cadence

# Basic Floorplanning in Innovus Rapid Adoption Kit (RAK)

**Click to get the test case**
**Floorplanning Lab data**

**Innovus Version 15.1**

**Nov – 2015**

Cadence Design Systems, Inc., 2655 Seely Avenue, San Jose, CA 95134, USA

## Cadence Trademarks



## Other Trademarks



## Confidentiality Notice

## Table of Contents

# Module 1: Introduction and Setup

## 1-1 Introduction

The Floorplanning Rapid Adoption Kit (RAK) introduces you to the basic floorplanning features of Innovus. It is very basic by design so we highly recommend users attend one of the several Innovus training classes provided by Cadence Educational Services. For more information on available training please visit www.cadence.com and click *Support & Training*.

Floorplanning involves defining the size of the chip or block, pre-placing hard macros, IO pads and other desired objects and defining a power grid for the design. Innovus floorplanning capabilities range from automated methods such as the Prototyping Foundation Flow and Automatic Floorplan Synthesis down to manual control with Relative Floorplanning and the Floorplan Toolbox which provides pinpoint control of macro placements. The automatic floorplan features allow you to converge on a floorplan quickly for designs containing hundreds of hard macros while the manual control provide the fine tuning required to finalize the floorplan.

In this module you will be introduced to the basic floorplanning features to place hard macros and complete the power plan resulting in a design that is ready for place and route implementation. You can continue your learning by referencing the following resources. If you are new to Innovus it's highly recommended you complete the *Intro to Innovus 15.1 and Block Implementation Flow RAK*:

- Floorplanning the Design chapter of the Innovus User Guide
- Automatic Floorplan Synthesis chapter of the Innovus User Guide
- Power Planning and Routing chapter of the Innovus User Guide
- Prototyping Foundation Flat Flow Rapid Adoption Kit (RAK)
- Intro to Innovus 15.1 and Block Implementation Flow (RAK)

## 1-2 Starting an Innovus Session and Importing the Design

1. Download and install the Innovus 15.1 software from http://downloads.cadence.com.

2. Extract the RAK database and change directory to the work directory:

```
linux% tar xfz Floorplanning151.tar.gz
linux% cd RAK_floorplanning_15.1
```

3. Verify the `innovus` executable is in your path by typing:

```
linux% which innovus
```

4. Invoke Innovus:

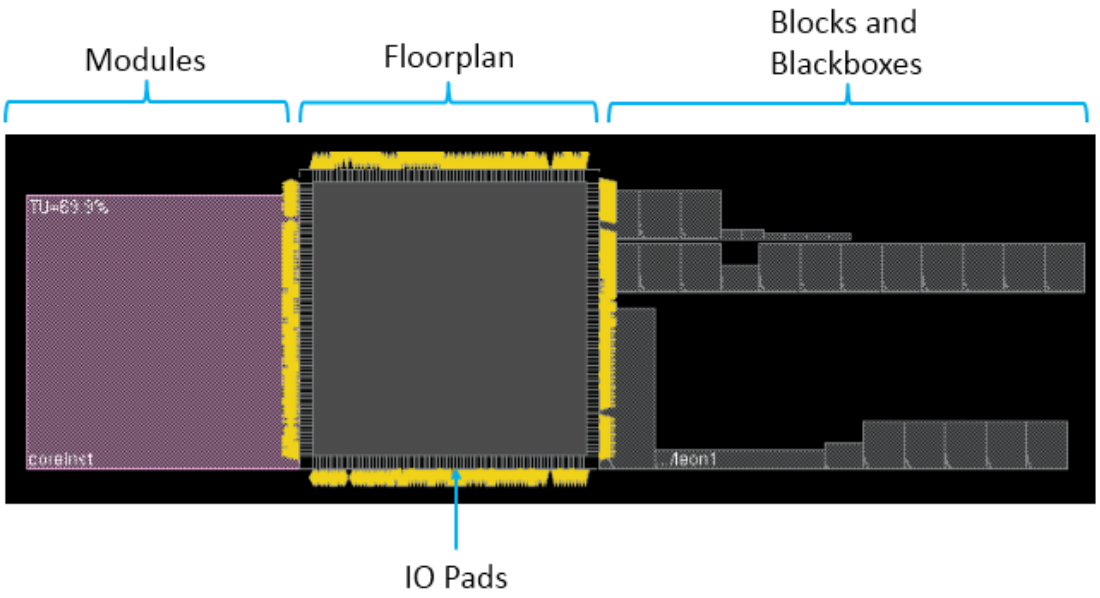
```
linux% innovus
```

# Module 2: Basic Floorplanning

## 2-1 Importing the Design

Import the Design.

1. Select **File - Import Design**.
2. Click the **Load** button on the Design Import form.
3. Open the file `DATA/asic_entity.globals`.
4. Click **OK** on the Design Import form to load the design.
5. Display the **Floorplan View** by selecting the icon in the upper right. 
6. Press the **F** key to fit the design.
7. Zoom out using the **Shift+Z** key until you can see all the objects shown below. Click and drag the Right Mouse Button (RMB) to zoom to the desired area.




After importing the design a basic floorplan is displayed. The IO pads are placed according to the IO placement file provided during Design Import. If one is not provided

then they are placed randomly. To the left of the floorplan are pink modules corresponding to the modules in the Verilog netlist. To the right of the floorplan are hard macros (blocks) and any blackboxes that have been defined.

## 2-2 Specifying the Floorplan

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**

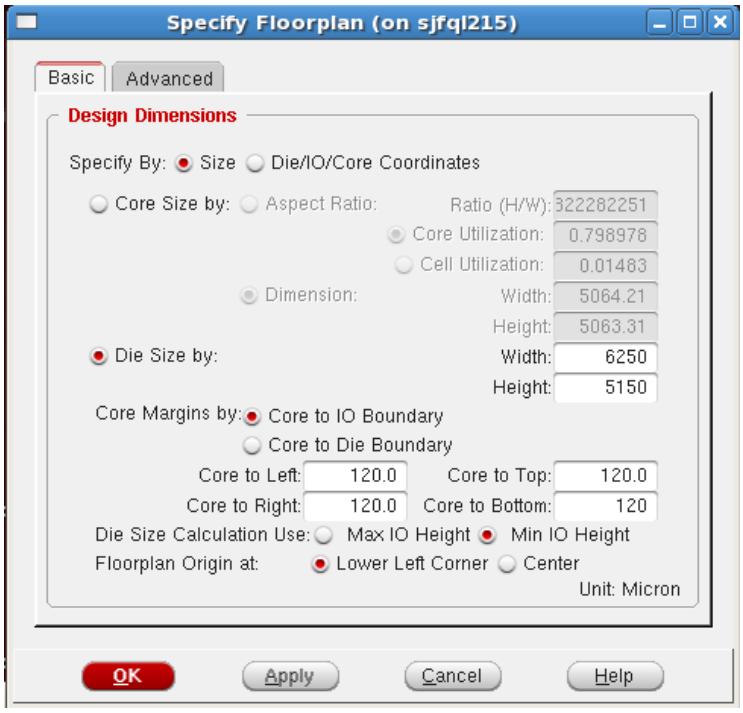
The Specify Floorplan form allows you to derive the floorplan size based on the target utilization or you can specify a specific size. For this design you will specify the size.

2. Select the radio button next to **Die Size By** to enable specification of the floorplan. Specify:

   **Width:** 6250
   **Height:** 5150

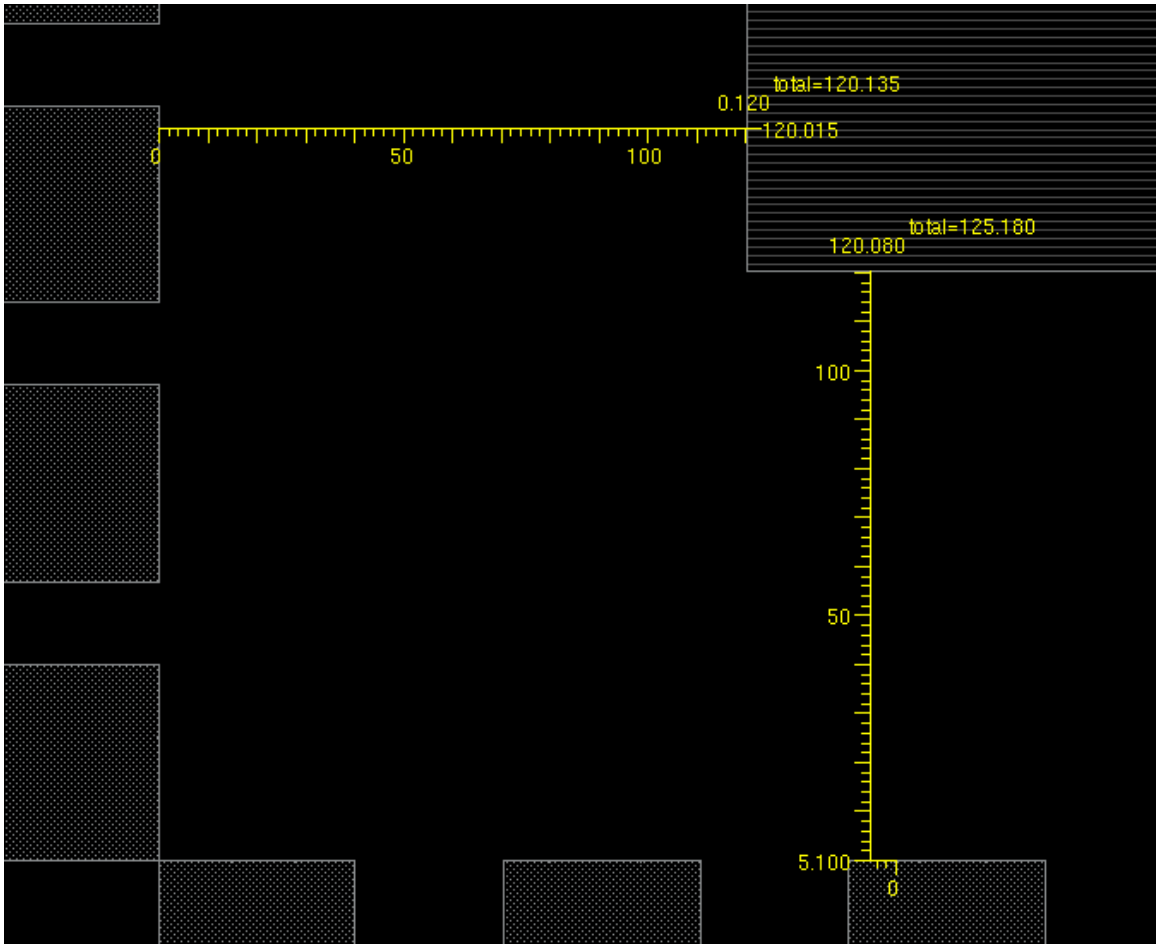
   Under the section of **Core Margins by: Core to IO Boundary**, make the following entries. These define the distance between the core box and IO box.

   **Core to Left:** 120
   **Core to Right:** 120
   **Core to Top:** 120
   **Core to Bottom:** 120




3. Click the **OK** button.

The floorplan is resized so the design has the specified size and there are 120 microns between the core box and IO box. Due to snapping you may see the 120um distance be slightly more by a fractional amount. You can use the ruler (**K**) to measure the distances shown below. Clear the rulers by pressing **Shift+K** and return to selection mode by pressing **A**.



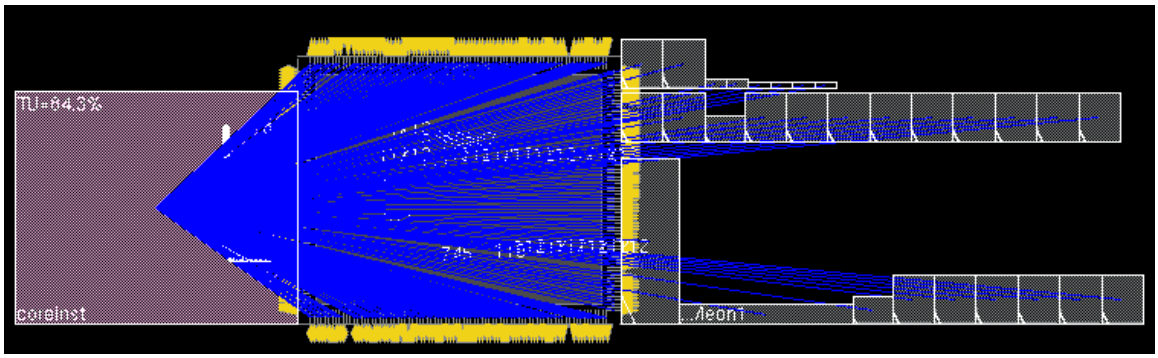

## 2-3 Moving and Editing Placement Constraints

In this section you will learn about placement constraint types and familiarize 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.

To the left of the floorplan are pink module(s) corresponding to modules in the Verilog netlist. Initially shown are the modules directly below the top level module. The module size is based on the target utilization of the modules using the area of hard macros and standard cells that are part of the module.

1. Click (LMB) on the large pink colored top-level module, *coreinst*, to select 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. You may need to zoom in to see the number on the flight line. The blocks are highlighted white because they are child instances of *coreinst*.

The **Hierarchy Down** widget  or enter the **Shift+G** keys (ungroup) is used to see modules lower in the hierarchy and Hierarchy Up (**G key**) to group them again. Ungroup three levels of hierarchy by doing the following:

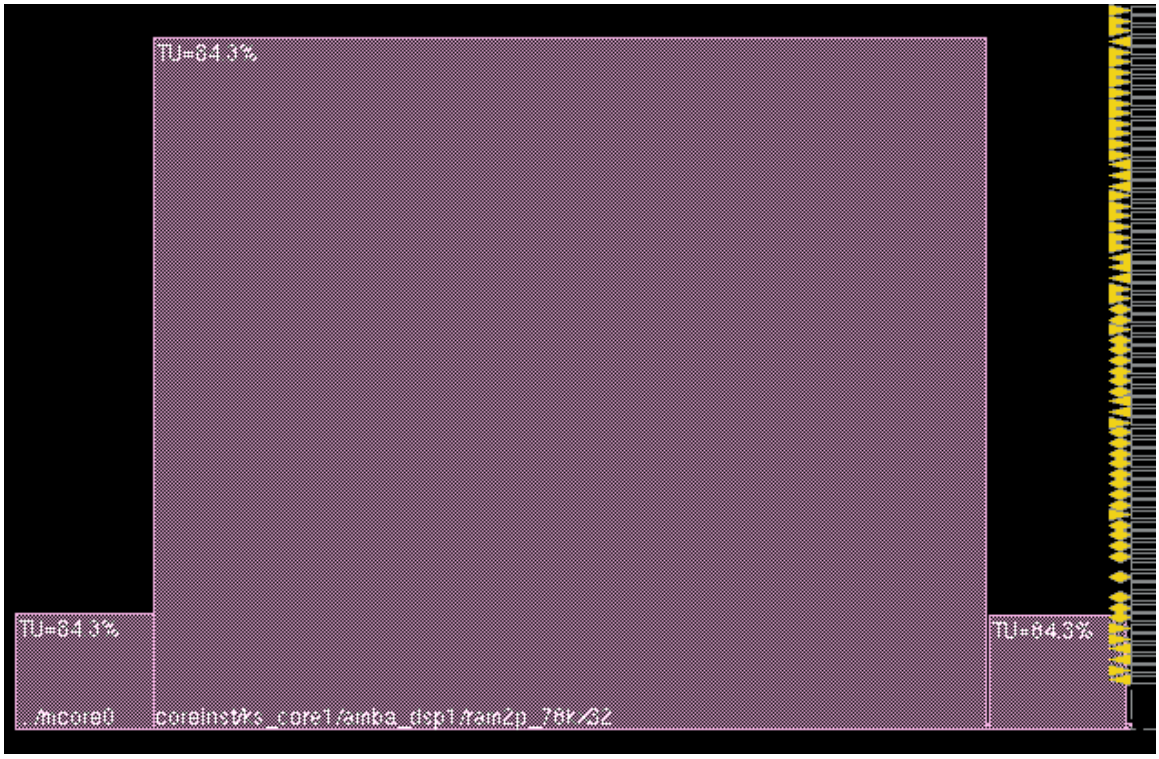
2. With *coreinst* selected press the **Hierarchy Down** widget or **Shift+G**. This reveals the next level module *coreinst/ks_core1*.

3. Deselect all by clicking in empty space or press **Ctrl+D** then select *coreinst/ks_core1* and press **Shift+G** to ungroup it.

4. Deselect all by clicking in empty space or press **Ctrl+D** then select *coreinst/ks_core1/amba_dsp1* and press **Shift+G** to ungroup it.

5. Deselect all by clicking in empty space or press **Ctrl+D**.

You can now see several pink modules which correspond to logic modules in the Verilog netlist.



