FLOOR PLANNING

Floor planning is a mapping between the logical description (the netlist) and the physical description (the floorplan).

Goals of **Floor planning** for a **Design**:

• Arrange the blocks/Macros on a chip.

• Decide the location of the I/O pads.

• Decide the location and number of the power pads.

• Decide the type of power distribution.

• Decide the location and type of clock distribution.

The above goals have to be achieved to

• Minimize the chip area

• Minimize delay

• Minimize routing congestion

Floor planning Inputs	Outputs
➢ Design netlist (required) ➢ Area requirements (required) ➢ Power requirements (required) ➢ Timing constraints (required) ➢ Physical partitioning information (required) ➢ Die size vs. performance vs. schedule trade-off (required) ➢ I/O placement (optional) ➢ Macro placement information (optional) Files: *.v/*.vhdl; *.sdc; *.lib; *.lef; *mmmc.view ; *.def; *.scandef; *.cpf	➢ Die/block area ➢ I/Os placed ➢ Macros placed ➢ Power grid designed ➢ Power pre-routing ➢ Standard cell placement areas ➢ Floorplan DEF

CPF file – Common Power Format (*.cpf)

Some of the important definitions:

Track: Track is a virtual guideline/path for the tool at which the signal routing happens in an SOC design. Tracks are defined for each metal layer in both preferred and non-preferred directions, which are used by the router. The router routes the signal assuming the track to be at the center of metal piece.

Row: This is the area defined for standard-cell placement in the design. A row height is based on the height of the standard cells used in design. There can be rows of various sites/heights in the design based on the type of standard cells used.

Placement constraint Types:

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.

Utilizations:

Target Utilization (TU): 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.

Effective Utilization (EU): 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.

Types of Placement Blockages:

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.

This tutorial helps for performing floor planning (only in elaborative steps) using Cadence Innovus Tools.

Contents

Basic Floor Planning

1. **Importing the Design**

File → Import Design → Click Load → Browse to *. globals file (to map all the initialization variables / files like netlist, LEF, LIB) → Click OK

Sample *.global Format :

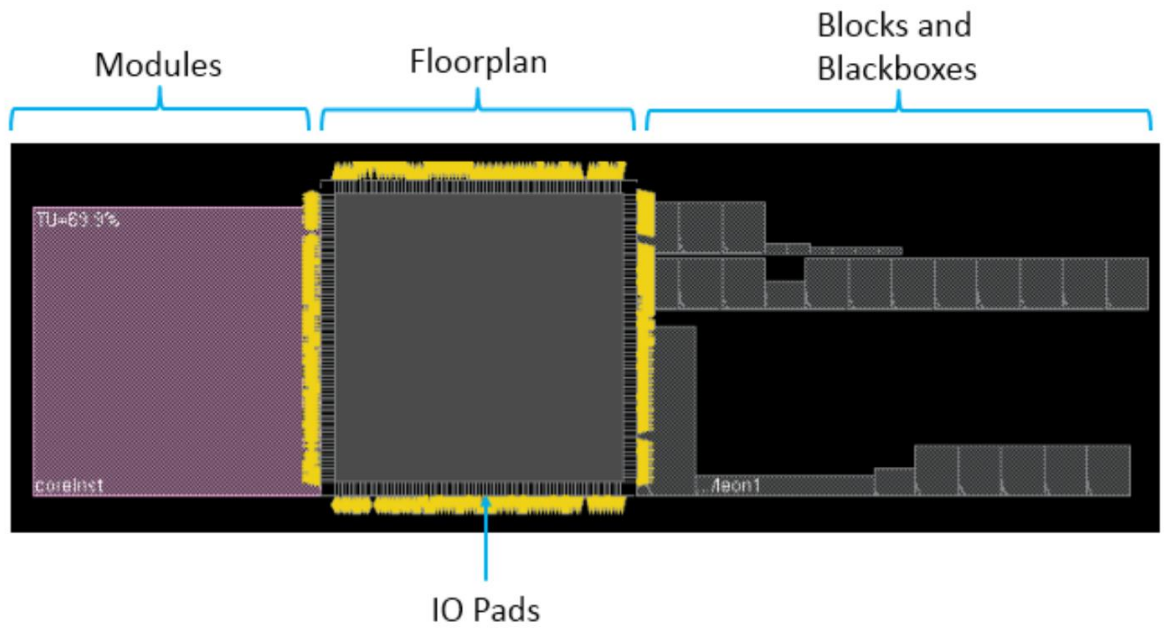
```
set init_assign_buffer {1}

set init_design_settop 0

set init_pwr_net {VDD}

set init_gnd_net {VSS}

set init_import_mode { -keepEmptyModule 1 -treatUndefinedCellAsBbox 0 -useLefDef56 1}

set init_io_file {../../*.io}

set init_lef_file {../../*tech.lef ../../*tech.lef }

set init_mmmc_file {../viewDefinition.tcl}

set init_top_cell {top_module_name}

set init_verilog {netlist.v}
```

Press **F** → Fit Design

Press **Shift+Z** → Zoom Out

Press **A** → Select Mode

After Importing you may see an image of a basic floorplan like shown below (www.cadence.com)



Left of the floorplan - pink modules corresponding to the modules in the Verilog netlist. Right of the floorplan - Hard macros (blocks) and any blackboxes that have been defined in the netlist.

2. Specifying the Floorplan

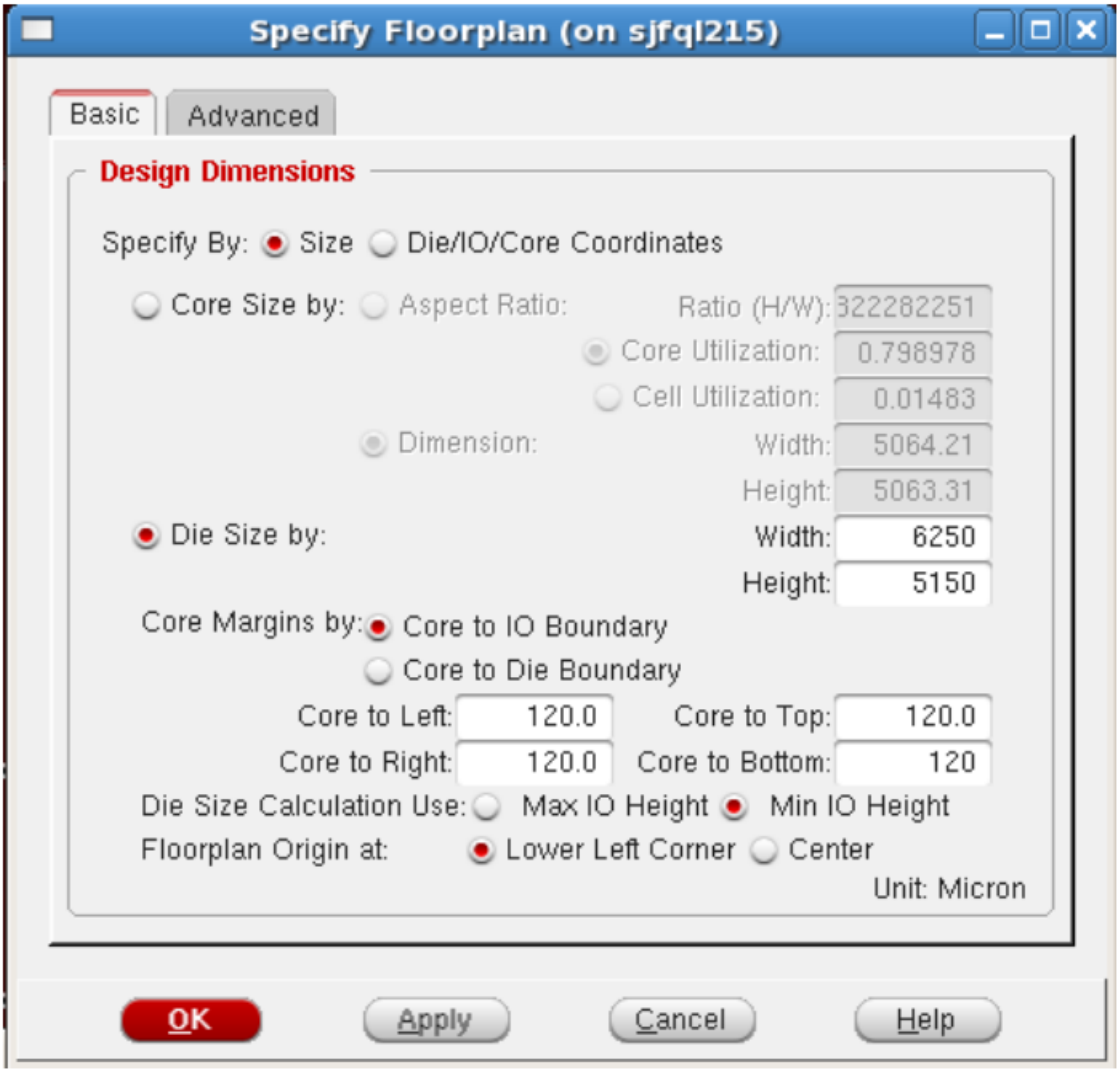
Specify Floorplan form allows you to derive the floorplan size based on the target utilization.

