Place - Place Standard Cell**
5. On the Place form observe different options:
   - Three modes allow you to run full, incremental or floorplan (prototype) placement.
   - Pre-place optimization optimizes the netlist prior to placement such as removing buffer trees.
6. Left click (LMB) on the **Mode** button which opens the **Mode Setup** form. Observe the different mode settings for placement. Scan chain reordering is done by default during placement.
   - Deselect - **Reorder Scan Connection**
7. Click **OK** on the **Mode Setup** form.
8. Click **OK** on the **Place** form to run placement. It will take a few minutes for placement to complete.

Viewing the design after Placement – Once the placement program is done, you can view the placed design in the Amoeba and Physical views. Observe the placed standard cells are visible in the Physical view. Also, notice there is routing. Placement automatically calls Trial Route so you can analyze congestion. You may need to zoom in or turn of the display of Nets to see the placed standard cells. In the next section you'll manually run Trial Route to analyze congestion.

9. Save the placed design:

```
saveDesign DBS/place.enc
```

Running Trial Route

Running Trial Route – Trial Route is a combination of global routing and track assignment which correlates well to detail routers while running in a fraction of the time. In this workshop, Trial Route is run twice. The first run will demonstrate route congestion by limiting the number of metal routing layers to 3 instead of 9 layers. The second run will use all 9 metal layers to complete the prototyping of the design.

1. Select **Route - Trial Route**

2. Change – **max: Route Layer** to **3**

3. Click **OK**.

Once trial route completes, analyze the congestion. This is more visible if you turn off the visibility of nets.

4. Turn off the visibility of nets by deselecting the checkbox next to **Net** on the right side of the GUI.

You should now see some red color diamond shapes which are Trial Route congestion markers. Zoom-in to view the congestion markers. Inside the diamond marker, there is a set of numbers in the format of either **H: #-top/#-bottom** or **V: #-top/#- bottom**. H stands for horizontal congestion and V stands for vertical congestion. The #- top is for the required number of routing tracks used in this area and #-bottom is the available routing tracks. These diamond shaped congestion locators represent an average in the area. Since there is only 1 vertical route layer (metal 2), most congestion markers are vertical.

Another display of congestion is to select the visibility of Horizontal Congest and Vertical Congest in the color display option area:

5. Select **Options - All Colors**

6. Select the **View-Only** tab

7. Turn on **Horizontal Congest** and **Vertical Congest**

These two congestion makers are color coded, and dark blue represents low congestion and red and white high congestion. They are easier to spot in a design.

8. Turn off **Horizontal Congest** and **Vertical Congest** then close the **Color Preferences** window.

9. Make nets visible again by selecting the checkbox next to **Net** on the right side of the GUI.

Another way to analyze congestion is to view the log file or the EDI System console for the congestion table produced by trial route. The label of the table is Congestion distribution: Just before this table, look for the last line with label *Overflow*. If both numbers in the (#% H) and (#% V) are less than 0.5%, then this design is good for any detail router. The less than 0.5% is good for three (3) layers of metal and less than 1.0% is usually good for five (5) or more layers of metal. For example with only 3 layers the overflow is very high:

```
Overflow: 7.65% H + 7.53% V (0:00:08.9 719.7M)
```

Now, run Trial Route with 9 metal routing layers.

10. Select **Route - Trial Route**

11. Change – **Max: Route Layer** to **9**

12. Click **OK**. Observe the overflow is much more reasonable:

```
Overflow: 0.00% H + 0.27% V (0:00:02.3 724.8M)
```

Viewing the Design after Trial Route – You can select the trial routed nets connected to an instance or module by double clicking a module, instance or block (this also opens the Attribute Editor form). This can be used in any view depending on what you want to view.

13. Double click on modules, blocks and standard cells to display the connections. Note standard cells are only visible in the Physical view while module guides are only visible in the Floorplan or Amoeba views.

Finding and highlighting a net – To locate and highlight a net in the core area, the Design Browser can be used.