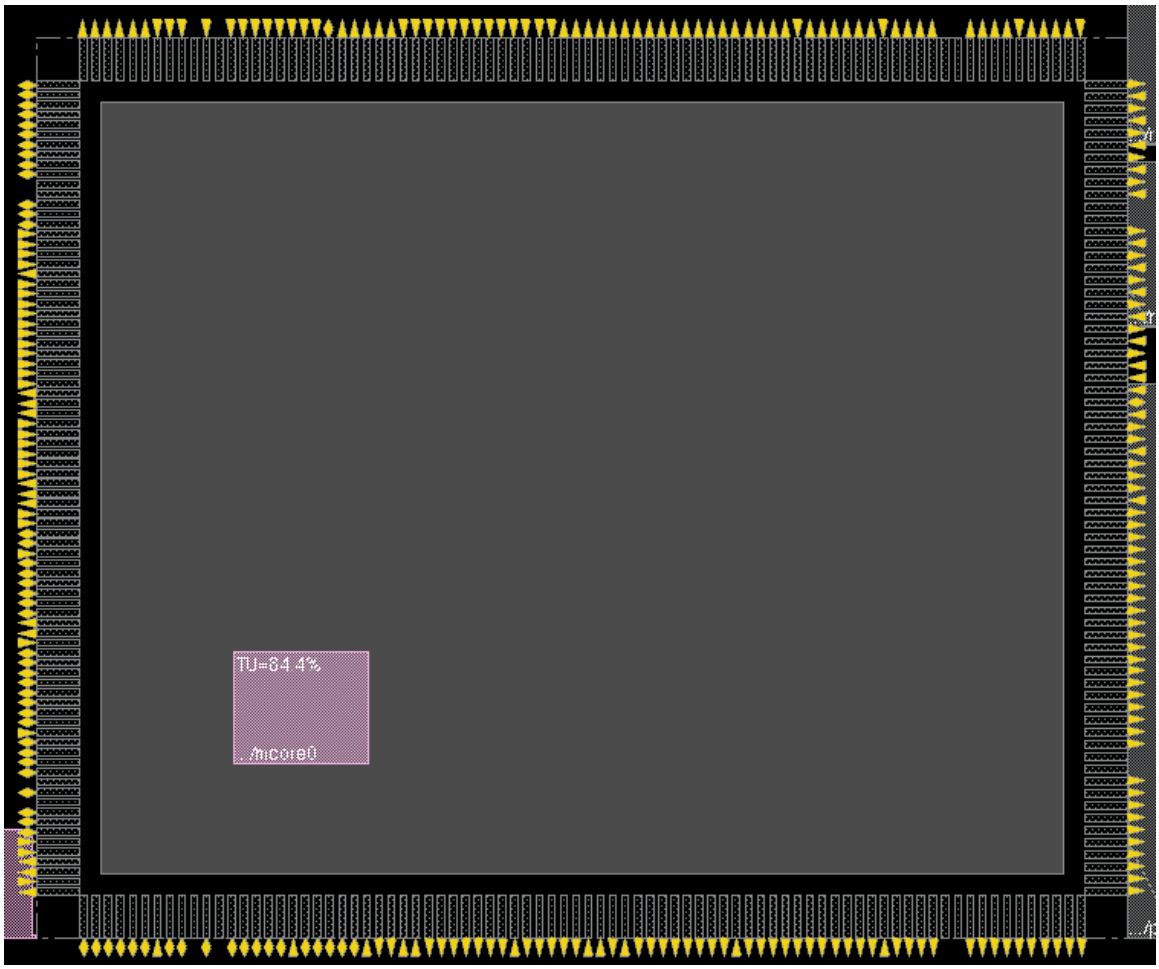

The Move/Resize/Reshape widget  is used to enable the movement of objects.

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

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.

7. Click (LMB) on the module *coreinst/ks_core1/amba_dsp1/mcore0* 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. Placing the module in the floorplan guides the placer to the location you want the logic placed. A guide is a soft constraint so logic belonging to the module does not need to be placed within the guide's boundary. The floorplan should like something like the following:




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

To reshape a module guide and to keep the area constant.

9. Select **View - 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.

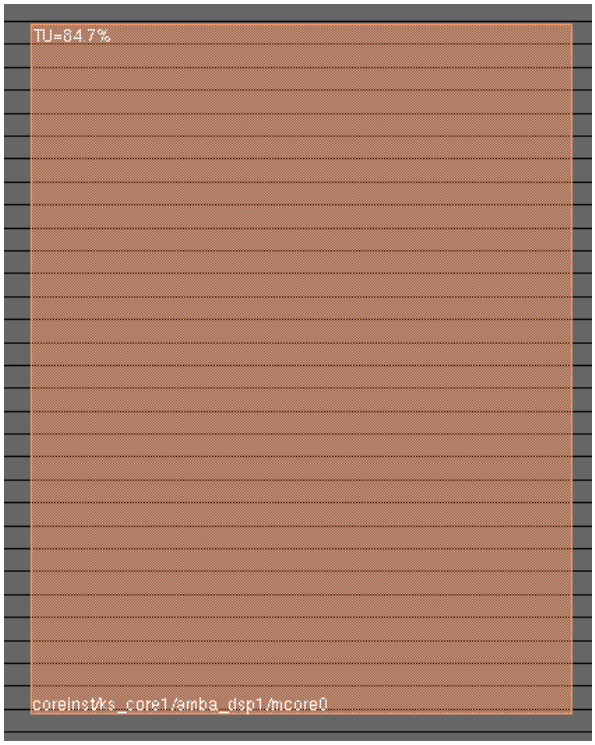
13. Click (LMB) to start the resize. Move the mouse to the desired location and click (LMB) again to complete the resize. Note the guide maintains its area as it is resized.

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.

14. Return to selection mode by pressing the **A** key.

15. Select the module which you placed in the core and open the Attribute Editor by pressing the **Q** key.

16. Change the **Constraint 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.

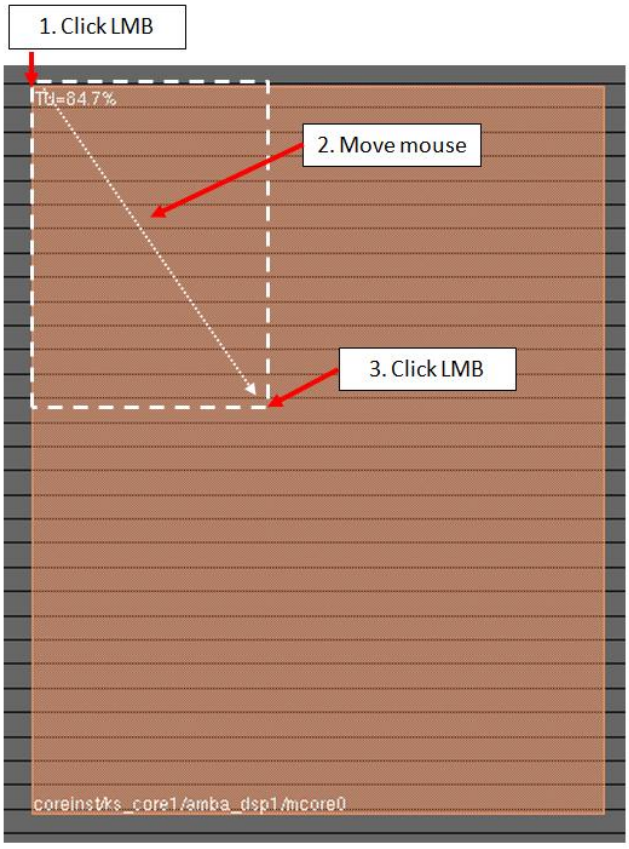
**Region** - This constraint is the same as a fence constraint except that instances from other modules can be placed within its physical outline by placement.

**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.

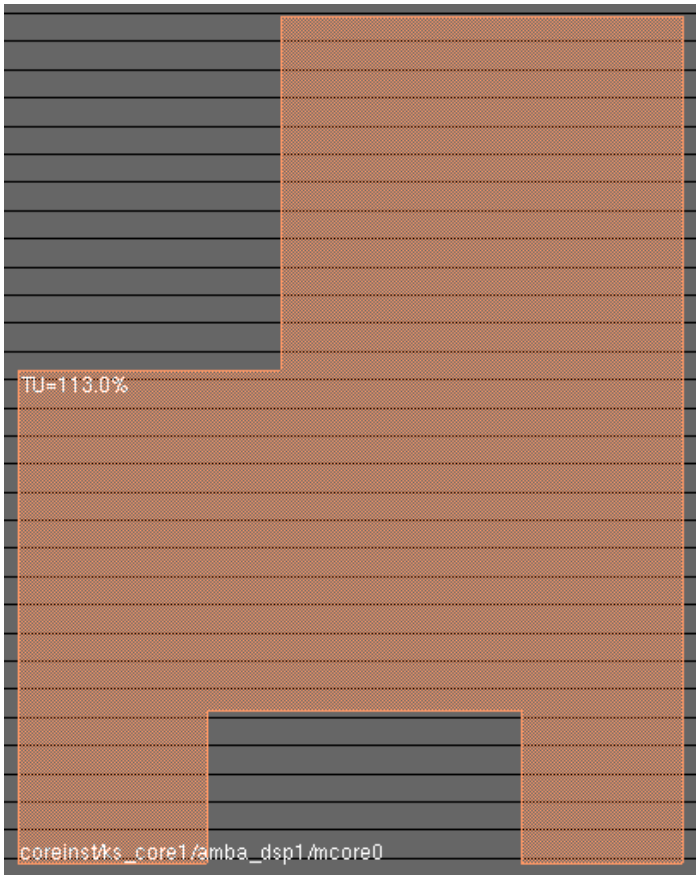
Next you will change the shape of the submodule to a rectilinear shape using the **Cut Rectilinear** widget. Make sure you still have a module placed in the core area.

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

18. 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.



19. 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.




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.

For this floorplan we want the placer to determine the best location for the standard cells logic and therefore want to remove all placement constraints by clearing the floorplan.

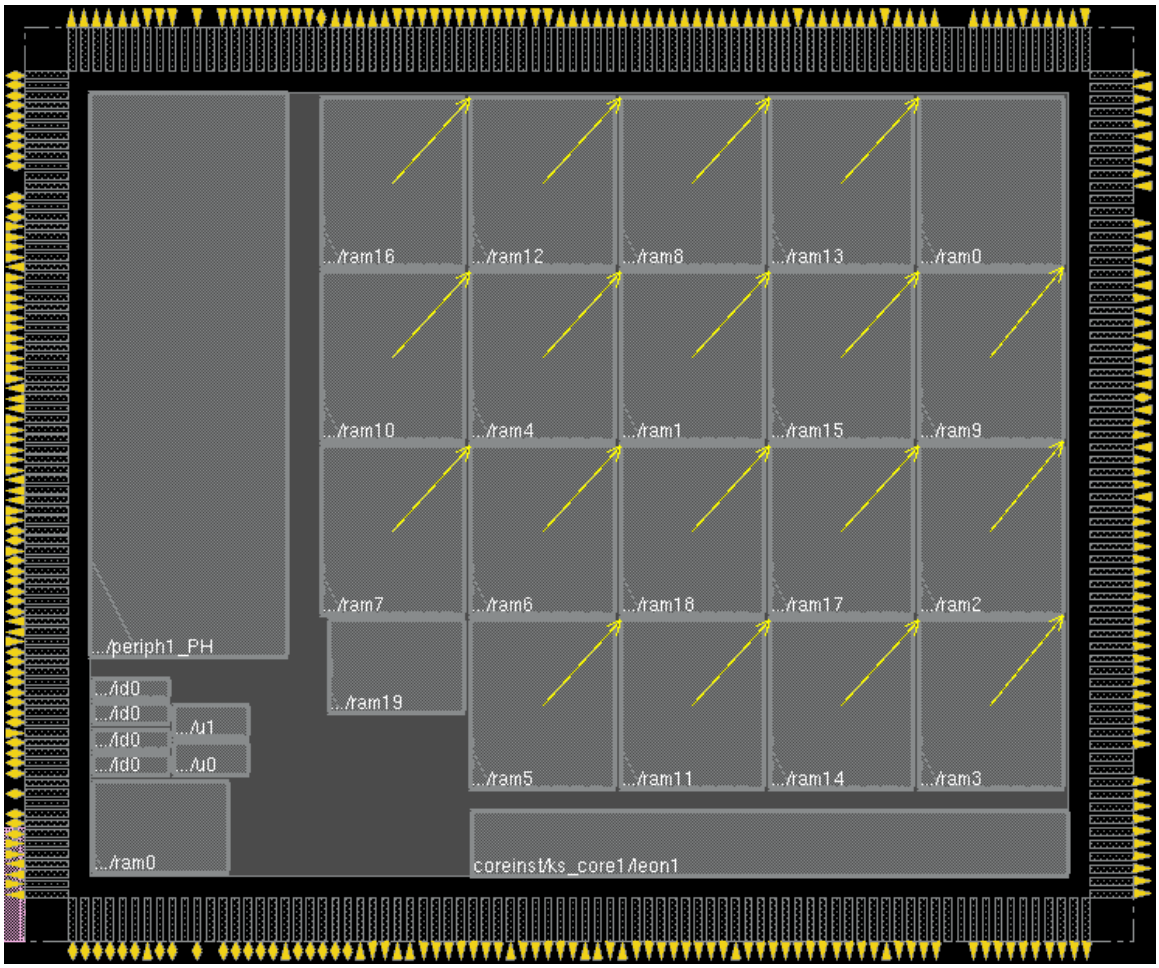
20. Select **Floorplan – Clear Floorplan**.

21. Select **All Floorplan Objects** and click **OK**.

22. Return to selection mode by pressing **A**.

## 2-4 Placing Hard Macros

In this section you will use several different methods to place blocks to achieve the floorplan below. This section utilizes Relative Floorplanning and manual movements which are useful for designs with a small number of blocks. For designs with a hundred or more blocks consider using Automatic Floorplan Synthesis or the Prototyping Foundation Flow to quickly converge on a floorplan.



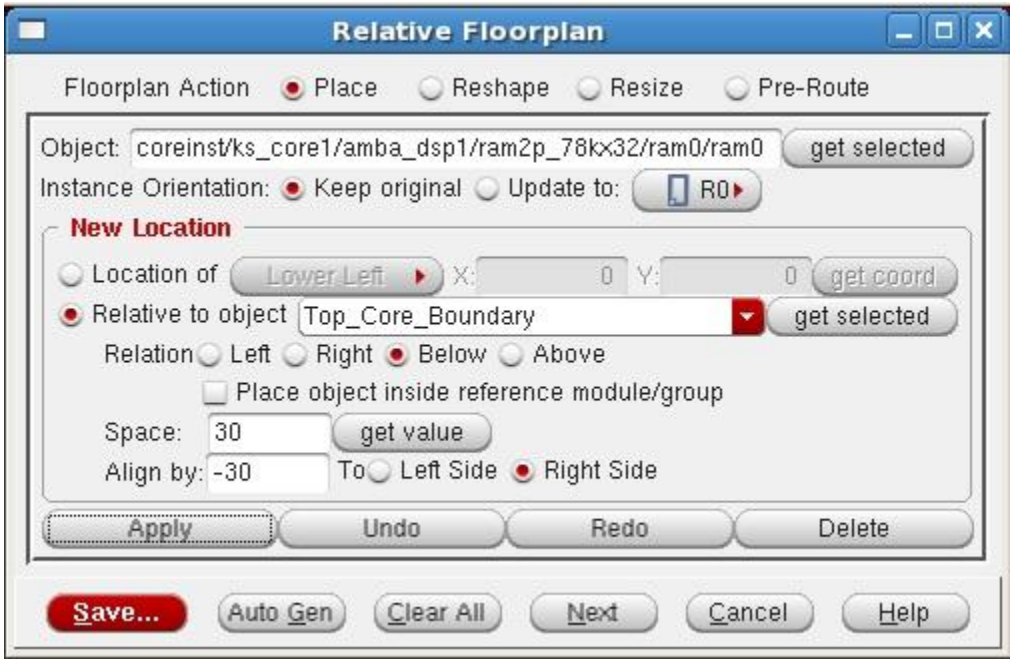

When a block is selected / highlighted, the flight lines are displayed to each pin of the block instead of the center of the block. This feature helps in determining the proper orientation of a block. For example, it helps with route congestion and timing when most of the block pins face the center of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

can turn off the display of flight lines by expanding **Miscellaneous** at the bottom of the Layer Control bar and deselecting **Flight Line**.

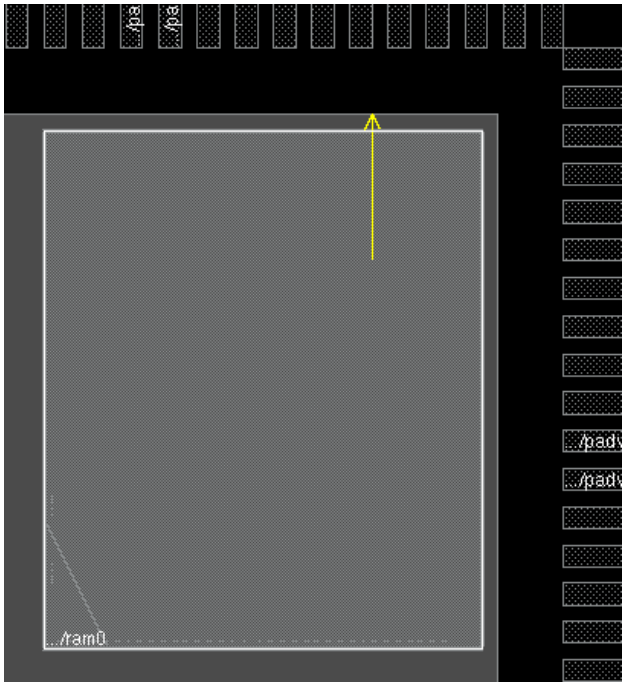
The **Relative Floorplan** feature allows you to place macros relative to other reference objects. When the reference object moves, so will the macros with relative placement constraints to the object. Use relative floorplan constraints to place the array of blocks in the upper right.