Floorplan → Specify Floorplan

Choose the following according to your target utilization requirement

- **Width & Height** in **Die Size By**
- **Core to Left, Core to right, Core to Top and Core to Bottom in Core Margins by: Core to IO Boundary**

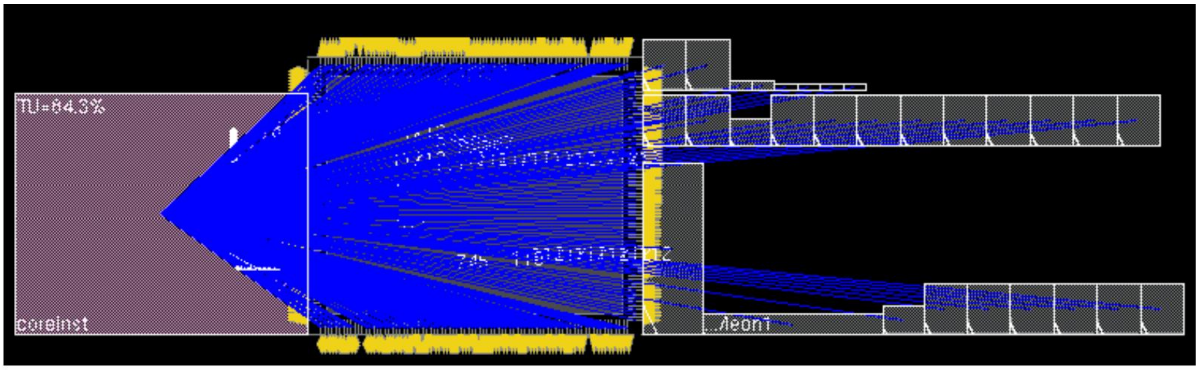
Click OK



3. Moving and Editing Placement Constraints

Floorplan Tool Widgets

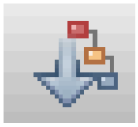
Module size is based on the target utilization of the modules using the area of hard macros and standard cells that are part of the module.



Blue colour connection -- **flight lines**

White colour -- **highlighted blocks**

The blue flight lines display the number of connections between the selected module and other instances such as other modules and blocks.

Press **Shift+G (or)**  **→ Hierarchy Down (Ungrouping)**

Press **G → Hierarchy Up (Grouping)**

Press Ctrl+D → Deselect all

Press  in Tool widget to **Move/Resize/Reshape** of objects

Click by selecting the object/module/instance/block you want to move and then click the move in widget and click in the destination location where you want to plane the instance/ block

Reshaping the Module [By keeping area constant]

View → Set Preference → open the Preferences form

Edit tab → Select Maintain area option → Box Stretch Restrictions → OK

Resizing the Module

Move the cursor over one of its edges or corners.

The cursor will change to a double arrow when positioned over the edge to move.

Move the mouse to the desired location and Click again to complete the resize.

In order to know the details of the block/modules/insrtances, Press **Q** to open the **Attribute Editor** that shows the properties.

Command line function to specify the user grid

```
setPreference ConstraintUserYGrid 0.0005
setPreference ConstraintUserXGrid 0.0005
```

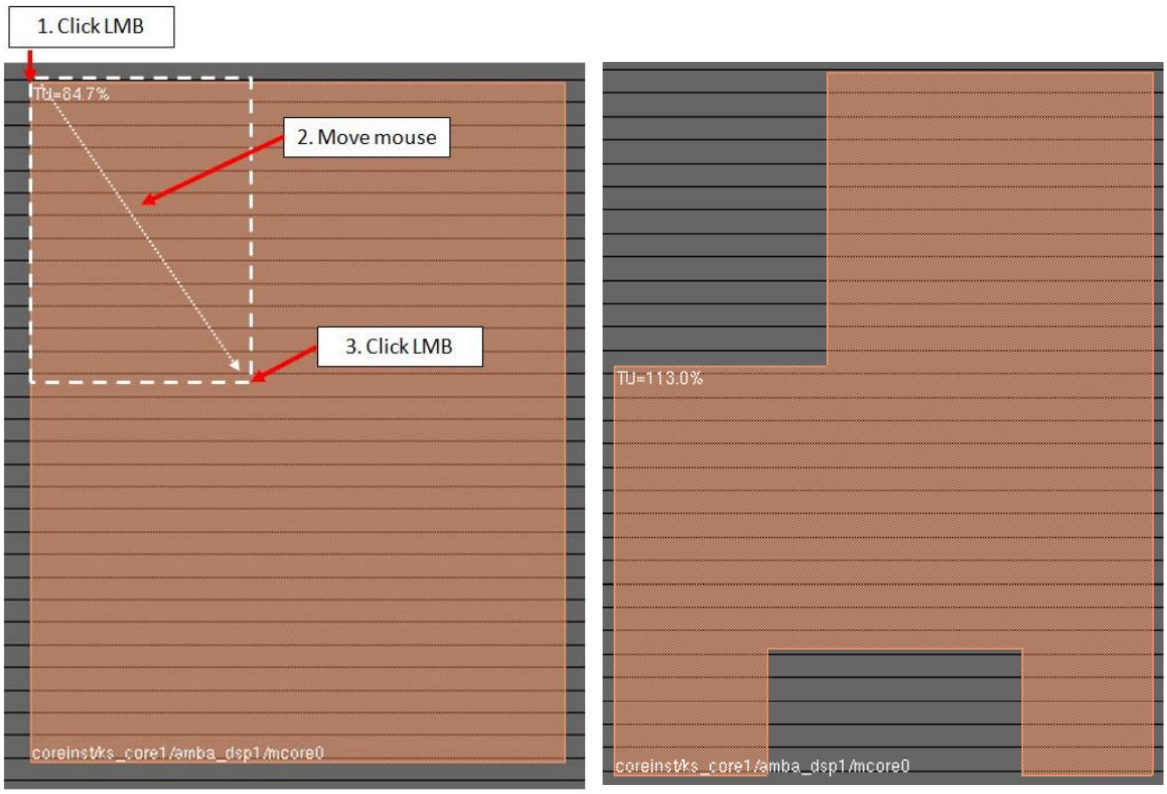
Changing the type of instance/block from **Guide** to **Fence** is done by opening the attribute editor of the block and change the type.

After changing to fence, the colour of the module changes from pink to brown.

Changing the Shape of the module:

Use the **Cut Rectilinear widget**  from tool widget.

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



To clear floorplan objects;

Floorplan → Clear Floorplan → All Floorplan Objects → OK

4. Placing Hard Macros

Relative Floorplanning and manual movements can be used for creating floorplan of a small number of blocks.

Relative Floor planning

Relative Floorplanning 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 suitable corner of the core (let it be upper right).

To place a macro we have to edit the constraints

Select the Macro you want to place in the core

Floorplan → Relative Floorplan → Edit Constraints

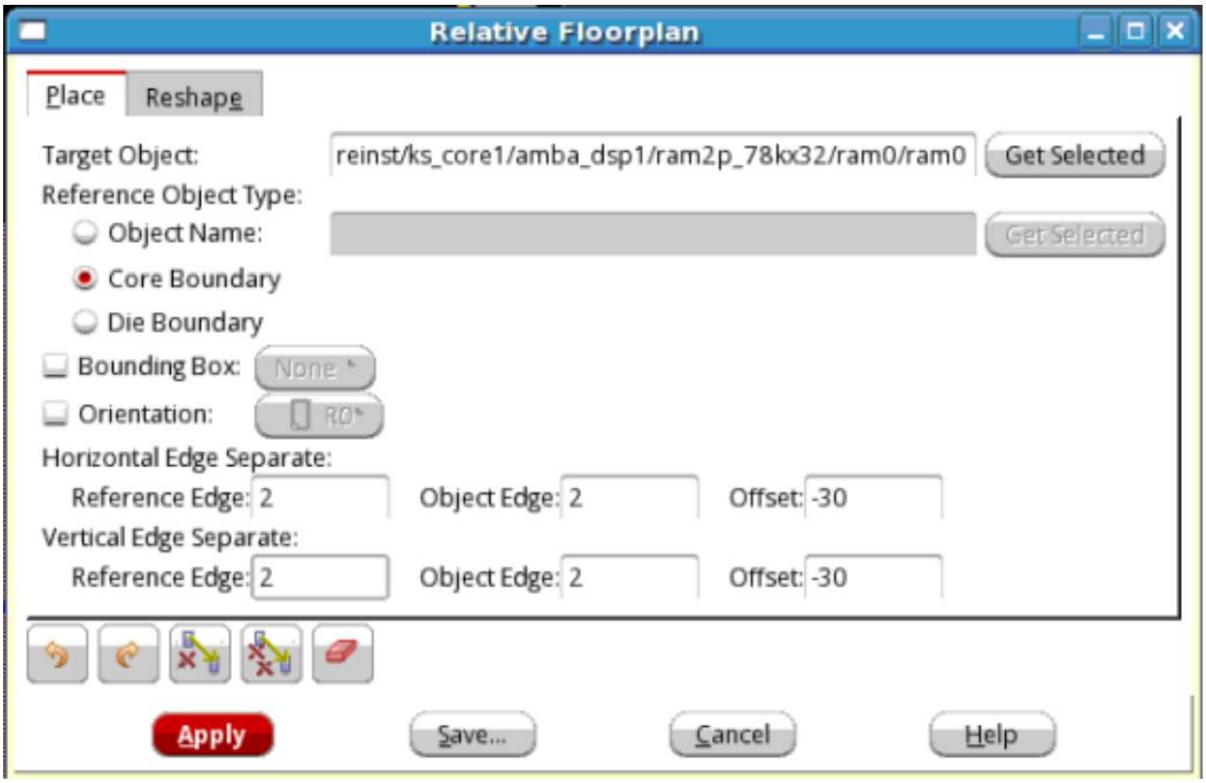
Choose the **Target Object** by clicking **Get Selected** and **select the Macro**

Choose the following values based on the location you want place the Macro

Reference Object Type → Core Boundary

Under **Horizontal Edge Separate** & **Vertical Edge Separate →** Choose values for **Reference Edge, Object Edge, Offset**

Click **Apply**



The macro is placed at 30um below the top core boundary and -30um from the right core boundary in the core area.