14. Select the Physical View icon if you are not currently in the Physical View.

15. Select **Tools - Design Browser**

16. In the Find text box enter the net name *proc0/rf0/u0/u0/n_7685* followed by a **<CR>**.

17. After the net is displayed in the Design Browser, select the net by clicking on it in the Design Browser and then click the **Select** widget  .

18. Click the **Zoom Selected** widget  to zoom-in to the highlighted net. To zoom-in and get a closer view of the net, use **Zoom In** widget in the Toolbar or use the (RMB).

Displaying path information – Note the paths end with either an X or an O. The X marks an input pin and O marks an output pin. The Xs and Os display better in the Floorplan view.

Displaying design objects in the Design Browser information – You can also select a net or instance in reverse with the Design Browser. The procedure is first, select a net or an instance in the core area, and then in the Design Browser, click the **Get Selected** widget  , and now the name of the highlighted object is displayed.

A second method in finding and highlighting object information – To select and highlight an instance, group, net, instances on a net, IO pin, and more, open the **Edit - Find/Select Object** form.

Viewing Different Objects – Click the **All Colors** button to open the **Color Preferences** form to see all the objects in EDI System. The color of each object can be changed, the display of these objects can be turned off or on, and the selectivity of objects can be changed.

Spend time familiarizing yourself with the Design Browser, Find/Select Object form and editing colors on the All Colors form.

Extracting RC Data

Extracting RC Data – The capacitance and resistance values for all the nets in the design are extracted by the Extract RC form.

Extraction is run in pre-route mode prior to signal routing and in post-route mode after the signals are routed with NanoRoute. In post-route mode there are four effort levels to choose from (low, medium, high and signoff) which increase with accuracy at the expense of longer runtimes.

The RC extraction mode can be changed by the **Options - Set Mode - Specify RC Extraction Mode** form. Since the design has not been routed we will leave the default mode set which is pre-route mode.

1. Select **Timing - Extract RC**

   Select – **Save SPEF to** and specify file *leon.spef*
   Select – **RC Corner to Output**: *rc_worst*

2. Click **OK**.

If you review the log file you'll see extraction is run and then a SPEF file is written out.

Calculating Delays – The delays are calculated for each wire (routes) and the delays include instance delay.

To run Delay Calculation, do the following:

3. Select **Timing - Write SDF**

   Select – **Ideal Clock** since clock tree synthesis has not been run yet

4. Click **OK**.

Delay calculation is run and an SDF file is written out.

Running Timing Analysis

Running Setup Timing Analysis and Displaying Violations and Slacks – Timing analysis can be run after extracting RC. Note if extraction was not run prior to timing analysis, timing analysis will automatically. Run RC extraction followed by timing analysis. Since clock tree synthesis was not run, an ideal clock is used and the ideal clock transition delay value is 0.1ps.

1. Select **Timing - Report Timing**

2. In the Design Stage section, Select – **Pre-CTS**.
   In the Analysis Type section, Select – **Setup** which is default.

3. Click the **Advanced** tab and view the options. Use the remaining defaults and click **OK**.

When timing analysis is done, a timing summary is printed to the EDI System shell. This information is also written to the file *timingReports/leon_preCTS.summary*. The *timingReports* directory contains several additional reports commonly used for debugging:

```
leon_preCTS.cap           - max capacitance violations
leon_preCTS.fanout        - max fanout violations
leon_preCTS.slk           - slack report for each endpoint
leon_preCTS.summary       - timing summary
leon_preCTS.tran          - max transition violations
leon_preCTS_all.tarpt     - timing report for all path groups
leon_preCTS_in2out.tarpt  - report for input to output paths
leon_preCTS_clkgate.tarpt - report for clock gating paths
leon_preCTS_in2reg.tarpt  - report for input to register paths
leon_preCTS_reg2out.tarpt - report for register to output paths
leon_preCTS_reg2reg.tarpt - report for register to register paths
```

4. After running timing analysis you typically want to explore and debug certain timing paths. Run `report_timing` to report timing for the worst path.