1. First, place the anchor object for the array in the upper right corner of the core by selecting **Floorplan – Relative Floorplan – Edit Constraint**.

2. Fill out the form as shown below to place the specified instance 30um below the top core edge and -30um from the right core edge.. You can quickly select the *ram0* instance by typing "`selectInst *ram0/ram0`" then use the **get selected** button to populate the Object field with the selected instance.




Click **Apply** and observe the macro is placed 30um below the top core boundary and -30um from the right core boundary. The yellow arrow indicates there is a relative constraint between the block and top core boundary:




3. Click **Cancel** on the Relative Floorplan window.

4. Next, you will place the remaining macros of the array using an array constraint. First, select the macros in the array by executing the following script at the Innovus prompt:

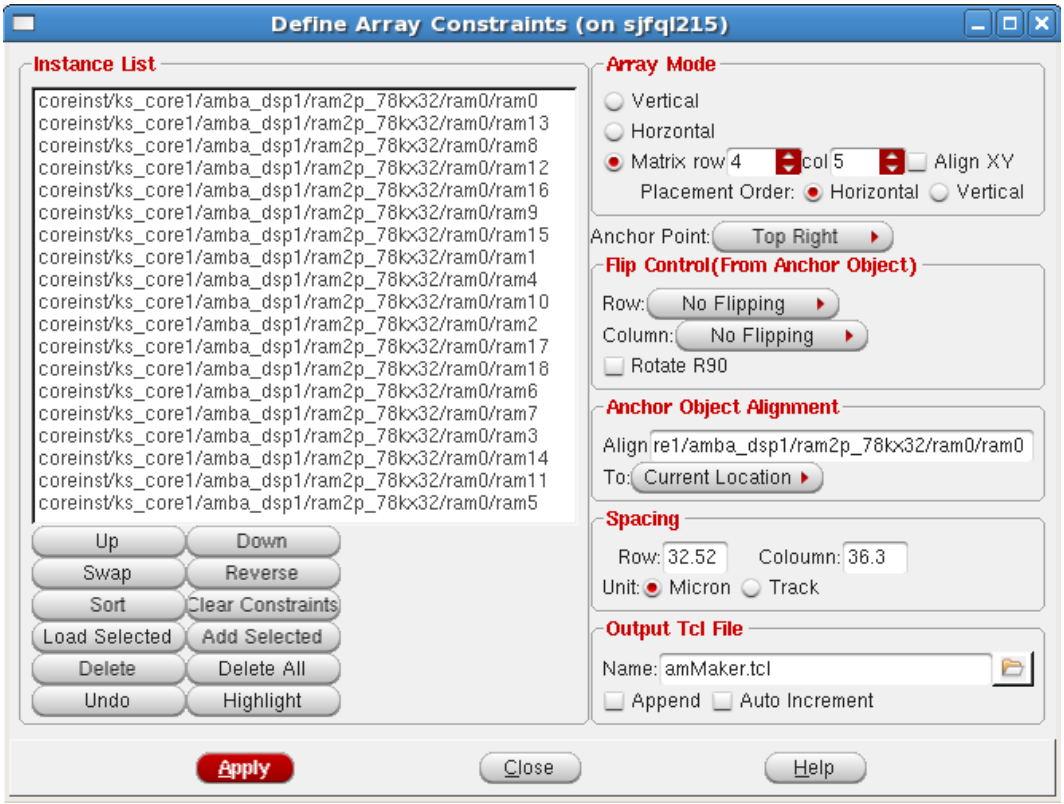
```
source SCRIPTS/selectBlocks.tcl
```

5. Select **Floorplan – Relative Floorplan – Define Array Constraint**.

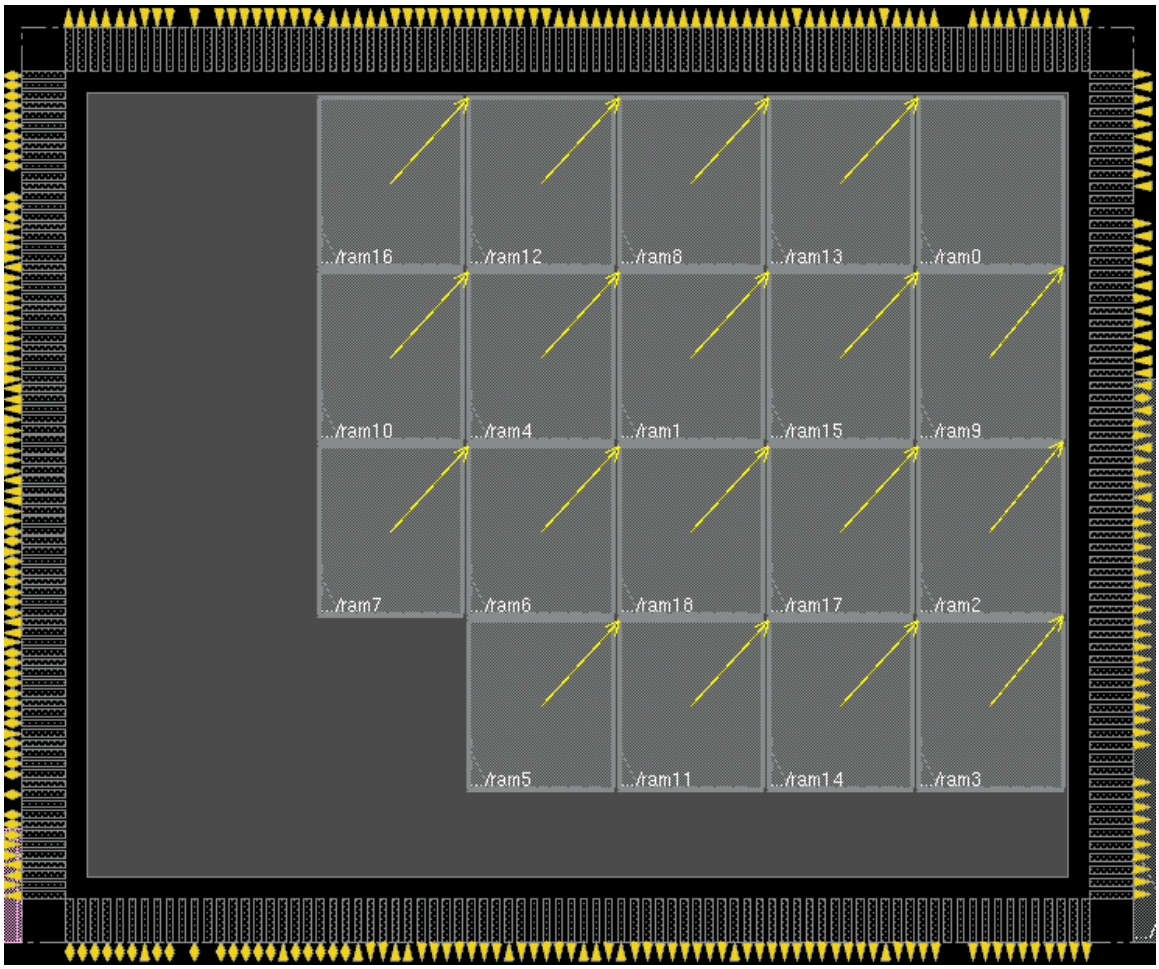
6. Click the **Load Selected** button to specify the instances in the array.

7. Fill out the form as follows:

   **Matrix row** 4 **col** 5
   **Anchor Point:** Top Right
   **Align:** coreinst/ks_core1/amba_dsp1/ram2p_78kx32/ram0/ram0.
   **Spacing Row:** 32.52 **Column:** 36.3 **Unit:** Micron




8. Click **Apply** and the blocks should be placed as shown below. Click **Close** on the Define Array Constraints form.




Objects can be placed at specific coordinates using the Attribute Editor or the `placeInstance` command.

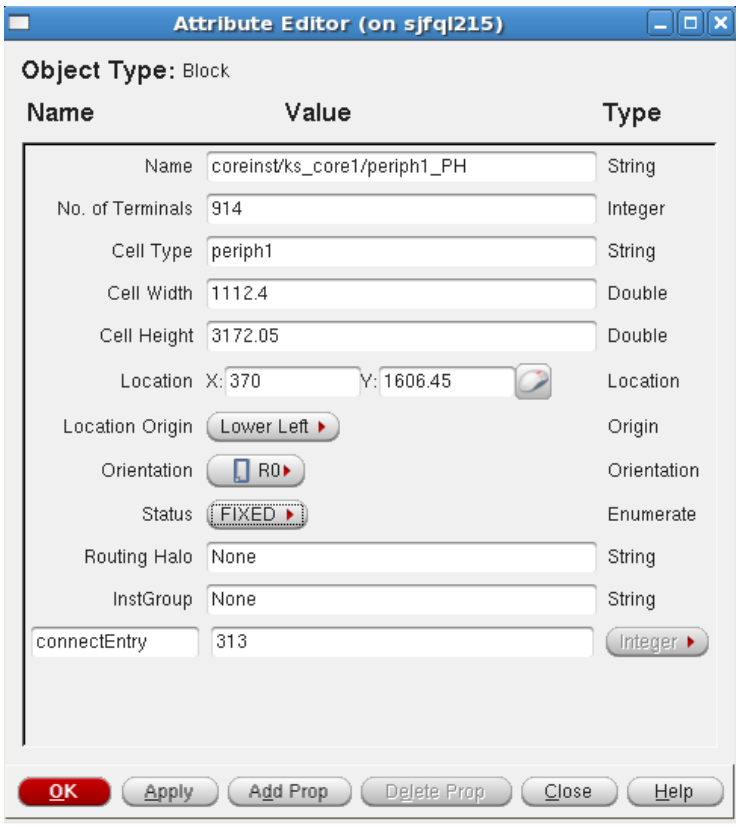
9. Select the macro *coreinst/ks_core1/periph1_PH*. You can click on it or type the following:

```
deselectAll
selectInst *periph1_PH
```

10. Type the **Q** key to open the Attribute Editor.

11. On the Attribute Editor make the following changes to specify the location and placement status of the macro.

    **Location X:** 370.0 **Y:** 1606.45
    **Orientation:** R0
    **Status:** FIXED




12. Click **OK** on the Attribute Editor and observe the macro is placed in the upper left corner of the core area at the location you specified.

13. Place the instance *coreinst/ks_core1/leon1* by entering the following `placeInstance` command at the encounter prompt.

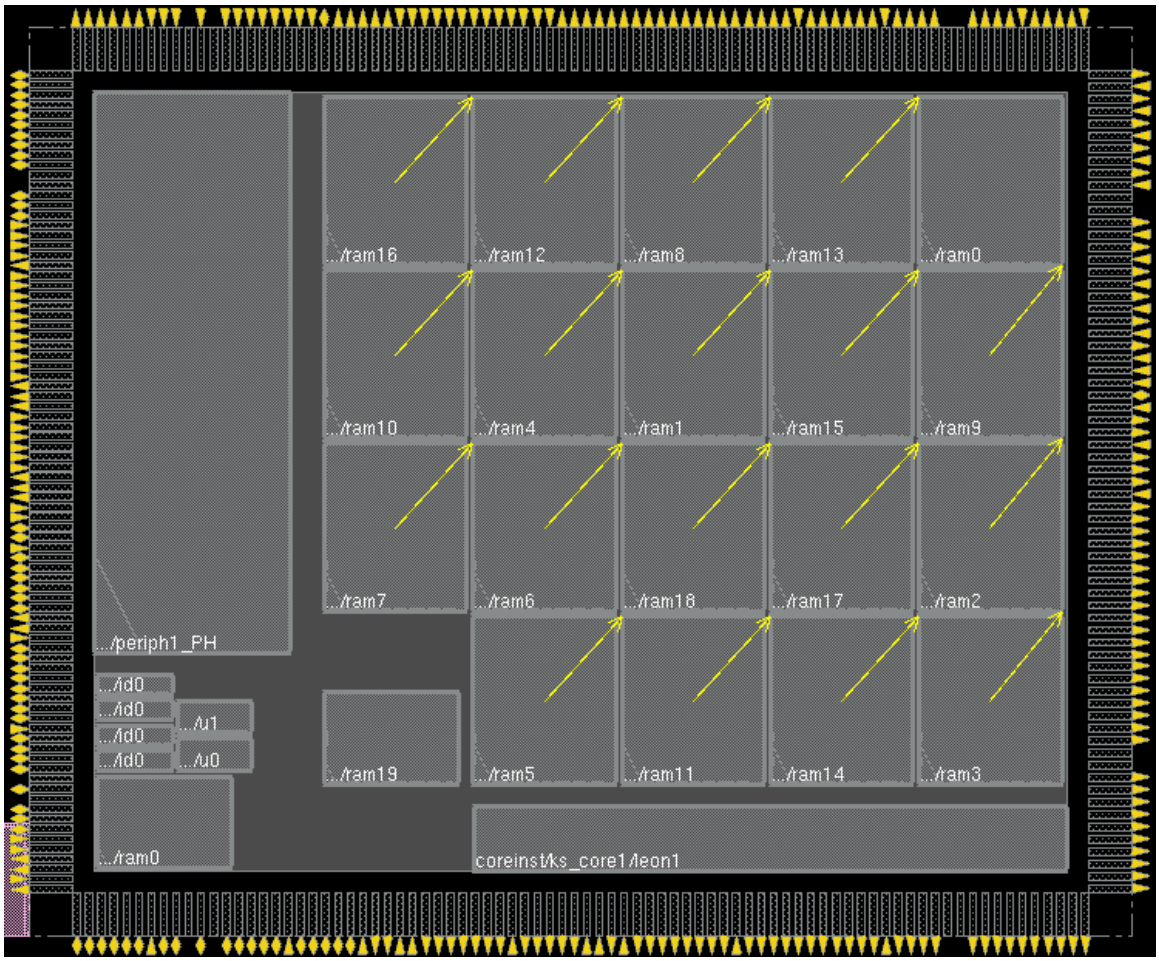
```
placeInstance coreinst/ks_core1/leon1 \
{2518.8 370.12} R0 -fixed
```

14. Redraw the floorplan (**Ctrl+R**) and observe this block is placed at the specified location in the bottom right corner of the core. You can select and zoom to it by running the following:

```
deselectAll
selectInst coreinst/ks_core1/leon1
zoomSelected
deselectAll
```

15. Run the following command to place the remaining macros. The floorplan should look like the picture below:

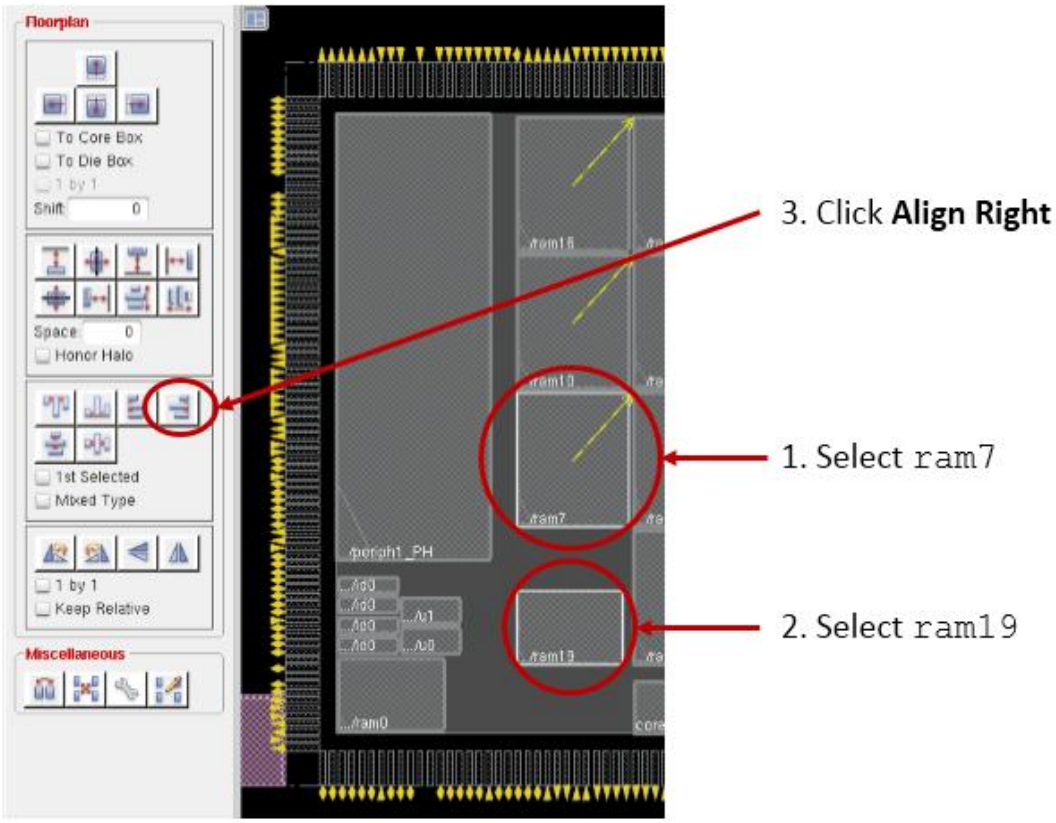
```
source SCRIPTS/placeInst.tcl
```



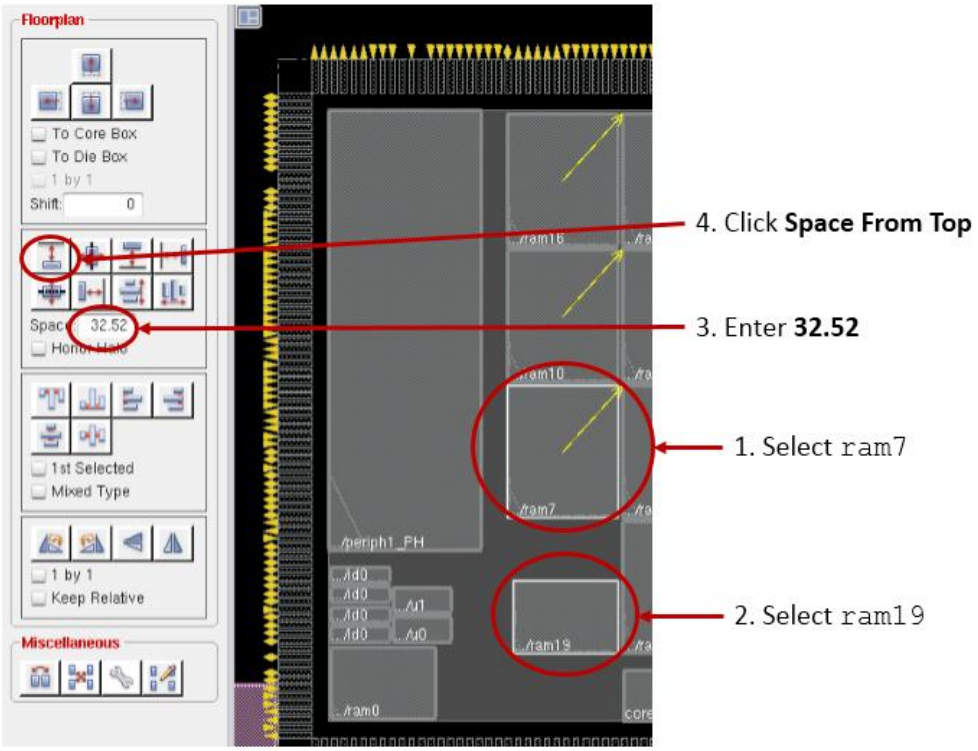

Next use the **Floorplan Toolbox** to fine tune the placement of the *ram19* macro. The Floorplan Toolbox provides several useful floorplanning commands in a single form.