Yellow lines (arrow) between the Macro block placed and the boundary of the core indicates there is a relative constraint between the block and top core boundary

To change the status of the instance to soft fixed;

Floorplan → Edit Floorplan → Set Instance Placement Status.

Select **Set to status:** as **Soft fixed**

Command to delete the relative floorplan is

```
delete_relative_floorplan -all
```

Similar to placing a single macro, groups of macros can be placed in an array by **defining array constraints**;

Select a Macro or all the group of Macros to be placed in core in GUI by selecting/clicking it or using the following command.

```
selectInst */coreinst/Macro1
```

```
selectInst */coreinst/Macro2
```

Floorplan → Relative Floorplan → Define Array Constraint form **→ click Load seleted → Apply**

Choose the following values how you want to place the group of Macros as an array (For E.g. 4x5).

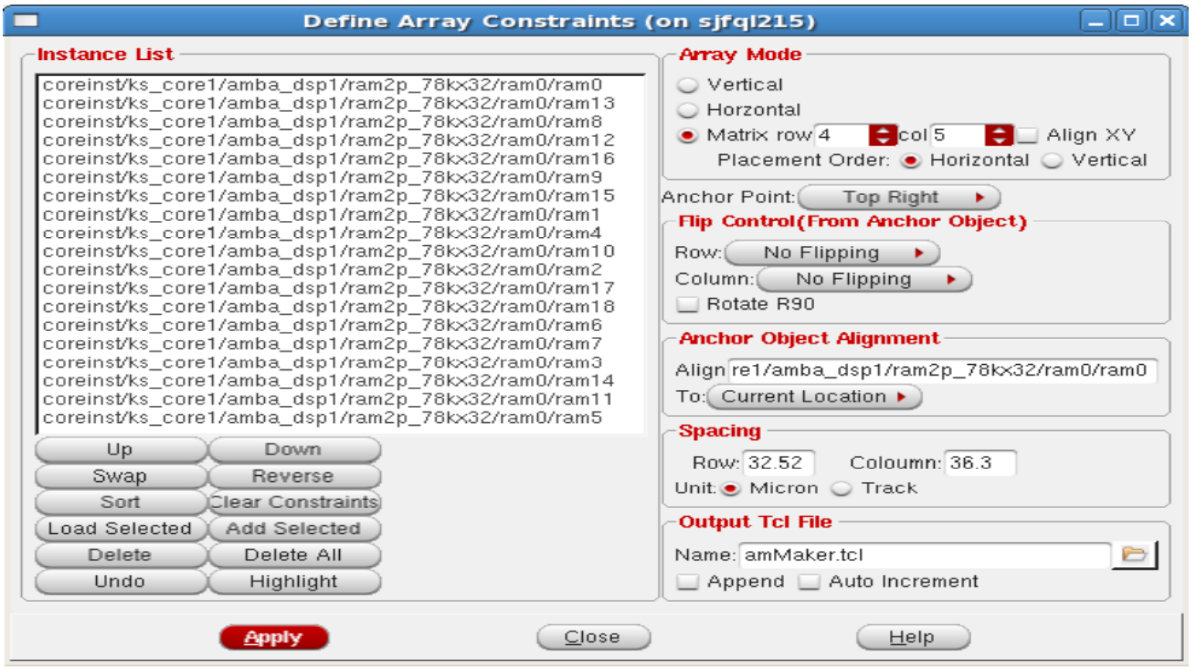
Matrix row __ col __

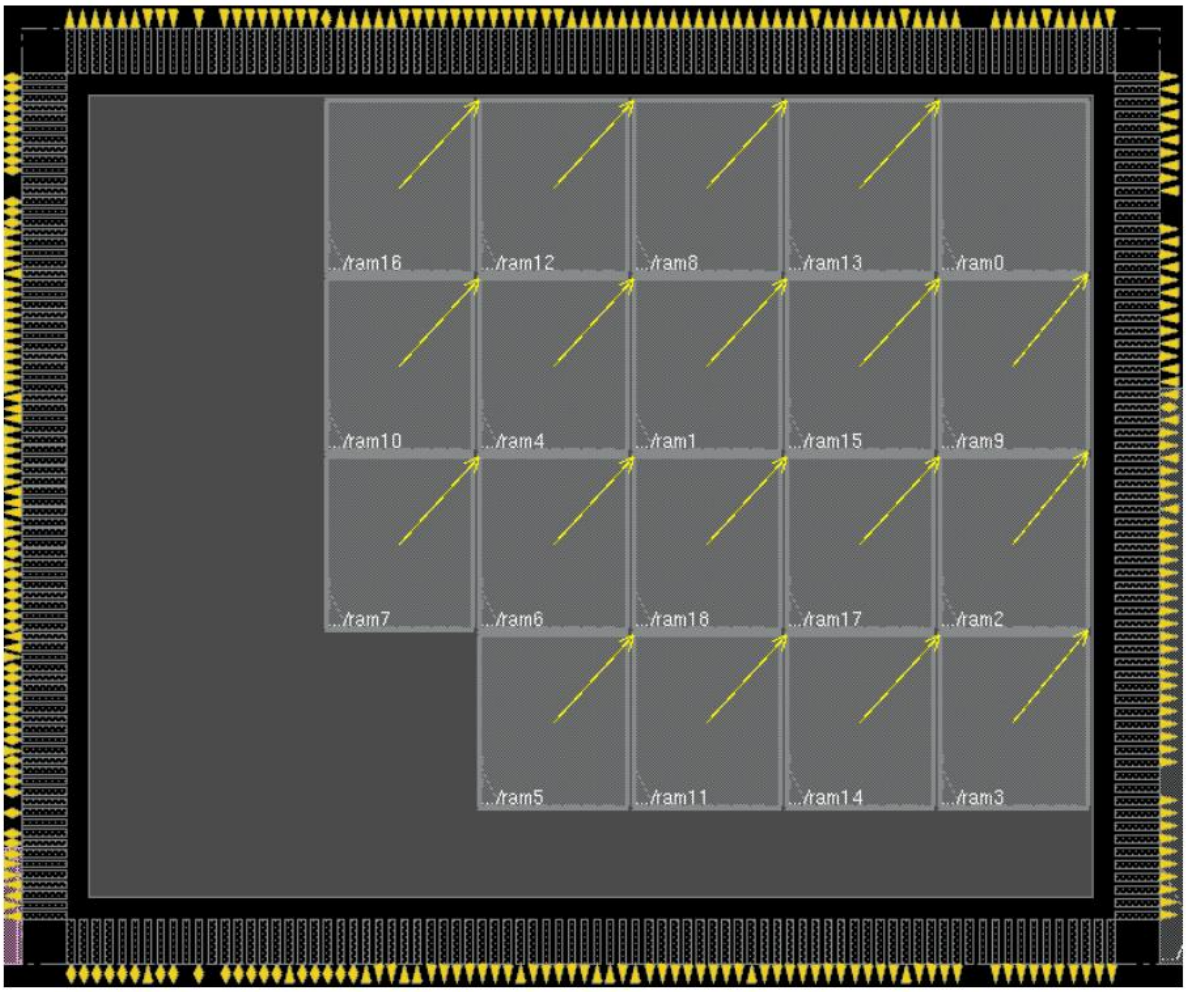
Anchor Point: Top Right

Align:_____

Spacing Row: ___ Column: ___ Unit: Micron

Select the spacing values that avoids DRC spacing violations.



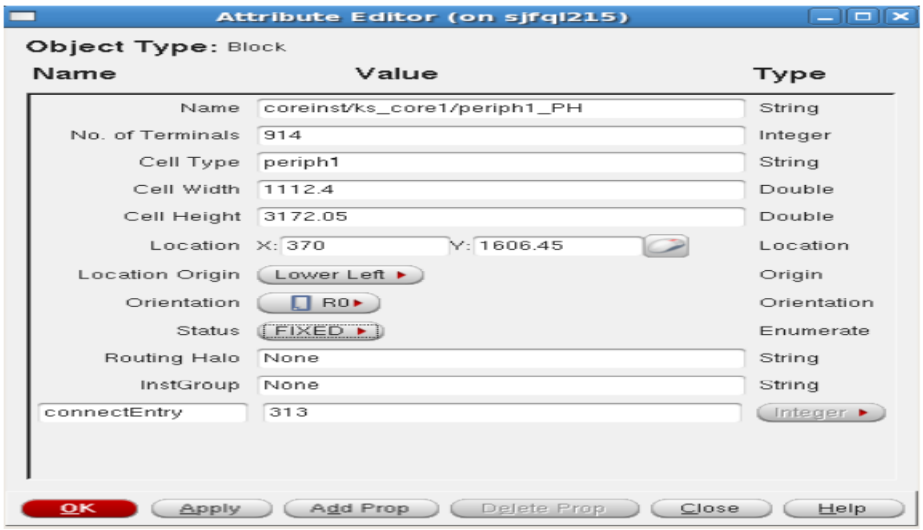


Placing the Macro at a suitable location inside the core, select the attribute Editor of the Macro and then choose the following according to designer

Location X: 370.0 Y: 1606.45

Orientation: R0

Status: FIXED



Dr.Udaya Shankar S, https://www.linkedin.com/in/dr-udaya-shankar-s-m-e-ph-d-86438968/

In Command;

```
placeInstance coreinst {2518.8 370.12} R0 -fixed
```

Press **Ctrl+R → Redraw Floorplan**

The same method can be used for placing the Macros inside the core at the suitable location without creating any Spacing Violations.