   ```
   report_timing
   ```

   `report_timing` reports timing for the worst path by default. You can report specific paths using the `report_timing` options. Type `report_timing -help` for a complete list of its options.

Use the Global Timing Debug Browser to get a better visual display of the timing paths. This browser is a powerful debug tool providing cross-probing between the report information and the EDI System design display area. Perform the following to open the Global Timing Debug Browser:

5. Select **Timing - Debug Timing**

6. Click **OK**.

A Timing Debug form displays path histogram and a Timing Browser displays in the EDI System main window.

The Global Timing Debug interface is a powerful tool for debugging timing issues. Features include:

- Histogram display of overall timing picture
- Ability to create categories of paths to group similar paths together
- Cross probing of paths to the design
- Graphical display of individual path (double-click on path)
- Several Analysis options including bottleneck analysis available through the Analysis menu

This tutorial does not go into detail on using Global Timing Debug but feel free to experiment with it. You can select **Help** from the **Help** menu to show details on the different capabilities provided by Global Timing Debug.

7. Close the Timing Debug window(s) once done.

Running Pre-CTS Timing Optimization

Timing Optimization is run at several times during the implementation flow to fix timing violations (setup and hold) and design rule violations (max transition and max capacitance violations). Run preCTS optimization:

1. Select **Optimize - Optimize Design**

2. Keep the defaults and click **OK**.

Timing Optimization will run through several iterations and when timing is met, the program will stop. You can view the EDI System console and read the *optDesign Final Summary* for a summary of the results.

You can reload the new timing report in the Timing Debug browser:

3. Select **Timing - Debug Timing**

4. When the timing browser displays in the EDI System main window, click the file widget on the right side of the **Report File(s)** field and this opens the **Display/Generate Timing Report** form.

5. Click **OK**. This generates and displays the new timing information.

Notice there are only a few, if any, violating paths in the histogram.

6. Close the **Timing Debug** form.

Running Clock Tree Synthesis (CTS)

Clock Tree Synthesis is run after running pre-CTS Optimization, but first the clock tree specification file must be created.

Creating the Clock Specification file – The design's clock information is contained in the timing constraint file (SDCs), and the Clock Tree Specification form is used to create the clock tree specification file.

1. Select **Clock - Synthesize Clock Tree**

2. Click on **Mode** and observe the mode settings you can control for CTS.

3. Click **Cancel** to close the **Mode Setup** form.

4. On the **Synthesize Clock Tree** form click **Gen Spec**. This will open the **Generate Clock Spec** form.

5. On the **Generate Clock Spec** form select **CLKBUFX12** and **CLKINVX12** then click **Add** to add them to the **Selected Cells** list.

6. Click **OK** on the **Generate Clock Spec** form.

7. In a text editor open the CTS constraints file *Clock.ctstch* that was created. Close the file after reviewing the file.

Now you are ready to synthesize the clock tree using CTS.

8. The **Synthesize Clock Tree** form should now have the following values:

   - Clock Specification Files: *Clock.ctstch*
   - Results Directory: *clock_report*

9. Clock **OK** to perform clock tree synthesis.

To view the results of clock tree synthesis:

10. Select **Clock - Display - Display Clock Tree**

11. In the Route Selection section,

    - Select – **Clock Route Only**

    In the Display Selection section,

    - Select – **Display Clock Tree**

- Select – **All Level**

12. Click **Apply**. Observe the clock nets are selected in the GUI.

13. Select – **Display Clock Phase Delay** and click **OK**.

When displaying the clock phase delays, note the multi-color clock instances, which represent the delay in the clock path. Note you may need to deselect nets to see the colors of the clock phase delay.

14. Clear the display by selecting the **Clock - Display - Clear Clock Tree Display**.

A clock report file and timing analysis files are generated and written to the *clock_report* directory. The ASCII text version of the clock report is *clock.report*.

Lastly, further analysis can be performed using the Clock Browser (**Clock - Browse Clock Tree**) and Clock Tree Analyzer (**Clock - Debug Clock Tree**). We do not cover these tools here but you're welcome to familiarize yourself with them leveraging the EDI System User Guide and Menu Reference as needed.