16. Open the Floorplan Toolbox by selecting **Floorplan – Floorplan Toolbox**.

17. Follow the steps in the picture below to align the right edge to the block above it:



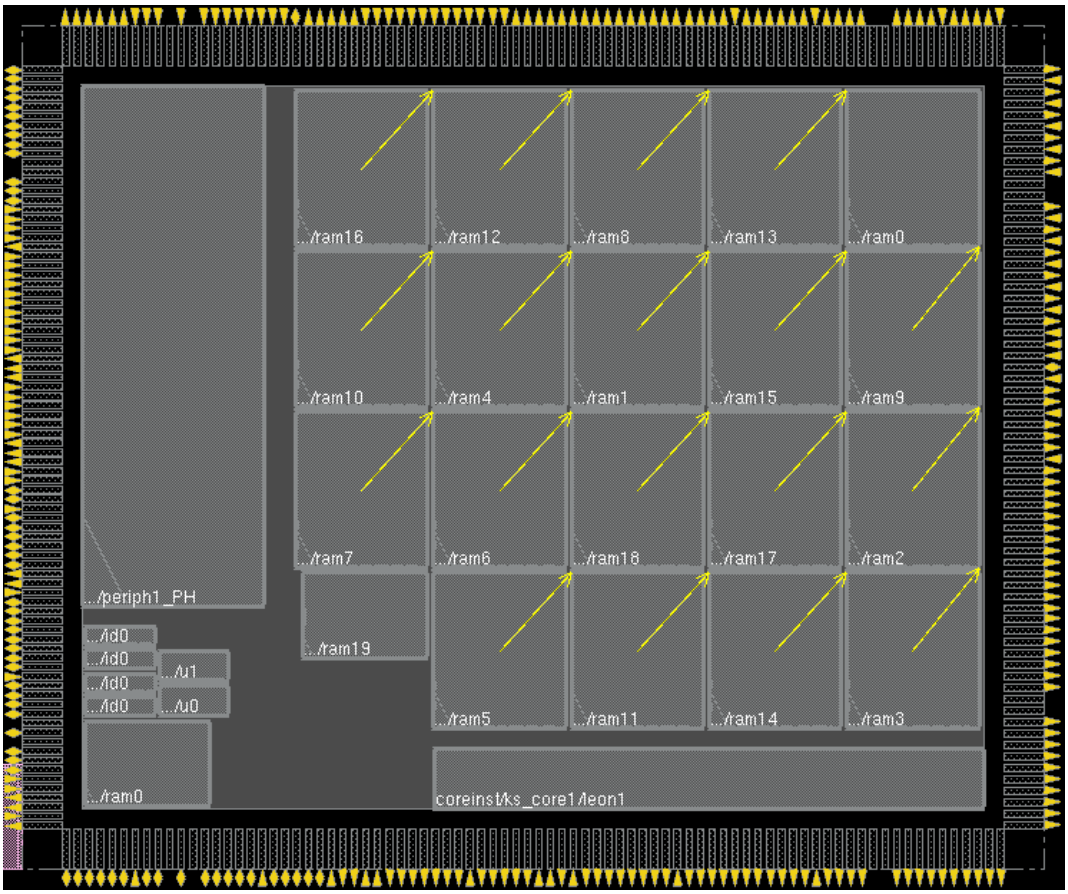

18. Space *ram19* 32.52um below *ram7* by following the steps below:



19. Close the Floorplan Toolbox by clicking on its tab in the upper right of the toolbox.

You have now completed the block placement and your floorplan should look like the one below. If your design does not match the one below load in an existing floorplan by entering the following command:

```
loadFPlan DATA/blocks.fp
```




20. Save the design:

```
saveDesign DBS/blocks.enc
```

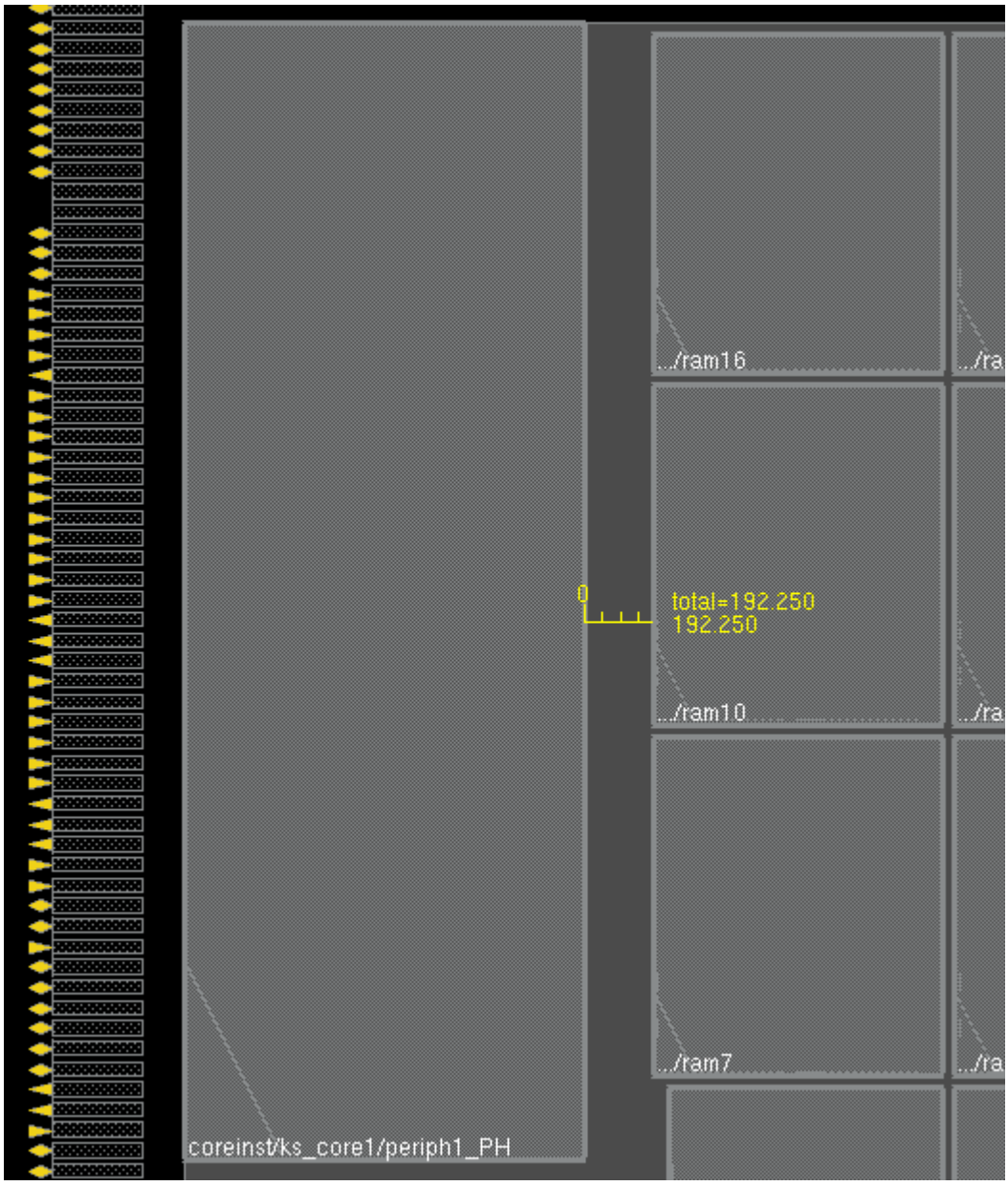
**Tip:** The command `writeFPlanScript` can be run to generate a script to generate the existing floorplan. This is useful if you want to avoid manually re-creating the floorplan by hand. Use the `-sections` options to output only commands to re-create certain objects. For example:

```
writeFPlanScript -fileName fp.tcl -sections {blocks}
```

## 2-5 Resizing the Floorplan

**Resize Floorplan** allows you to shrink or expand the floorplan in either direction while automatically adjusting the block placement. Use Resize Floorplan to shrink a vertical channel by 100um.

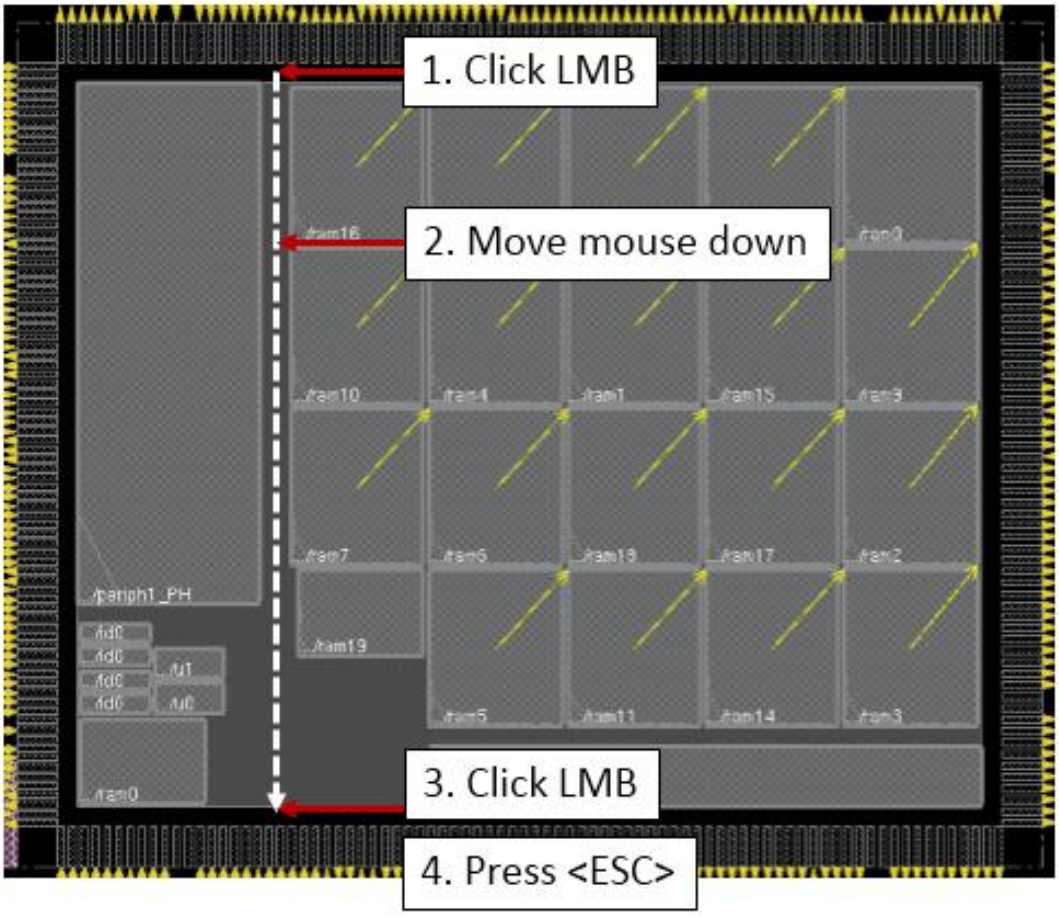
1. First, measure the channel to the right of the *periph1_PM* block as shown below using the ruler:



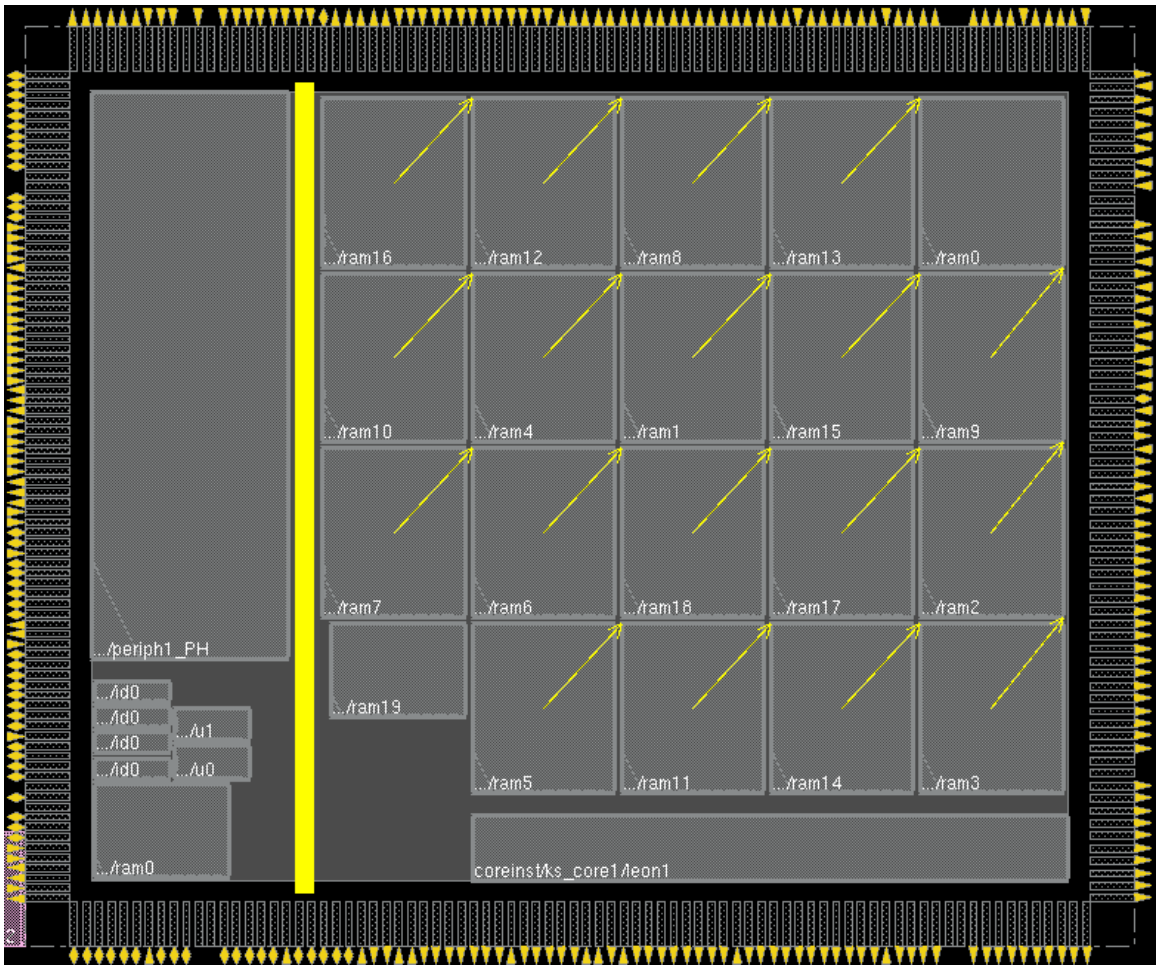

2. Select **Floorplan – Resize Floorplan**.

3. On the Resize Floorplan form:

- Select **Mode:** *Shift Based*
- Specify **Shrink in X Direction** and specify a value of **100um**.
- Select **Based on Resize Line**.
- Make sure you're in **Select** mode then click the **Draw** button. Draw a line where to shrink the floorplan as shown below:

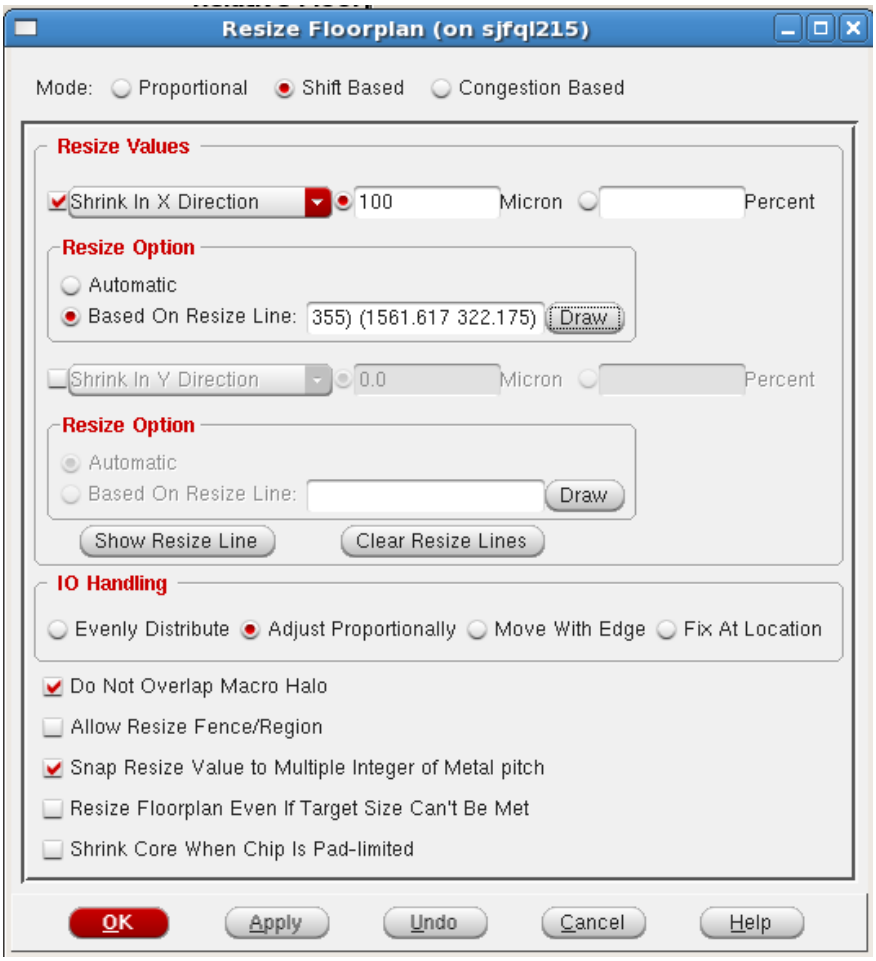



After you press ESC the resize line will be drawn in yellow with a width equal to the width to expand or shrink the floorplan. In this case the line is 100um wide.

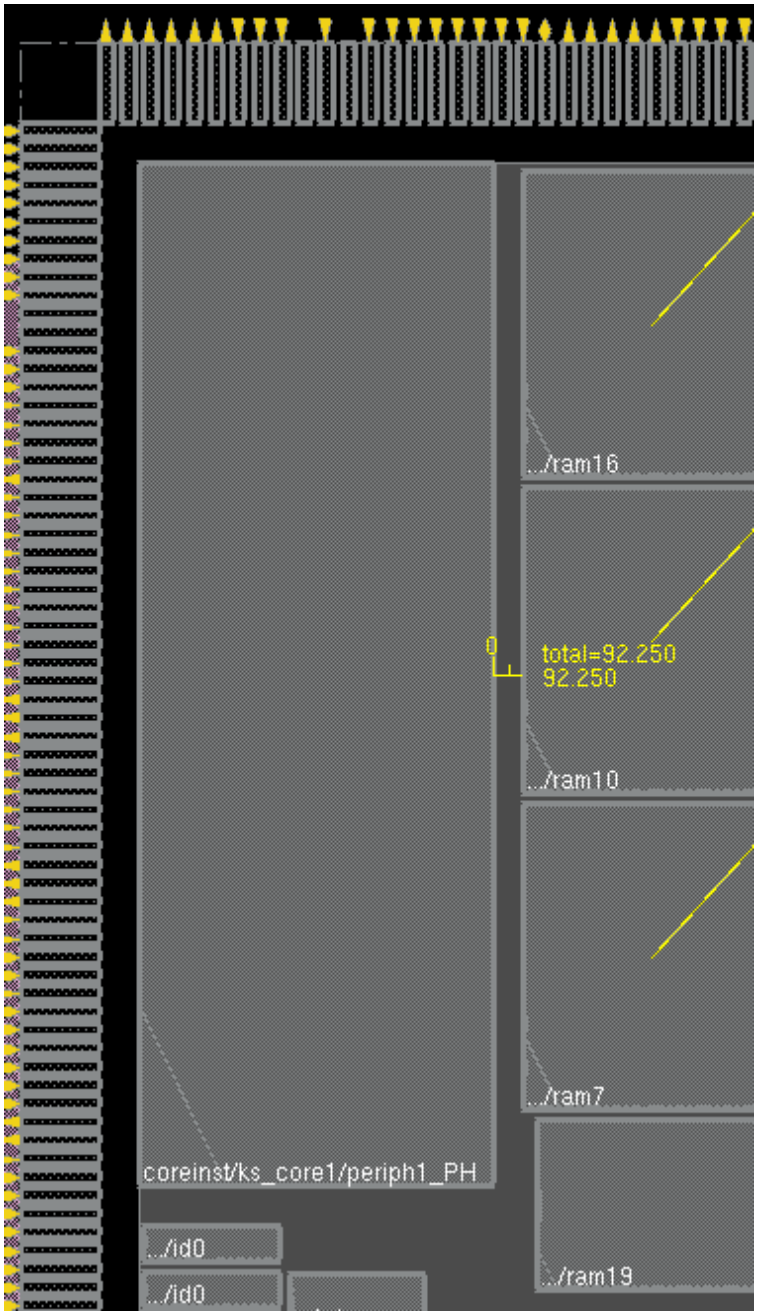



**Tip:** The resize line is drawn by a series of clicks with LMB and moves. The line can have multiple segments. You can specify the exact channels to shrink by drawing the resize line in the desired channels. Press the **ESC** key to complete the line.

The Resize Floorplan form should appear as below.




4. Click **OK** to resize the floorplan and measure the channel again to confirm it is 100um smaller.



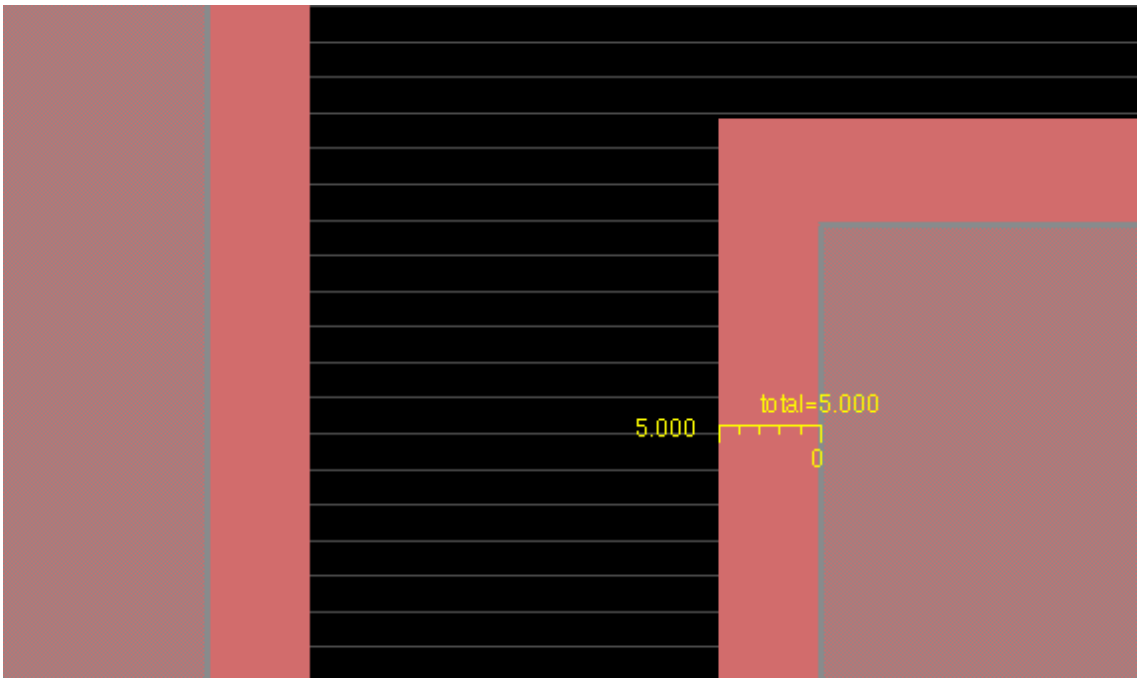

## 2-6 Placement and Routing Blockages

Placement blockages are used to control placement of cells in specified areas.

1. Placement halos are placement blockages around blocks which prevent cells from being placed inside the halo area. Select **Floorplan – Edit Floorplan – Edit Halo**.

2. On the Edit Halo form add a placement halo of 5um around all blocks by filling out the form as follows and clicking **OK**.

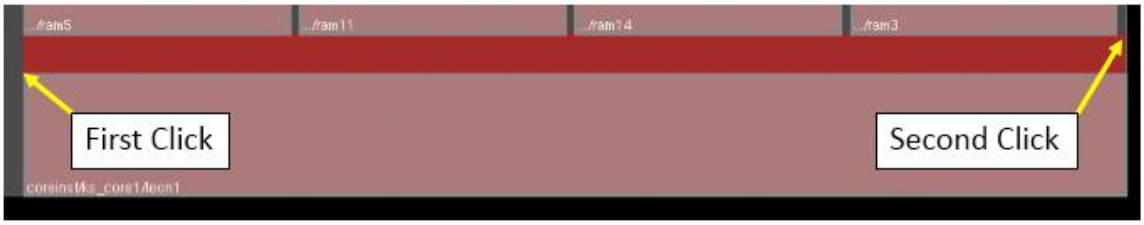

Zoom into the blocks and observe the orange halo extends 5um beyond the block boundary. A halo is associated with a block so if the block is moved, the halo moves with it.




A placement blockage prevents cell placement in a specific area but unlike a halo it is not associated with any block.

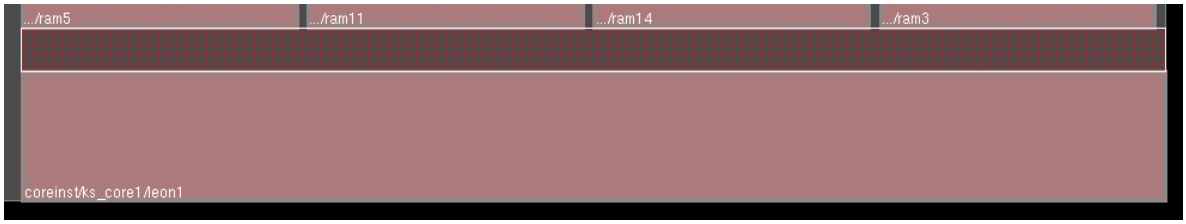
3. Select the **Create Placement Blockage** widget  or press **Shift+Y.**

4. Draw a placement blockage over the channel above the `leon1` block as shown below. Make sure the blockage slightly overlaps the blocks above and below the channel.




5. Change to selection mode by pressing **A** then double-click on the placement blockage you drew to open the **Attribute Editor**.

6. Notice the Type is set to Hard which means no standard cells can be placed in this area. This is a long channel and nets going through it may require buffering so change the type to **Soft** and click **OK**.




Notice the shape turns to a red mesh indicating it is a soft placement blockage. A soft placement blockage prevents global placement from placing cells in the area but allows placement refinement, timing optimization and clock tree synthesis to place cells in this area as needed.

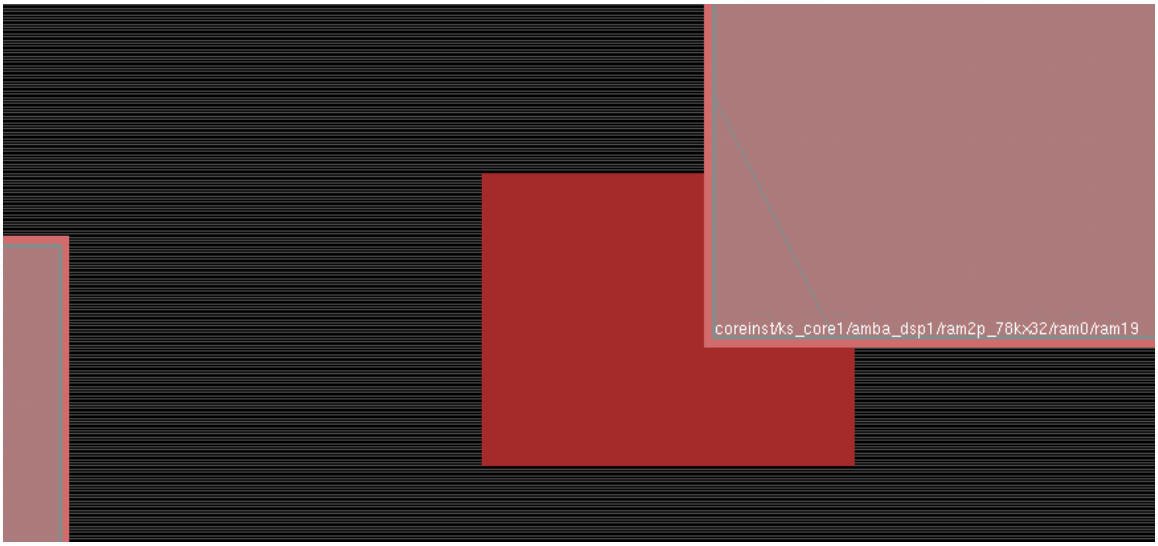
The valid placement blockage types are:

- **Hard** - The area cannot be used to place blocks or cells. This is the default.
- **Partial** - Sets a percentage for the maximum cell utilization in this area. 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 Yplacement, 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.

A partial placement blockage can reduce routing congestion by setting a maximum placement density in a specified area. For this design a partial placement blockage is added near the lower left corner of the *ram19* instance.

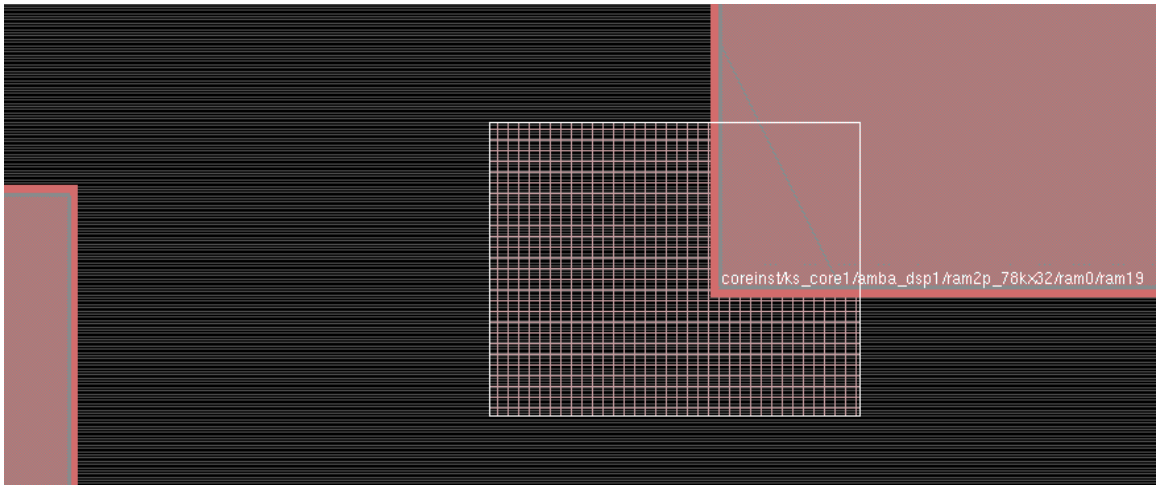
7. Select the **Create Placement Blockage** widget  or press **Shift+Y.**

8. Draw a blockage over the lower left corner of *ram19* as shown below:



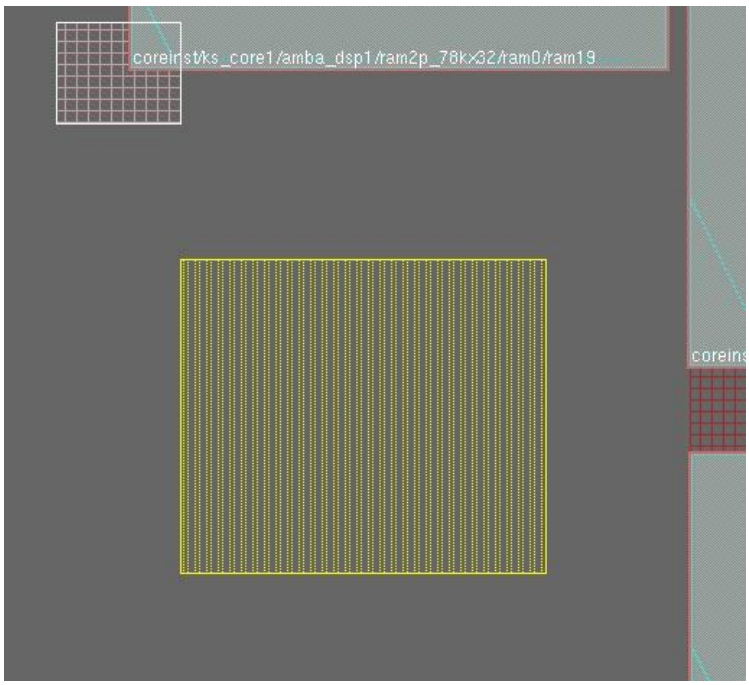

9. Change to selection mode by pressing **A** then double click on the blockage to open the **Attribute Editor**.

10. Change the **Type** to **Partial** and specify **Placement Percentage** of **30%**. This means the maximum cell density allowed in this area is 30%. Click **OK**. Observe the shape is now a pink mesh to indicate it is a partial placement blockage.