Tracing Macro – Finding Connections between Macro to Other Macros/Ports/Blocks

Floorplan → Trace Macro to open Trace Macro GUI

Select **Macro Names → Get Selected → Choose the Macro for tracing connections**

Choose **Level __; Trace Mode → Netlist Based → Click Run**

Shows trace how the other Macros are connected to this macro

Red Lines → Port/Macro Connections

Blue Lines → Instance/Module Connections

To save the generated trace to a report by

Click **Report button (or) in command**

```
report_obj_connectivity -insts *macro -file traceMacro.rpt
```

Command to trace the connection of any selected/specified Macros or ports

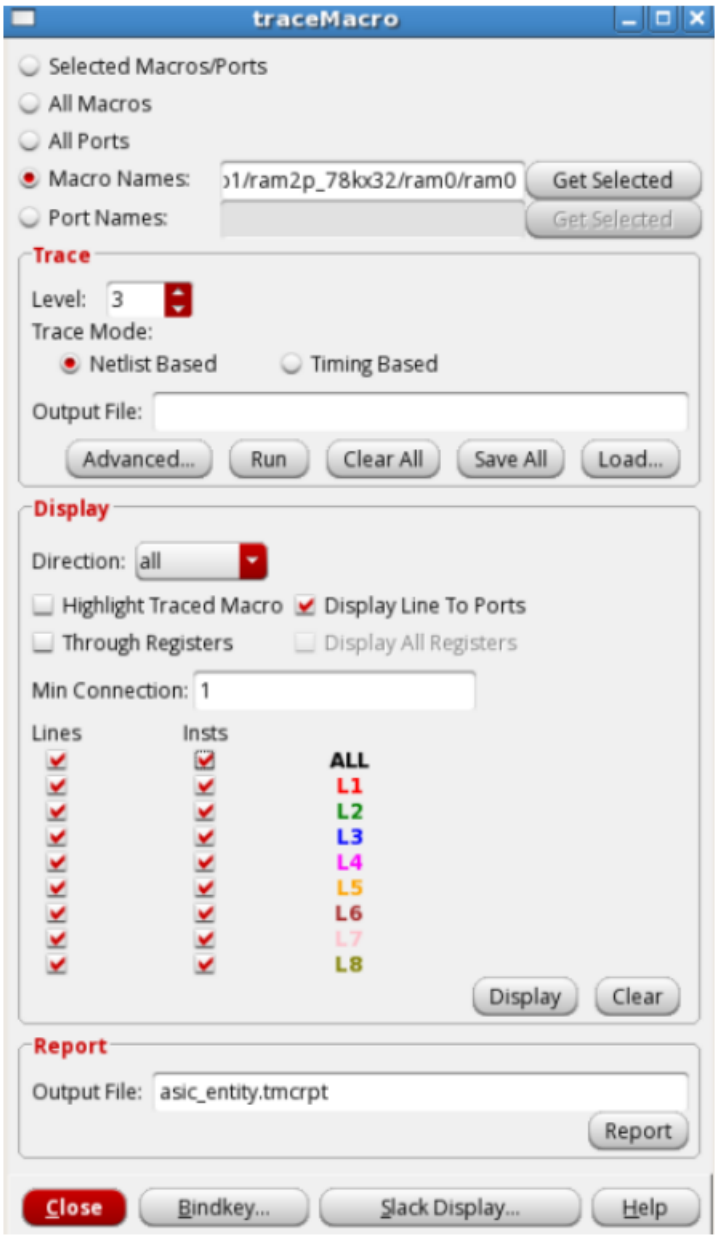
```
trace_obj_connectivity -insts */macroname -level 3 -mode netlist_based
```

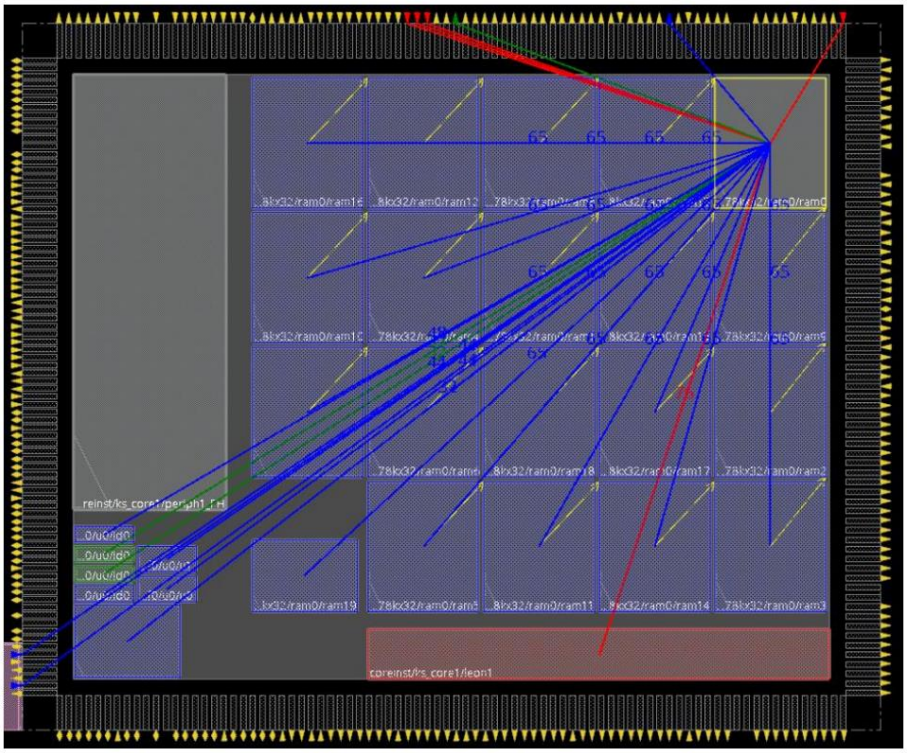
Command to trace connectivity of the specified block

```
display_obj_connectivity -insts *macro/block -level 3 -direction all -line_to_ports -min_connection 1
```

Command to clear the trace display in the GUI,

```
display_obj_connectivity -reset
```

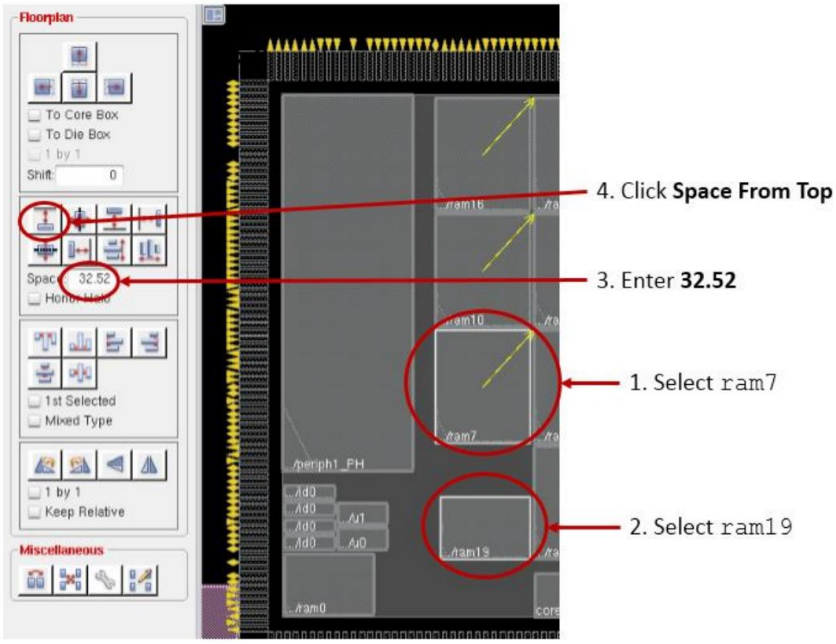

traceMacro
Selected Macros/Ports
All Macros
All Ports
Macro Names:
o1/ram2p_78kx32/ram0/ram0
Get Selected
Port Names:
Get Selected
Trace
Level: 3
Trace Mode:
Netlist Based
Timing Based
Output File:
Advanced...
Run
Clear All
Save All
Load...
Display
Direction: all
Highlight Traced Macro
Display Line To Ports
Through Registers
Display All Registers
Min Connection: 1
Lines
Insts
ALL
L1
L2
L3
L4
L5
L6
L7
L8
Display
Clear
Report
Output File: asic_entity.tmcrpt
Report
Close
Bindkey...
Slack Display...
Help



5. Using Floorplan Toolbox

Floorplan → Floorplan Toolbox

Floorplan Tool box helps to adjust the spacing between macros, adjust location of macros inside the core.