15. Save the design by typing the following at the EDI System command prompt:

```
saveDesign DBS/cts.enc
```

Running Post-CTS Timing Optimization

Running Post-CTS Setup Timing Analysis and Optimization – At this point the clock tree has been synthesized and we can analyze timing based on the actual clock tree delay and skew. Previously, we ran timing analysis and optimization using the GUI. This time use the text commands:

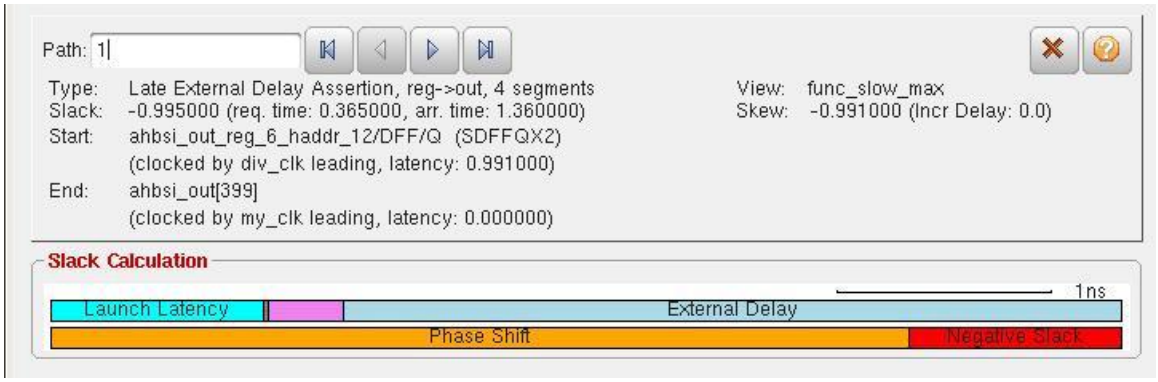
1. Run post-CTS timing analysis by typing the following commands in the EDI System console:

```
setAnalysisMode -analysisType onChipVariation
setAnalysisMode -cppr both
timeDesign -postCTS
```

2. The timing summary shows the timing violations exist primarily on register-to-output paths. This is because the clocks are now propagated so the launching of data occurs later. But the clock used to check timing at the output pin still has a latency of 0. You can see this easily if you view the worst path in the Global Timing Debug GUI as before:

   A. Select **Timing – Debug Timing**
   B. Select **File – Load Report File** then click **OK**.
   C. Double-click the worst path in the Path List

3. The top path will look something like the following. Notice the Start point has a latency of 0.991ns (your value may differ) while the End point has a latency of 0. Its latency is 0 is because it is a primary output and the timing constraints do not define a latency for this output port. Therefore, there is a skew of -0.991ns making it impossible to meet timing on this path.



4. We can adjust the latency of these clocks by running the command `update_io_latency`. Enter the `update_io_latency` command at the EDI System shell prompt:

   ```
   update_io_latency
   ```

5. Run timing analysis again to see the updated latencies resolve the register-to-output path violations:

   ```
   timeDesign –postCTS
   ```

   Optionally, open the Global Timing Debug tool and observe the latency at the output port is no longer 0.

6. Review the timing results. If timing violations exist run the following to fix them:

   ```
   optDesign –postCTS
   ```

   If only DRVs remain you can perform just DRV fixing by running:

   ```
   optDesign –postCTS -drv
   ```

   Next you will perform Post-CTS hold fixing.

Running Hold Timing Analysis and Generating a Slack Report – Perform timing analysis in hold mode:

7. Select **Timing - Report Timing**

8. In the Design Stage section,
   - Select – **Post-CTS**.

   In the Analysis Type section,
   - Select – **Hold**.

9. Use the remaining defaults and click **OK**.

When hold timing analysis is done, view the EDI System console and read the *timeDesign Summary* for the worst slack values. The paths with negative slack value have hold time violations.