Routing blockages prevent routing in an area for specified layers. The router must also space all routing away from the routing blockages based on the spacing rules in the technology file.

11. A routing blockage is created in a similar manner to a placement blockage. Select the **Create Routing Blockage** widget  and draw a rectangle similar to the yellow routing blockage in the picture below.




12. Press the **A** key for selection mode and double-click on the routing blockage you created to open the Attribute Editor.

13. On the Attribute Editor observe you can specify the layers prevented from routing in this area. Click **Close** on the Attribute Editor.

14. This design does not need the routing blockage so delete it by selecting the routing blockage and pressing the **Delete** key.

## 2-7 Finishing the Floorplan

The **Finish Floorplan** command adds placement and routing blockages globally to the design saving you time from having to add them individually.

1. Select **Floorplan – Automatic Floorplan – Finish Floorplan**.

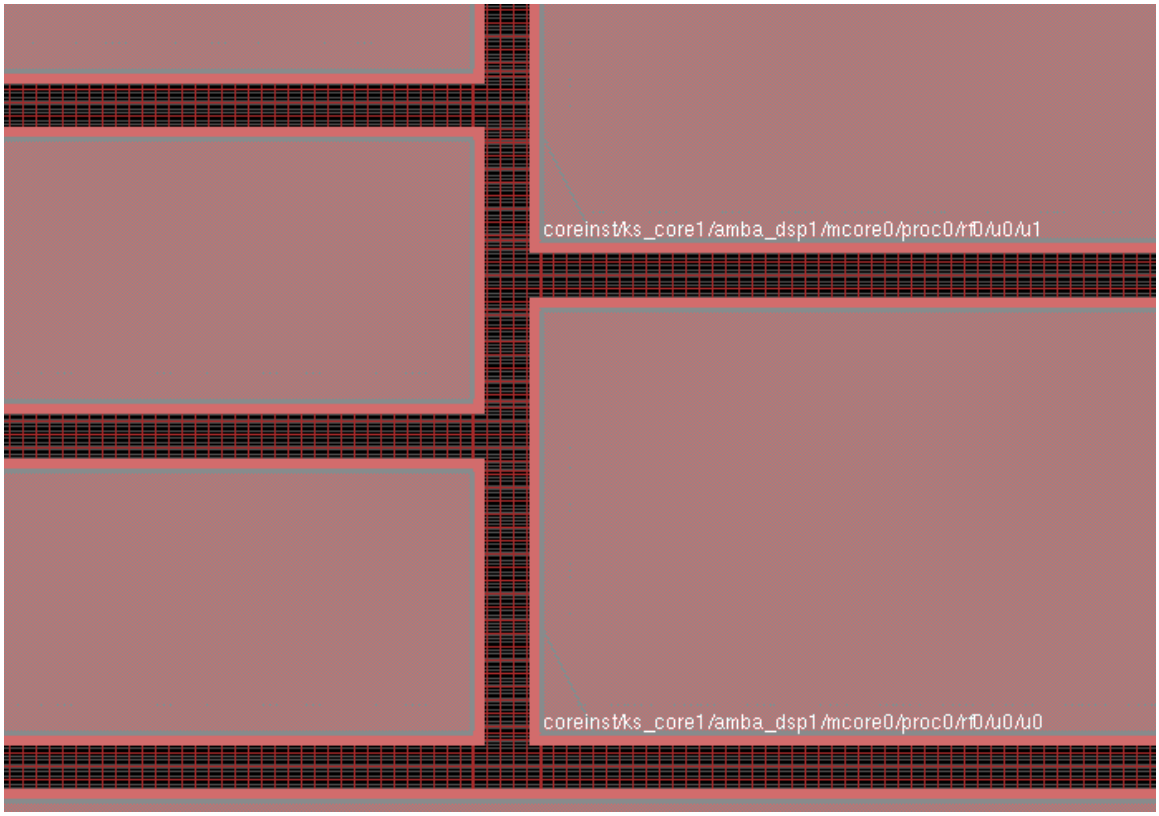
2. On the **setFinishFPlanMode** tab in the **Active Objects** section select **Core**, **Macro** and **Soft Placement Blockage**.


3. Select the **FinishFloorplan** tab.

4. Add soft placement blockages in channels that are 40um or less by specifying the following:

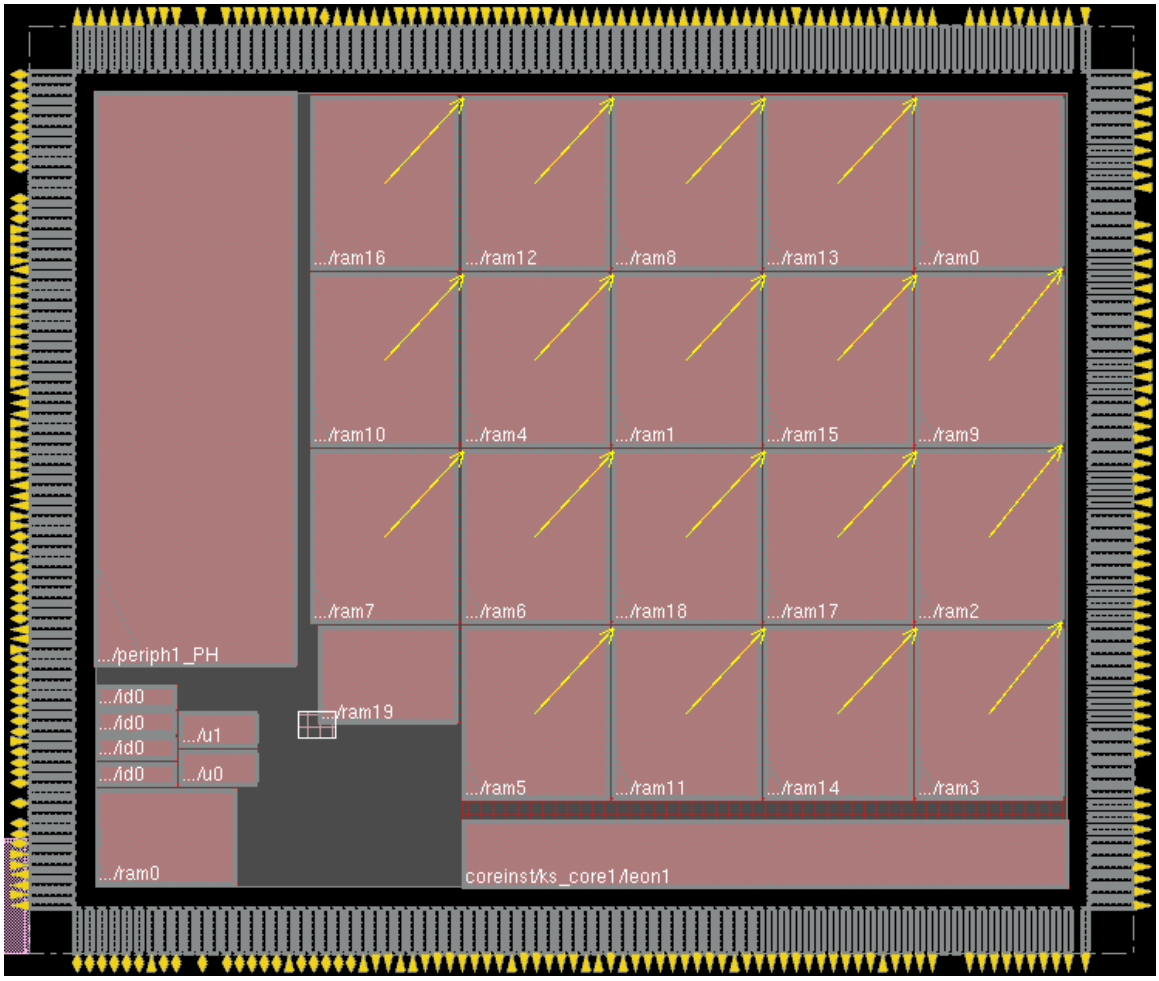
    - Select **Add Placement Blockage**
    - Set **Blockage Type** to **Soft**
    - Set **Max Gap** to **40um**


5. Click **OK** on the Finish Floorplan form. Zoom into the small channels and observe soft blockages are added.




Your floorplan should now look like the picture below. If it does not, load in an existing floorplan using the following command:

```
loadFPlan DATA/floorplan.fp
```




6. Save the design with the following command:

```
saveDesign DBS/floorplan.enc
```

# Module 3: Power Routing

## 3-1 Creating Power and Ground Rings and Stripes

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 for this design 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 |

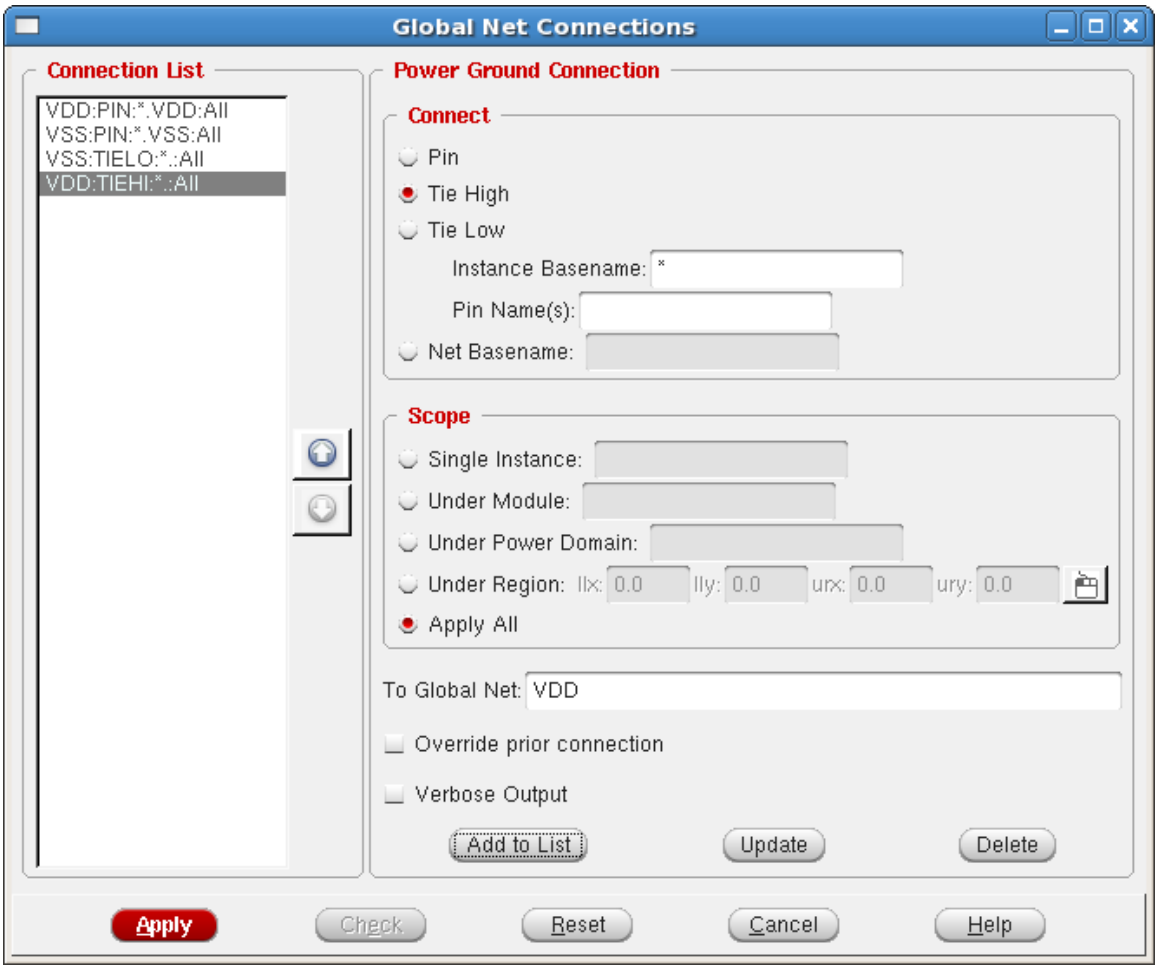
n/s = not selected

Set 1 and 2 indicate that instance pins VDD and VSS on all instances connect to the global power nets VDD and VSS. Set 3 and Set 4 indicate the tielo and tiehi connections (1'b0 and 1'b1) connect to VSS and VDD.

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 Innovus console:

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

   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.



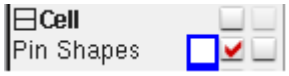

Click the **Apply** button when all 4 sets appear correct in the **Connection List**. This will apply the connections to the netlist.

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

## 3-2 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 and around specified blocks. Power and ground stripes are also added to create the power grid. The Add Ring and Add Stripe commands are used to create the power grid.

1. Make instance pins visible. This is done in the Layer Control bar 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.

To add the power rings around the core do the following:

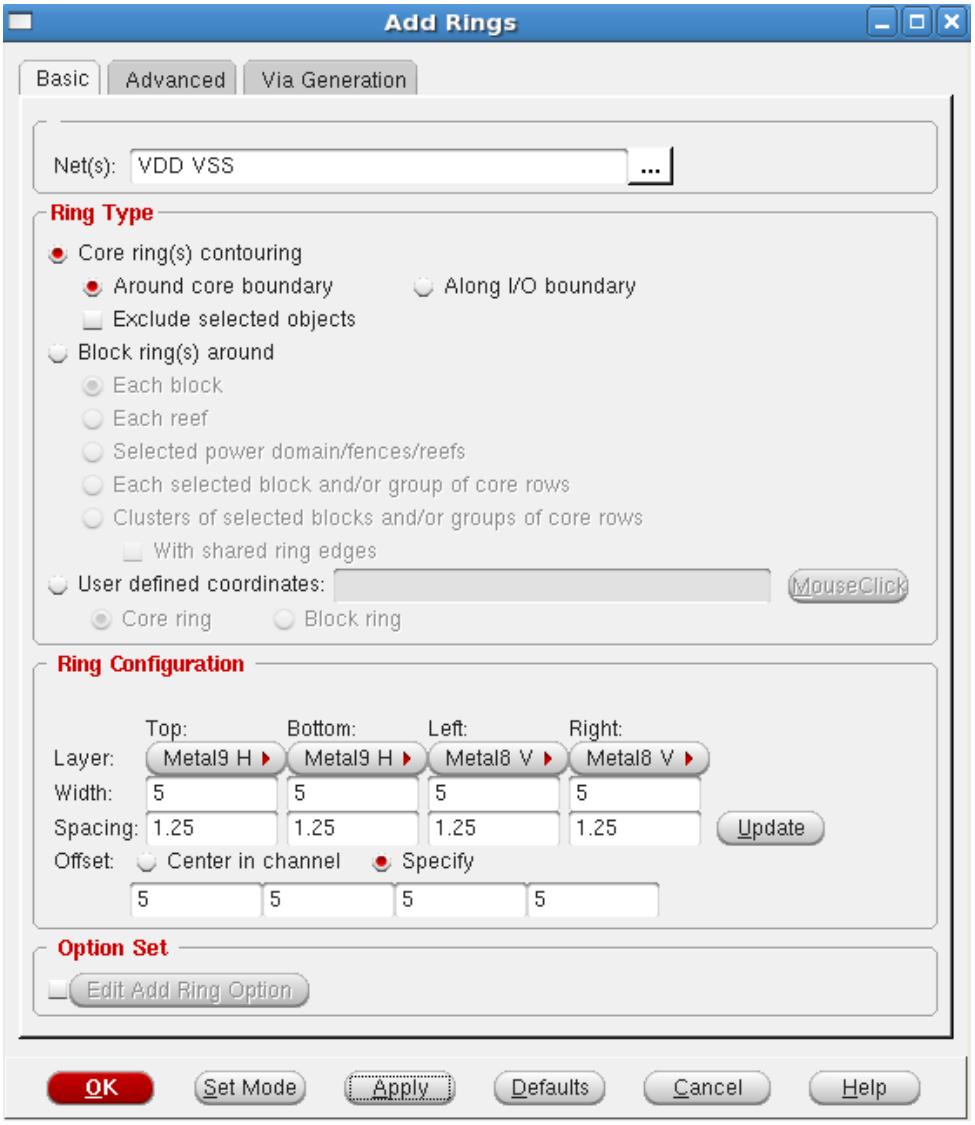
2. Select **Power - Power Planning - Add Ring**