Core Utilization can be checked using the command

`checkFPlan -reportUtil`

The Narrow channel locations inside the core can be reported out using the command

```
report_narrow_channel -width 10.0 -ignore_placement_blockage
```

The command to generate a script to generate the existing floorplan

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

6. Resizing the Floorplan

Expanding/shrinking the floorplan in either direction by adjusting the block placement.

Measure the channel distance you want to expand or shrink using Ruler.

Press **k → Ruler/Scale**

After measuring, click

Floorplan → Resize Floorplan (for shrinking Vertical Channel)

In Resize Floorplan form, choose the options/values according to your requirement;

Mode: *shift based*

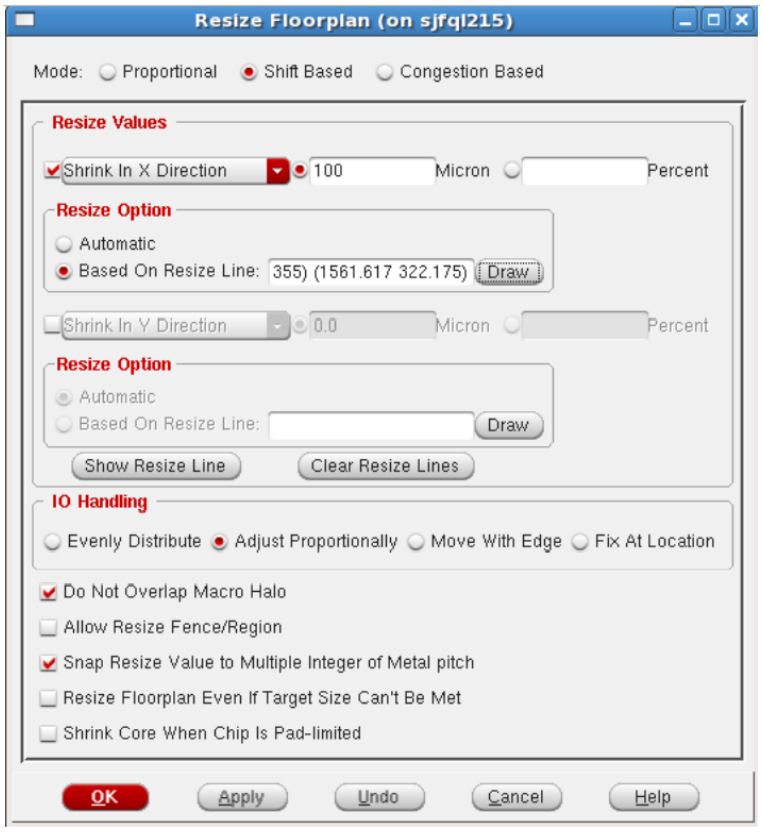
Shrink in X Direction:____um

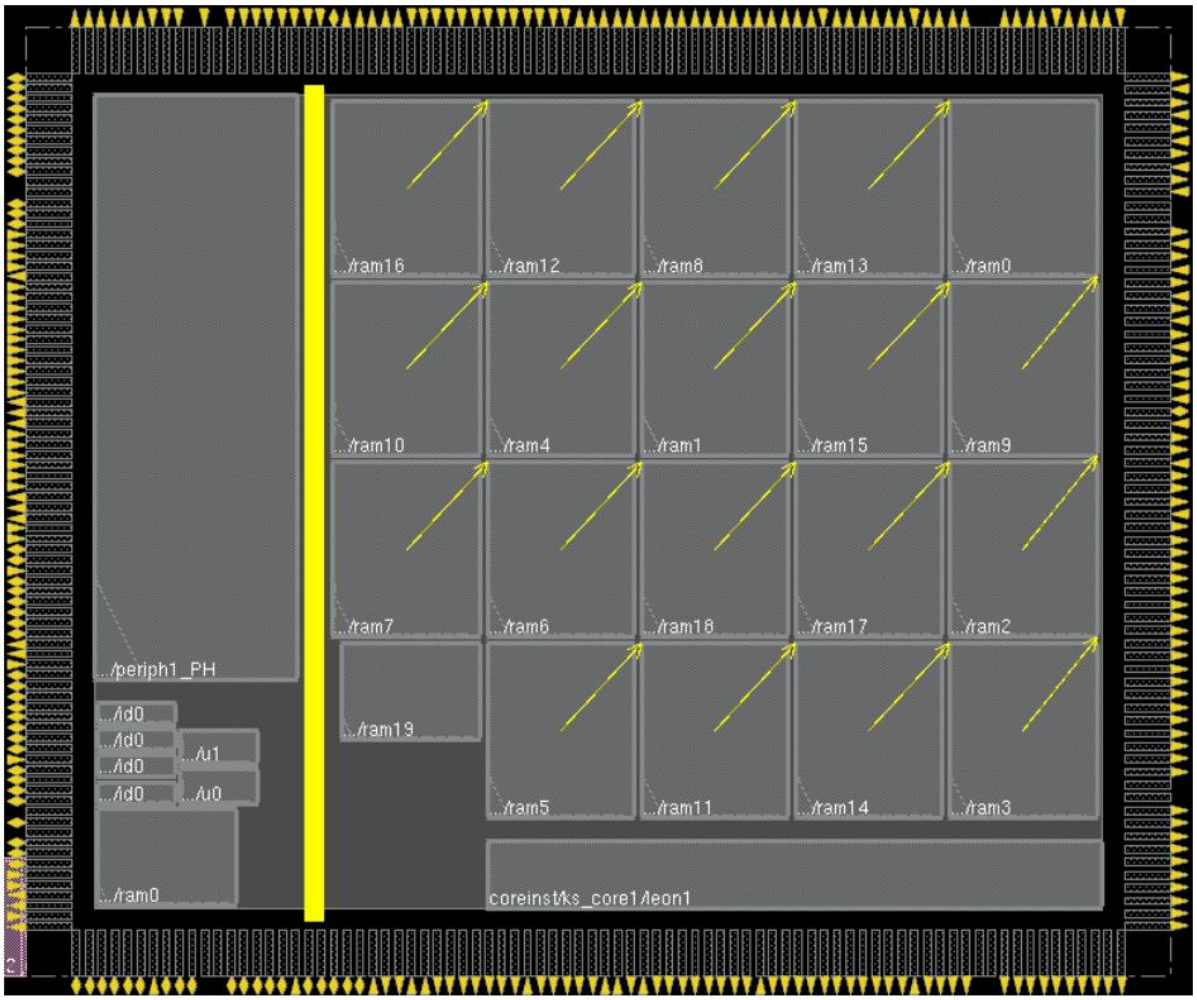
Select **Based on Resize Line → Click Draw → In Innovus Window**

Go to select mode by pressing A, then draw a line by double clicking at the starting pint and clicking at the end point, then press ESC.

Click Ok in **Resize Floorplan form**

Now that the Channel area where you draw the line has been shrinked (or expanded), if you want measure using ruler and check it.


Resize Floorplan (on sjfql215)
Mode: Proportional Shift Based Congestion Based
Resize Values
Shrink In X Direction
100
Micron
Percent
Resize Option
Automatic
Based On Resize Line:
355) (1561.617 322.175)
Draw
Shrink In Y Direction
0.0
Micron
Percent
Resize Option
Automatic
Based On Resize Line:
Draw
Show Resize Line
Clear Resize Lines
IO Handling
Evenly Distribute
Adjust Proportionally
Move With Edge
Fix At Location
Do Not Overlap Macro Halo
Allow Resize Fence/Region
Snap Resize Value to Multiple Integer of Metal pitch
Resize Floorplan Even If Target Size Can't Be Met
Shrink Core When Chip Is Pad-limited
OK
Apply
Undo
Cancel
Help



7. Creating Placement & routing Blockages

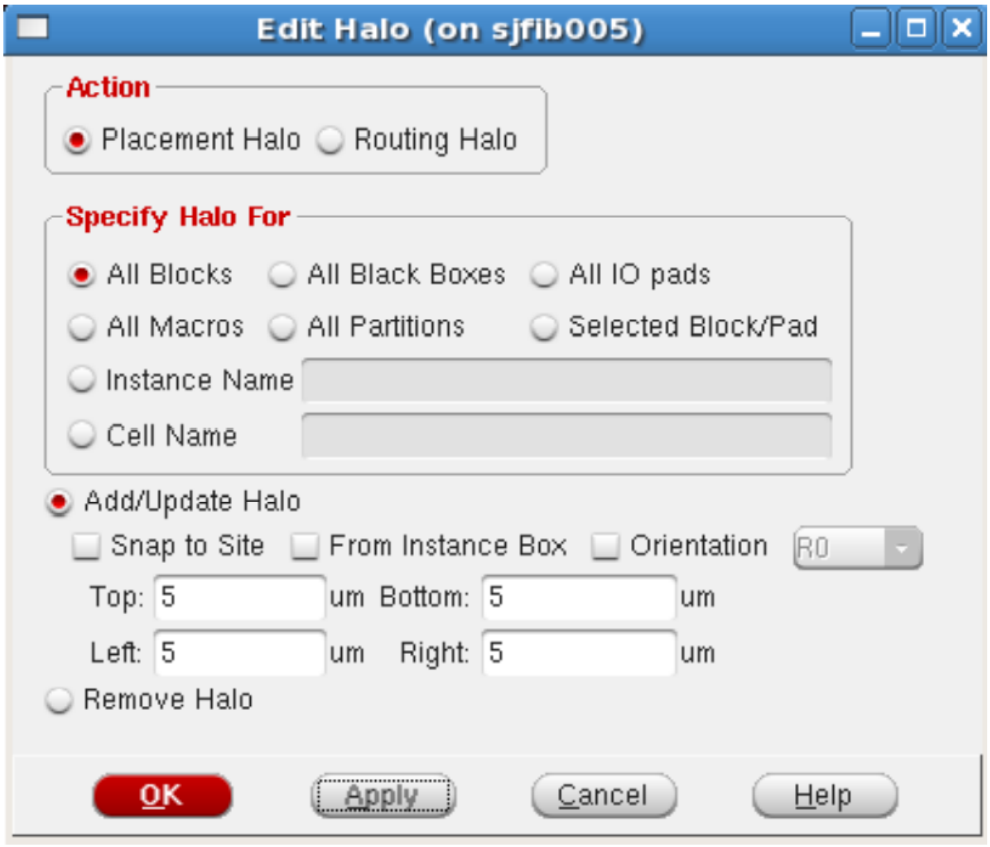
Placement blockages: control placement of cells in specified areas.