Running Hold Timing Optimization – Perform timing optimization in hold mode to fix the hold violations:

10. Select **Optimize - Optimize Design**

11. For the Design Stage section:
    A. Select – **Post-CTS**.

    For the Optimization Type section:
    - Deselect – **Setup**
    - Select – **Hold**

12. Click **OK**.

Note if Setup and Hold are both selected then EDI System will run *optDesign –postCTS* to fix setup violations followed by *optDesign –postCTS –hold* to fix hold violations. We disabled setup timing optimization here because we ran it above.

All hold time and setup time violations should be fixed. View the EDI System console and read the *optDesign Final Summary* for the worst slack values for both Setup and Hold modes.

At this point, the design has met setup and hold timing constraints, but timing is really not closed on a design unless crosstalk is prevented and then analyzed. This exercise is next.

13. Save the design:

```
saveDesign DBS/postcts_hold.enc
```

Timing and SI Driven Routing

Timing is not closed on a design unless crosstalk is prevented, analyzed, and then fixed. To really achieve the above, the design is run with global and detail routing using NanoRoute. When running NanoRoute, we enable both the Timing Driven and SI Driven (Signal Integrity) options in the NanoRoute form. These options are important in helping to close timing and preventing crosstalk.

Running NanoRoute – Run NanoRoute in Timing and SI driven mode:

1. Select **Route - NanoRoute - Route**

2. In the Concurrent Routing Features section,
   - Select – **Fix Antenna**
   - Select – **Timing Driven**
   - Select – **SI Driven**

3. Click **OK**.

NanoRoute information – Running NanoRoute with the SI Driven option requires the capacitance table file (or QRC technology file), and this file was read in during design import. Choosing SI Driven option in NanoRoute is our first line of defense against noise.

Post-Route Timing and SI Optimization

The design is fully routed and timing analysis should be run to analyze timing based on the actual routes.

4. At the EDI System prompt run Post-Route timing analysis to report any setup or hold violations. The following increases the RC extraction effort level to medium so turbo-QRC is run which correlates better to signoff extraction. Note an effort level of medium or higher requires a QRC technology file and effort level of high or signoff also requires a QRC license.

```
setExtractRCMode -engine postRoute
setExtractRCMode -effortLevel medium
timeDesign -postRoute
timeDesign -postRoute -hold
```

5. Run post-route timing and SI optimization using the Advanced Analysis Engine (AAE). AAE is a new unified delay calculation engine that simultaneously computes base and SI delays. It takes advantage of multi-CPU optimization and runs significantly faster than the traditional CeltIC based flow.

```
setDelayCalMode -engine default -SIAware true
optDesign -postRoute
optDesign -postRoute -hold
```

6. After running optimization with AAE it is important to run timing analysis with CTE for signoff. If you have a QRC Extraction license and have a QRC in your path you can run signoff timing by running:

   ```
   setDelayCalMode –SIAware false
   setDelayCalMode –engine signalStorm
   timeDesign –signoff –si
   timeDesign –signoff –si –hold
   ```

   If you do not have a QRC license then run timing analysis with medium effort extraction:

   ```
   setDelayCalMode –SIAware false
   setDelayCalMode –engine signalStorm
   setExtractRCMode –effortLevel medium
   timeDesign –postRoute –si
   timeDesign –postRoute –si –hold
   ```

   If you see timing violations in the reg2out path group it's likely the IO latencies you created earlier with `update_io_latency` were not saved. This will be fixed in EDI 11 USR1. You can re-run `update_io_latency` followed by the `timeDesign` commands above to verify this.

7. Save the design:

   ```
   saveDesign DBS/postroute_hold.enc
   ```

8. As a final step export GDS by selecting **File – Save – GDS/Oasis…**

   - Specify **Output File:** *leon.gds*

9. Click **OK**. This will generate a default mapping file called streamOut.map and then export GDS based on this mapping file. You can open streamOut.map to view the example mapping file.

This concludes Workshop 1 of the EDI System tutorial.