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

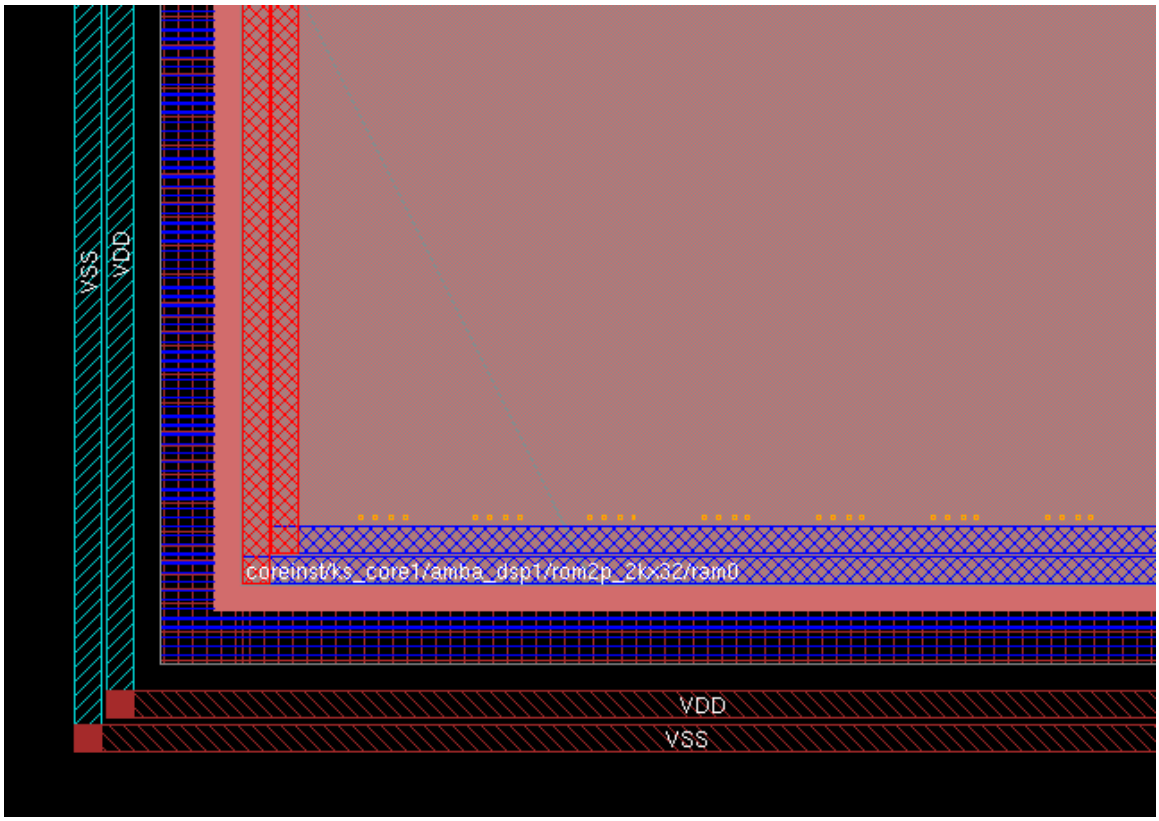
   Net(s): ***VDD VSS***

   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 **5**.
   - 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:** *Specify* and enter a value of 5 for each side.




Click **Apply** and you should see two power rings going around the core on Metal8 and Metal9 with a width of 5um.




Add Power Rings around selected blocks by doing the following:

4. Select the block `coreinst/ks_core1/periph1_PH` in the upper left corner.

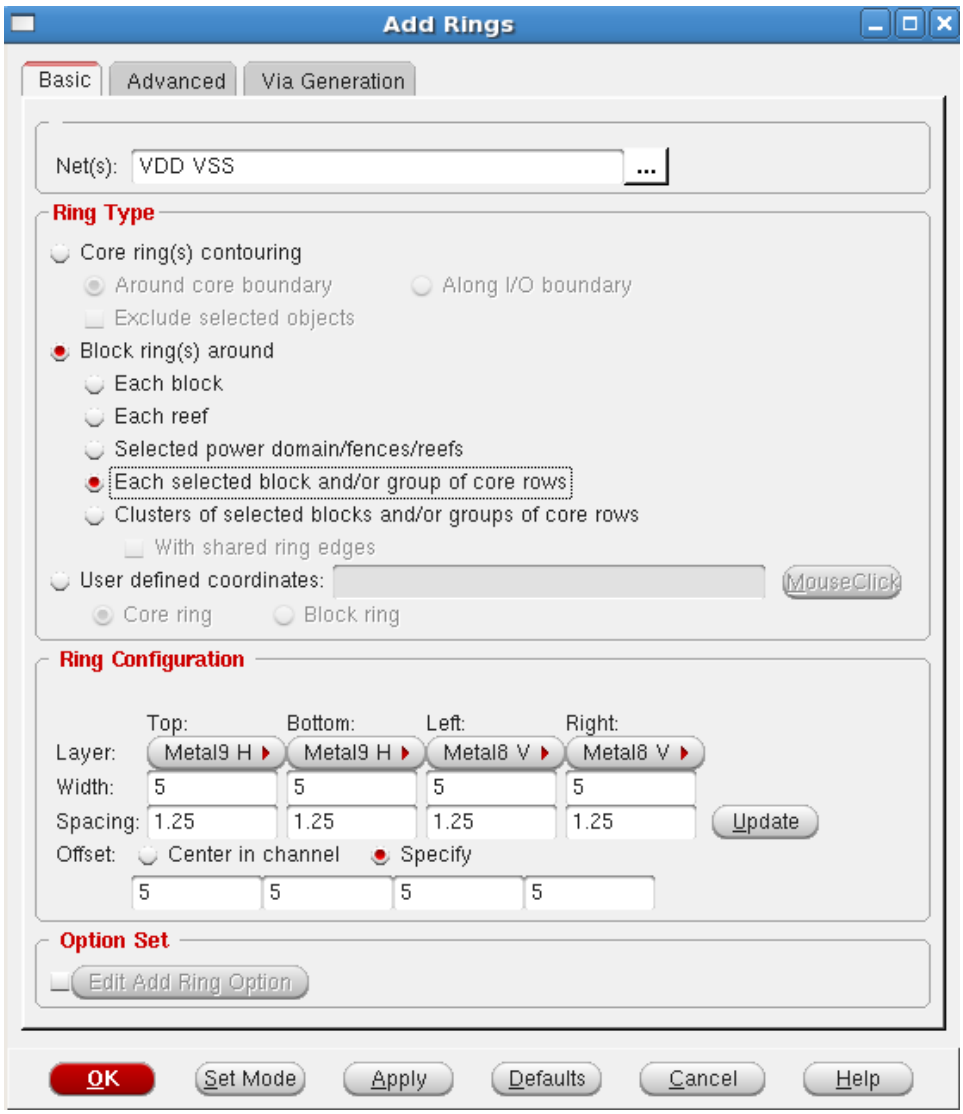
5. Add power rings around this instance by completing the Add Rings form as follows and click Apply.

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

   In the Ring Type section:
   - Select – **Block ring(s) around**
   - Select – **Each selected block and/or group of core rows**

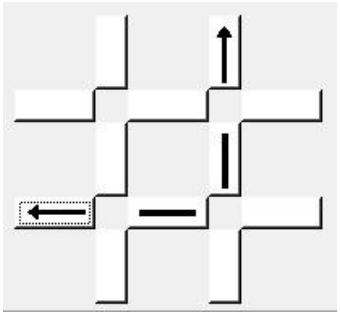
   In the *Ring Configuration* section:
   - Set **Top** and **Bottom** layers to **Metal9 H**
   - Set **Left** and **Right** layers to **Metal8 V**
   - Set the **Width** for each side to **5**.
   - Set **Spacing** to **1.25** for each side or click the **Update** button to automatically set the minimum 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:** *Specify* and enter a value of 5 for each side.

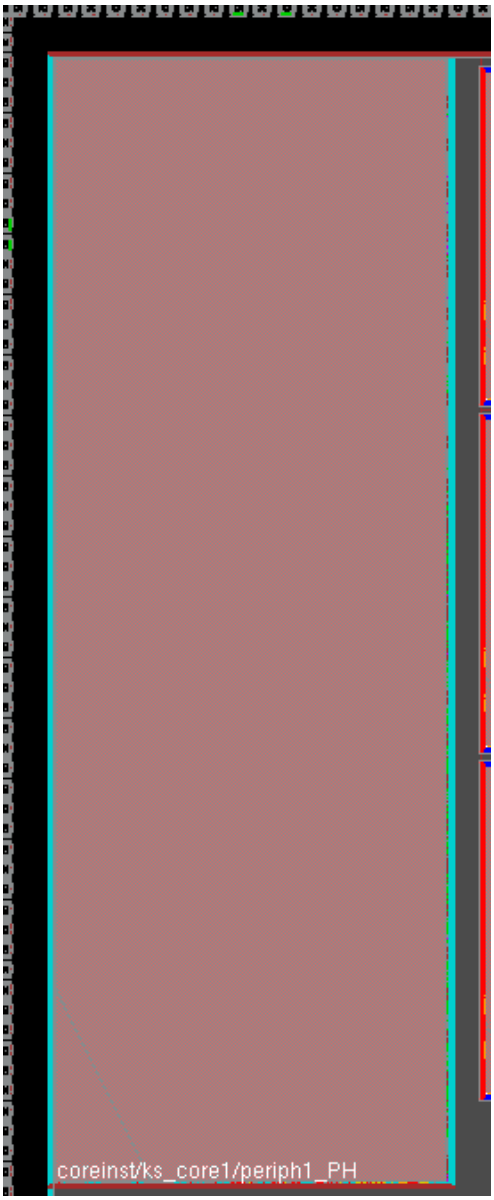



Select the **Advanced** tab on the Add Rings form

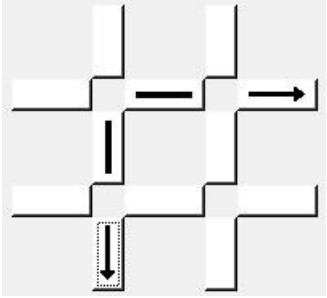
- To extend the ring around this block to the core ring set the Customer Ring Sides and Extensions as shown below:



Click **Apply** and observe the ring around this block (you may have to zoom in to the edge).




6. Select the block `coreinst/ks_core1/leon1` in the bottom right corner.
    - Keep all the settings the same on the Add Rings form except change the ring extensions on the Advanced tab as shown below:



Click **OK** and observe the ring around this block.


## 3-3 Creating Power Stripes

Power stripes are added using the **Add Stripe** command.

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

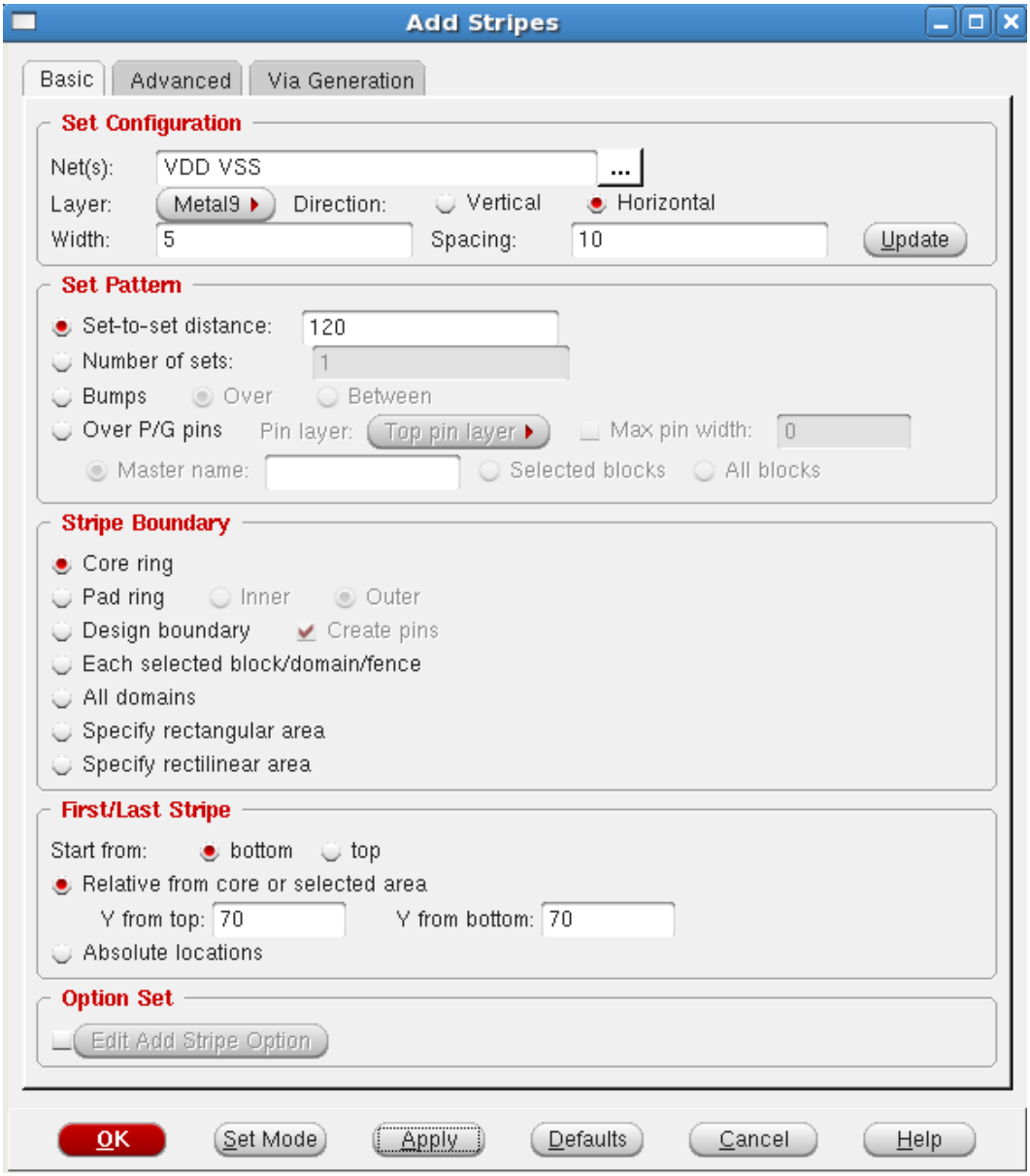
2. Complete the **Basic** tab as shown below. Do not click **OK** yet.

   Net(s): ***VDD VSS***
   Layer: **Metal9**
   Direction: **Horizontal**
   Width: **5**
   Spacing: **10**

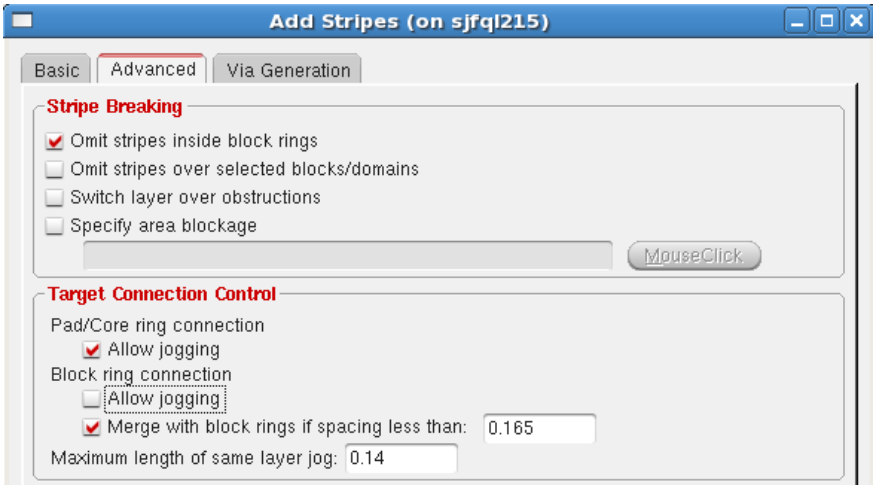
   In the Set Pattern section:
   a. Set to set distance: **120**

   In the Stripe Boundary section:
   b. Select – **Core ring**

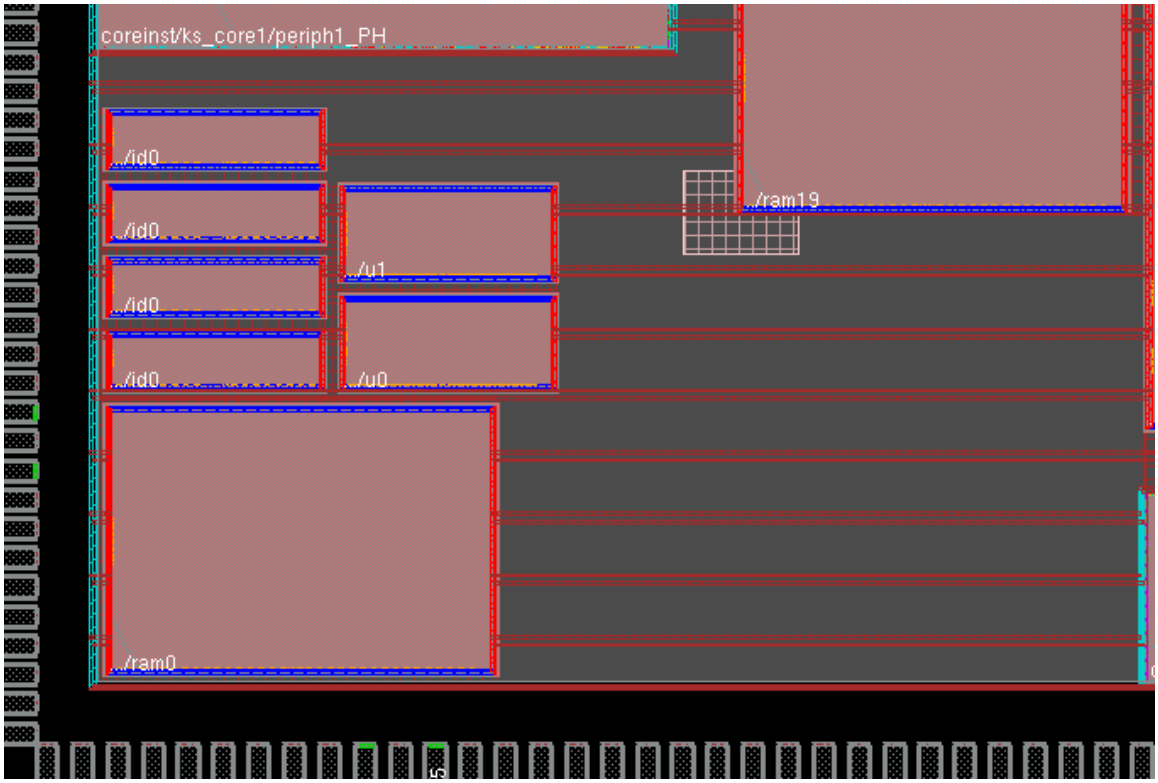
   In the First/Last Stripe section:
   c. Select – **Start from**: **bottom**
   d. Select – **Relative from core or selected area**
   e. Specify – **Y from bottom offset: 70 Y from top offset: 70**




3. On the **Advanced** tab:
    - Select **Omit stripes inside block rings**. This will break stripes so they do not go over the blocks.
    - Under **Block Ring Connection** deselect **Allow jogging**.