Placement halos are placement blockages around blocks that prevent cells from being placed inside the halo area.

Floorplan → Edit Floorplan → Edit Halo

In **Edit Halo form** → Choose Placement Halo → Specify halo for → Choose All Blocks → In Add/Update Halo → chose the width of halo to be created (E.g., 5um) in **Top**, **Bottom**, **Right** and **Left** → click OK

This adds a placement blockage for all blocks around them in all sides.

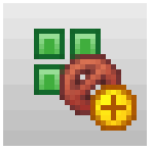


Note:

- A halo is associated with a block, so if a block is moved the halo also moves with it.
- A placement blockage prevents cell placement in a specific area but unlike a halo it is not associated with any block.
- The halo (size) may vary according to designer based on technology to avoid DRC Violations

Creating Soft Placement Blockage:

You can create a soft placement blockage to an individual module manually;

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

Draw a placement blockage over the channel above the corresponding block.

Change to selection mode by pressing A.

Double-click on the placement blockage you drew to open the Attribute Editor. Change the type of blockage created from hard to soft.

The shape turns to a red mesh indicating it is a soft placement blockage.

Note:

A partial placement blockage can reduce routing congestion by setting a maximum placement density in a specified area.

Creating Partial Placement Blockage

Press **Shift+Y** (or) **Create placement blockage using Tool Widget button**

Draw the blockage where you need the placement blockage around the block.

Press A to go to select mode

Double click the blockage created to open attribute editor and then change the **Type** to **Partial** and **change Partial Percentage** to **30% (as per requirement)**. [This means maximum allowed cell density in this blockage area is 30%]

Blockage turns to mesh indicate as partial placement blockage.

Creating Routing Blockages

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.

To create a routing blockage, click symbol  from Tool widget

Draw the area (rectangle) where you want to create a routing blockage in the core.

Double Click the blockage created to open the attribute editor, then specify the range of layers prevented from routing in the created blockage area.

Finishing the Floorplan

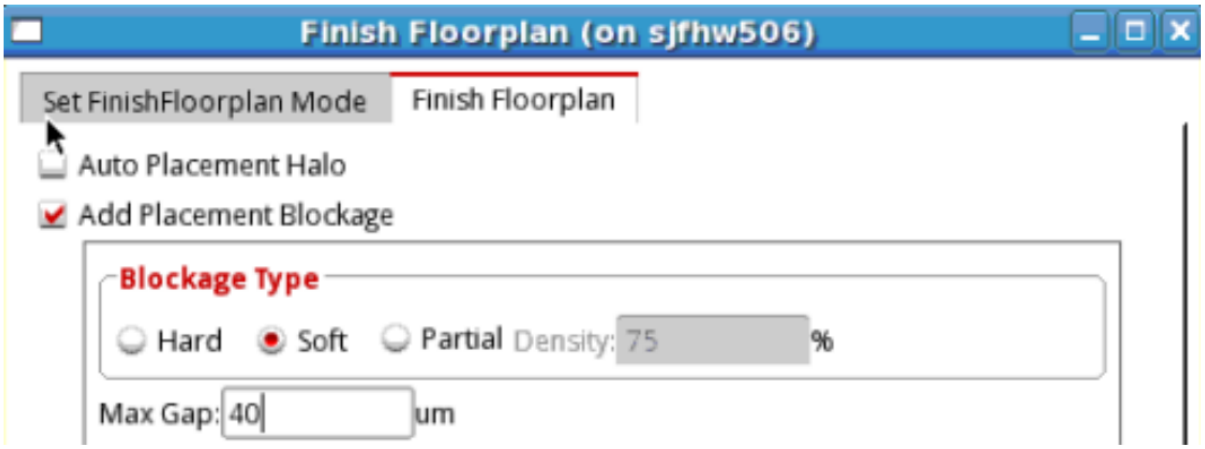
[For Automatic Insertion of Halo & Blockage to the Blocks/Macros]

The **"finishFlooplan"** command adds placement and routing blockages globally to the design instead of adding it individually to blocks/Macros.

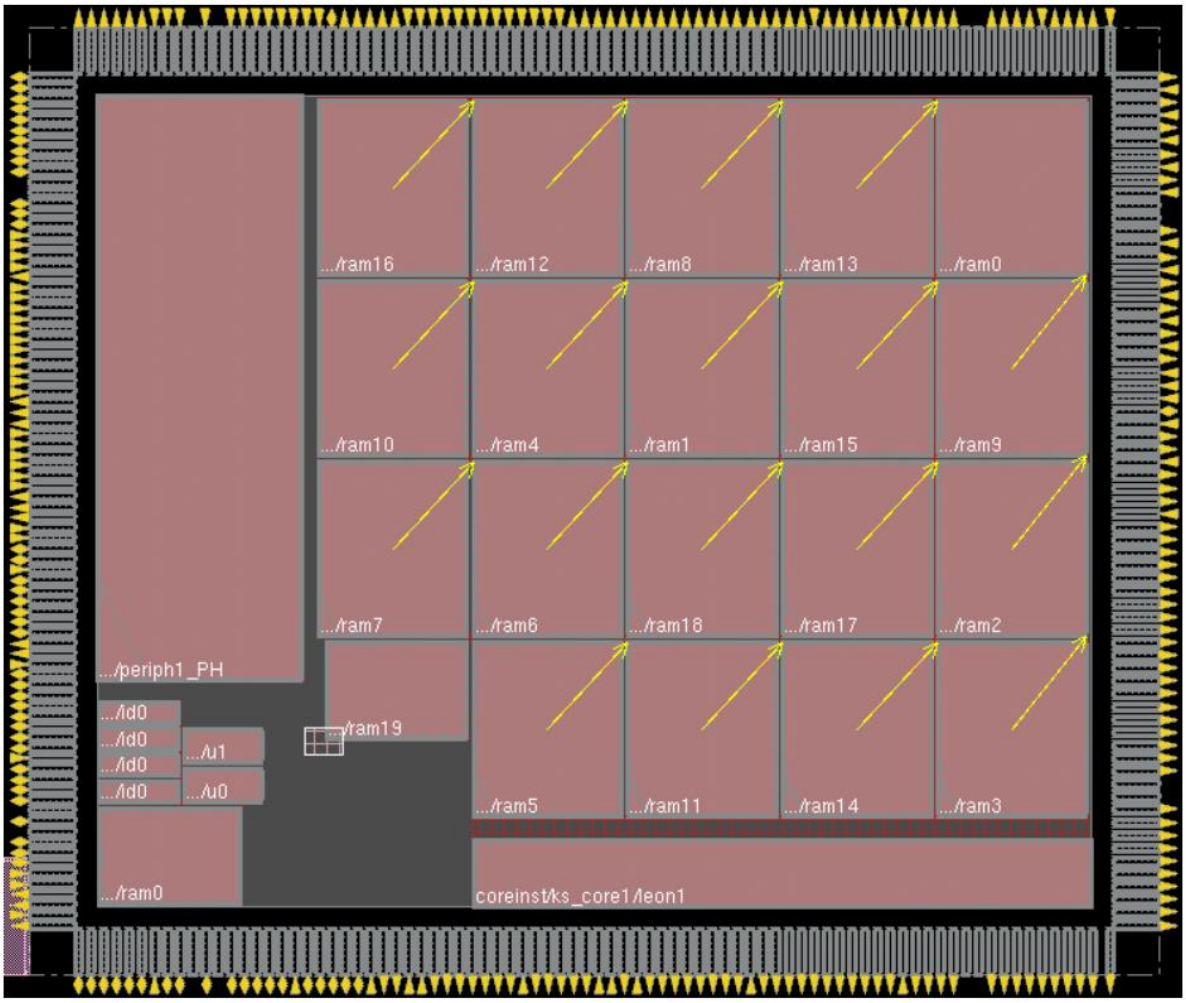
Floorplan → Automatic Floorplan → Finish Floorplan

In **setFinishFloorPlanMode** tab → Active Objects section → select Core, Macro and Soft Placement Blockage

In **FinishFloorplan** tab → Select Add Placement Blockage → Set Blockage Type to Soft → Set Max Gap to value (For E.g., 40um) to avoid DRC violation → OK



Now Blockage are automatically created around all blocks/Macros even in the small narrow channels.



8. Power Routing

Global nets for Power and ground must be assigned for entire design using command **“globalNetConnect”.**

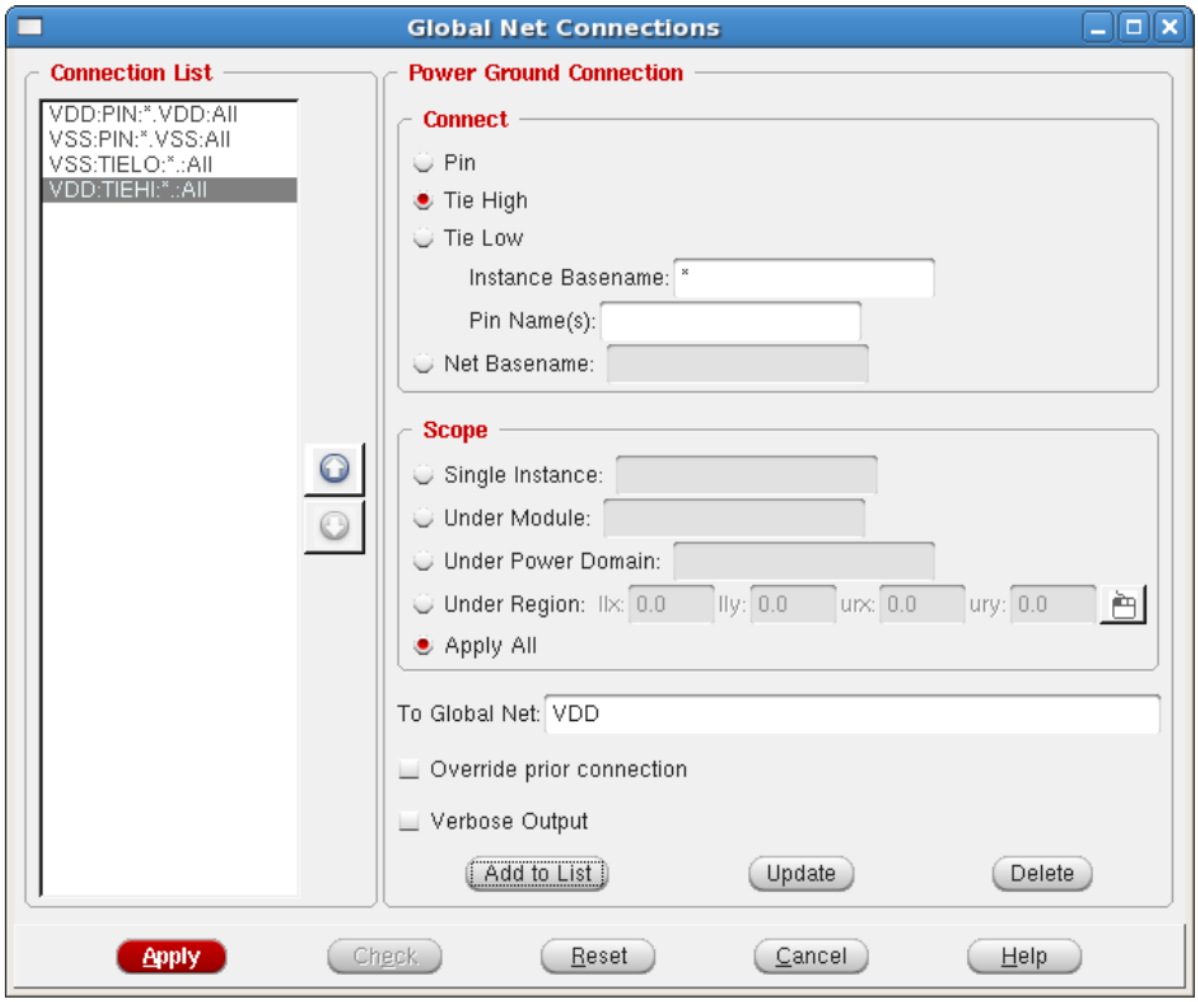
From netlist, power pins, tie high pins, tie low pins must be connected to Power (VDD) & ground (VSS) nets.

In command;

```
globalNetConnect VDD -type pgpin -pin VDD -all

globalNetConnect VSS -type pgpin -pin VSS -all

globalNetConnect VDD -type tiehi

globalNetConnect VSS -type tielo
```

In GUI as shown in below;

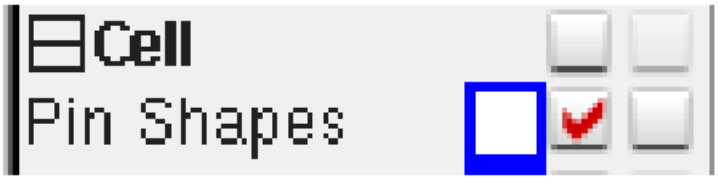
Power → connect Global Nets → Enter data → Click **Add to list**



Creating Power and Ground Rings

With the Power and Ground nets logically assigned, Power planning can be done. Power and ground rings are added around core area and also around specified blocks.

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.



To add the power rings around the core in GUI;

Click **Power → Power Planning → Add Ring**

Choose the following options as required;

Net(s): **VDD VSS**

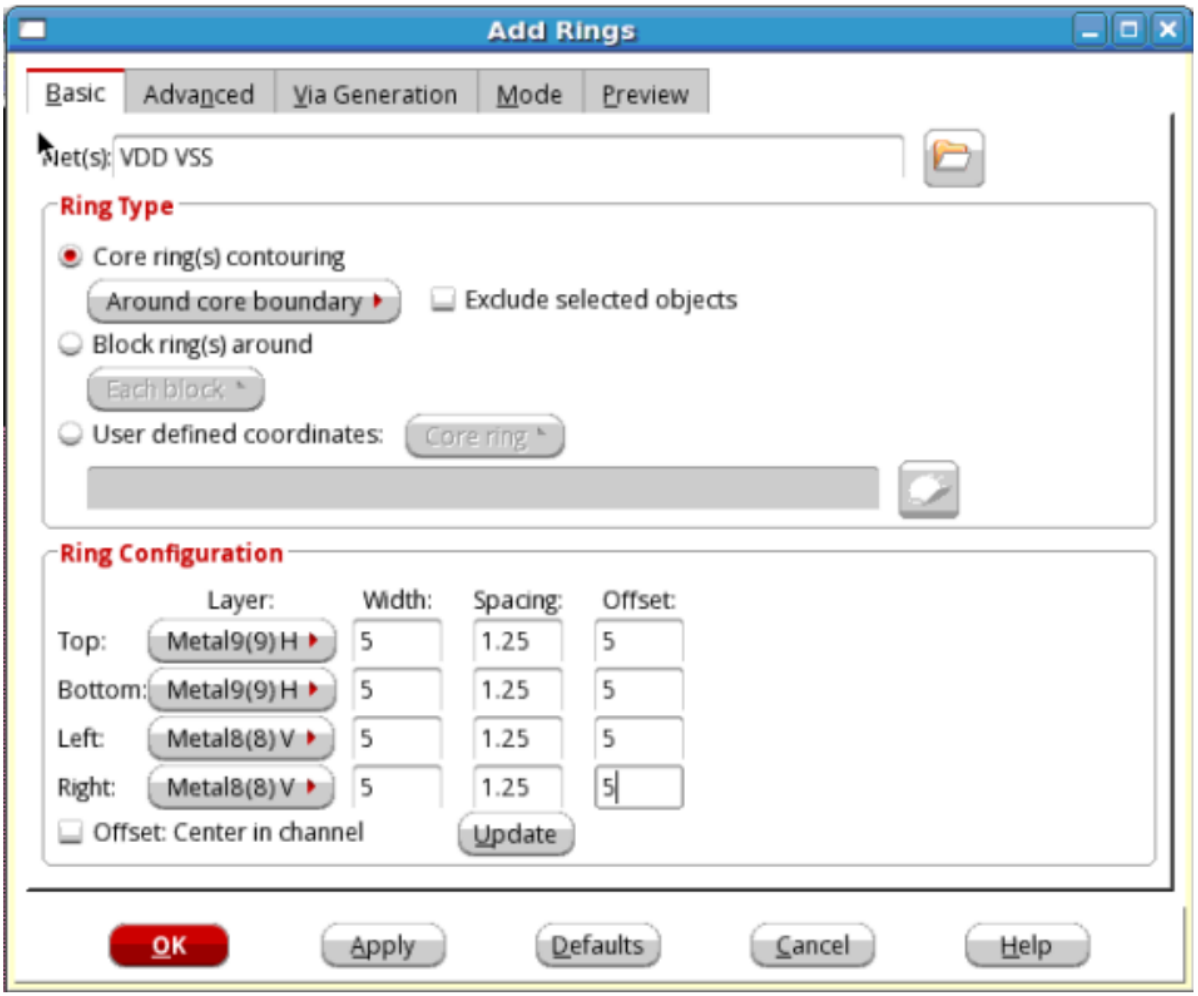
In the **Ring Type section**:

Select **Core ring(s) contouring** and **Around core boundary**

In the **Ring Configuration** section

Choose the **Metal layers** for **top, bottom, left and right**

Choose the values of **width**, **spacing** and **offset** depending on the requirement based on technology to avoid violations (spacing).