4. Click **Apply**. Now you should see the horizontal power and ground stripes created for the design as shown below:




**Tip:** If you are generating stripes for a large design you can utilize multi-threading. Run `setMultiCpuUsage` to specify the number of CPUs to use then run `addStripe`.

Add Metal8 Power Stripes as follows:

5. Select **Power - Power Planning - Add Stripe** if the Add Stripe form is not open.

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

Net(s): ***VDD VSS***
Layer: **Metal8**
Direction: **Vertical**
Width: **5**
Spacing: **10**

In the Set Pattern section:

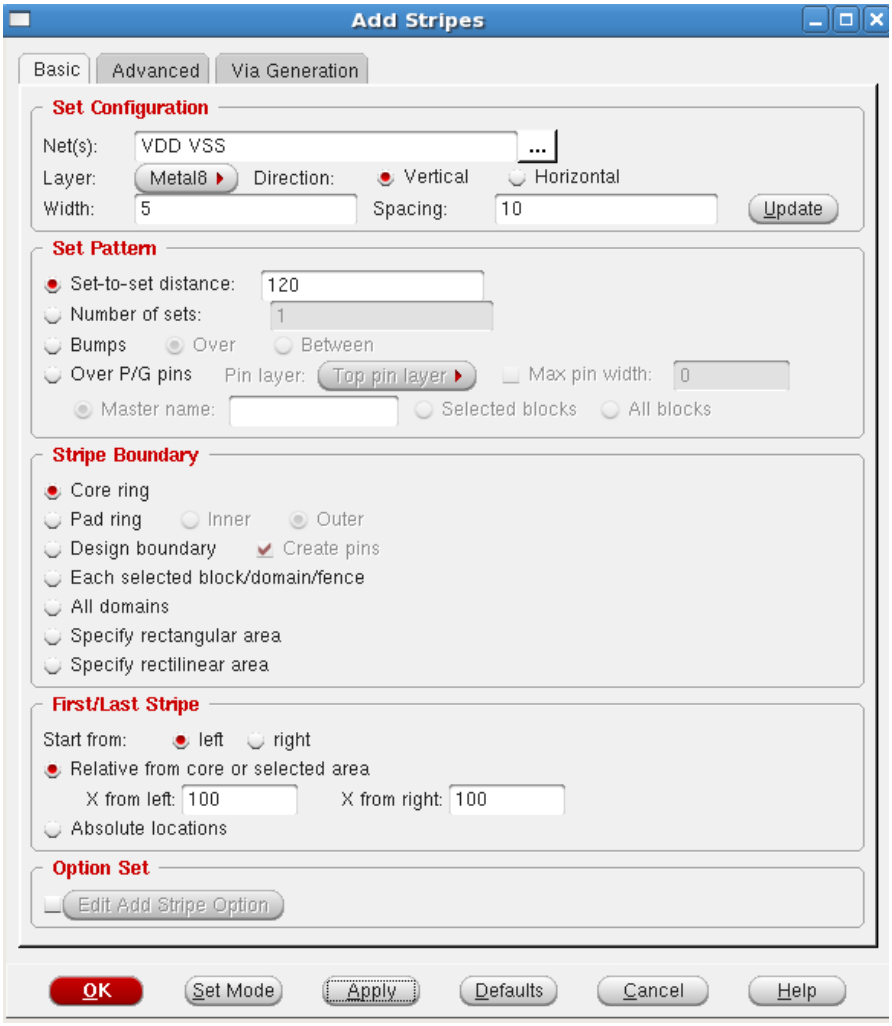
f. Set to set distance: **120**

In the Stripe Boundary section:

g. Select – **Core ring**

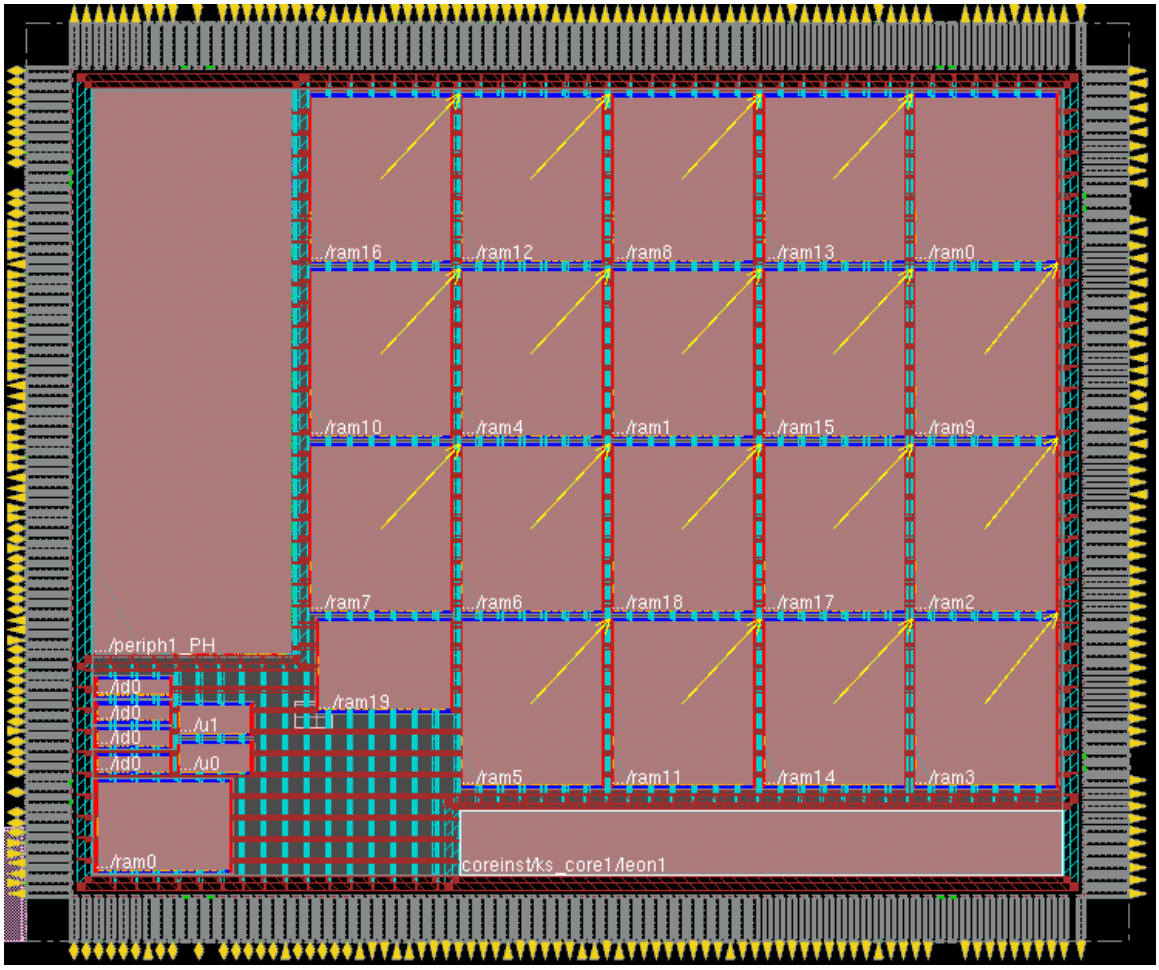
In the First/Last Stripe section:

h. Select – **Start from**: **left**
i. Select – **Relative from core or selected area**
j. Specify – **X from left offset: 100 X from top offset: 100**



7. On the **Advanced** tab:
    - Select **Omit stripes inside block rings**. This will break stripes so they do not go over the blocks.
    - Under **Block Ring Connection** deselect **Allow jogging**.

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




## 3-4 Routing the Power/Ground Structures

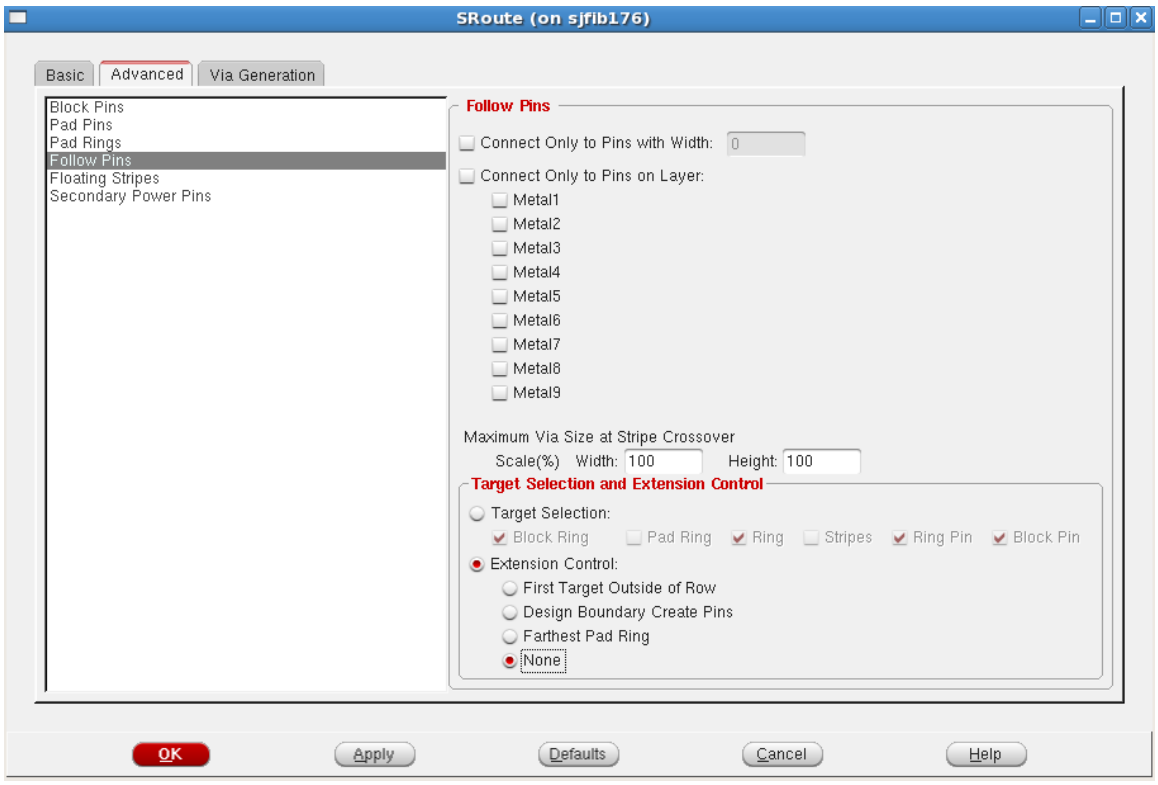
To route the remaining power and ground structure, the SRoute form is used. The SRoute program routes the block pins, pad pins, pad rings, and standard cell pins. For our exercise, power and ground needs to be routed to the standard cell pins, blocks and pad pins. Make the following entries:

1. Select **Route - Special Route**

2. Complete the Basic tab as follows:

    a. In the **Net(s)** field specify: *VDD VSS*
    b. Deselect **Pad Rings** and **Floating Stripes**. Keep everything else as the default.

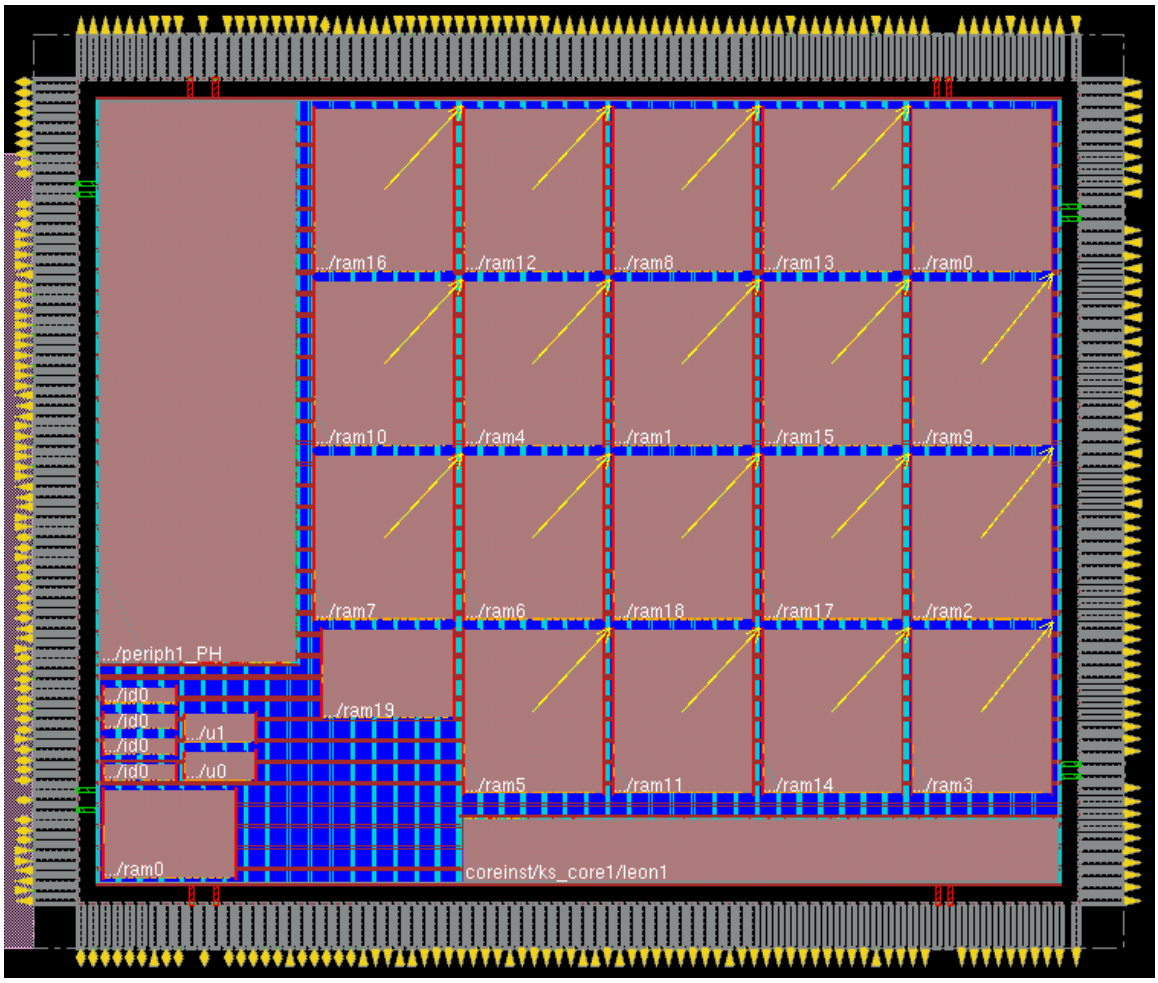

3. Select the **Advanced** tab and the select **Follow Pins** in the left column to specify mode settings for follow pin routing. Follow pin routing is the routing connecting the standard cell power pins together. Under **Extension Control** select **None** as shown below.




4. Click **OK**.

This creates the standard cell power rails on Metal1 horizontally across the core. Your floorplan should now look like the following. If it does not load in the existing floorplan using the command below.

```
loadFPlan DATA/power.fp
```




5. Save the design by typing the following at the encounter prompt:

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

This concludes the floorplanning lab. Once a floorplan is completed you then apply the place and route implementation flow.

End of Lab