In Command to add rings

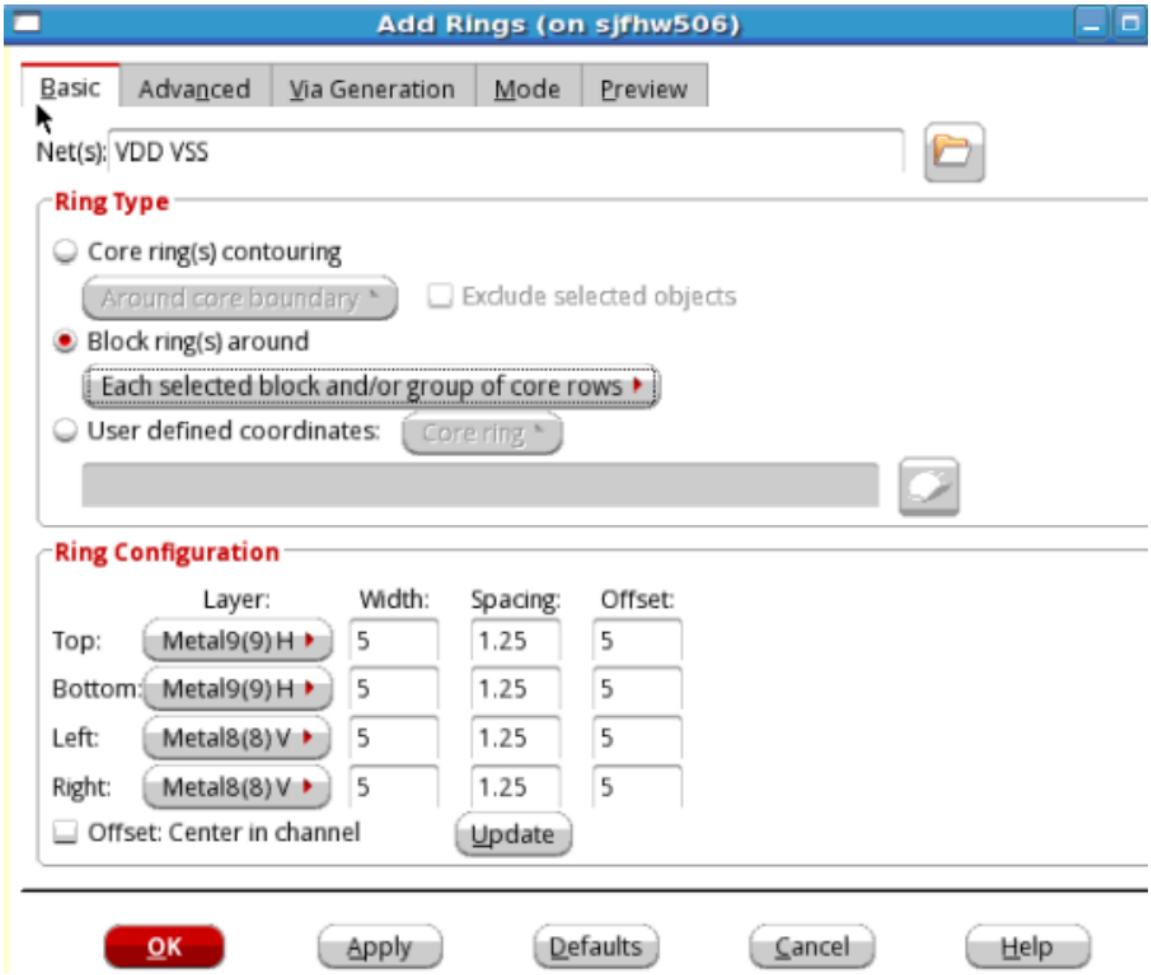
```
addRing -nets { VDD VSS } -type core_rings -around user_defined
-center 0 -spacing $pspace -width $pwidth -offset $poffset -
threshold auto -layer {bottom Metal1 top Metal1 right Metal2 left
Metal2 }
```

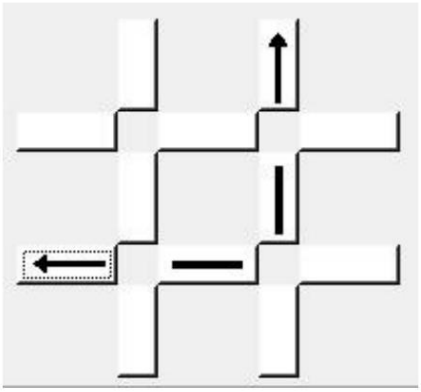
Adding Power Rings to a Particular Block

Select the block and repeat the process of adding rings with only change in following;

In **Ring type Sections** → Select **Block Rings around** and Select **Each Selected block or Group of Core Rows**

In **Advanced** Tab → **Add Rings** Form → **Set the Customer Ring Sides and Extensions** as per requirement for that block → Click **Apply**





Now that Rings are created around that block to supply VDD and VSS.

Creating Power Stripes:

Power and ground stripes are also added to create the power grid.

Additional connections from power rings to power/ground rails in the core.

Similar to Add Rings, the stripes are added using GUI or Command;

In GUI;

Select **Power → Power Planning → Add Stripe**

Don't Click OK

Choose the following options as required;

In **Basic** tab;

Net(s): **VDD VSS**

Layer: choose metal layer (For E.g., Metal9)

Direction: **Horizontal [for adding Horizontal Stripes Only]**

Width: Choose Value (For Eg., 5)

Spacing: Choose Value (For Eg., 10)

In the **Set Pattern section**:

Set to set distance: Choose distance Value (For Eg., 100)

In the **Stripe Boundary section**:

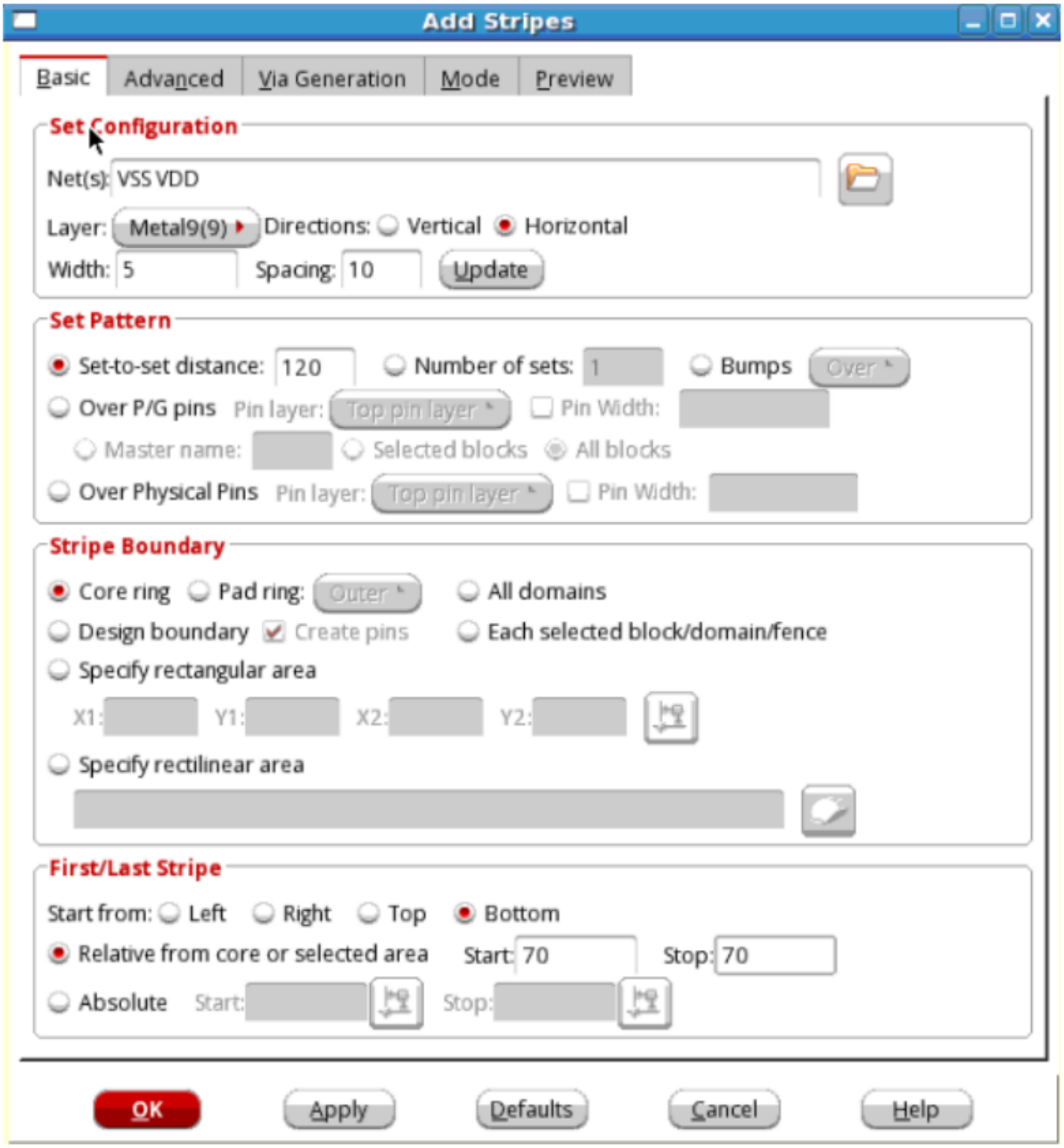
Select – **Core ring**

In the **First/Last Stripe section**:

Select – **Start from: bottom**

Select – **Relative from core or selected area**

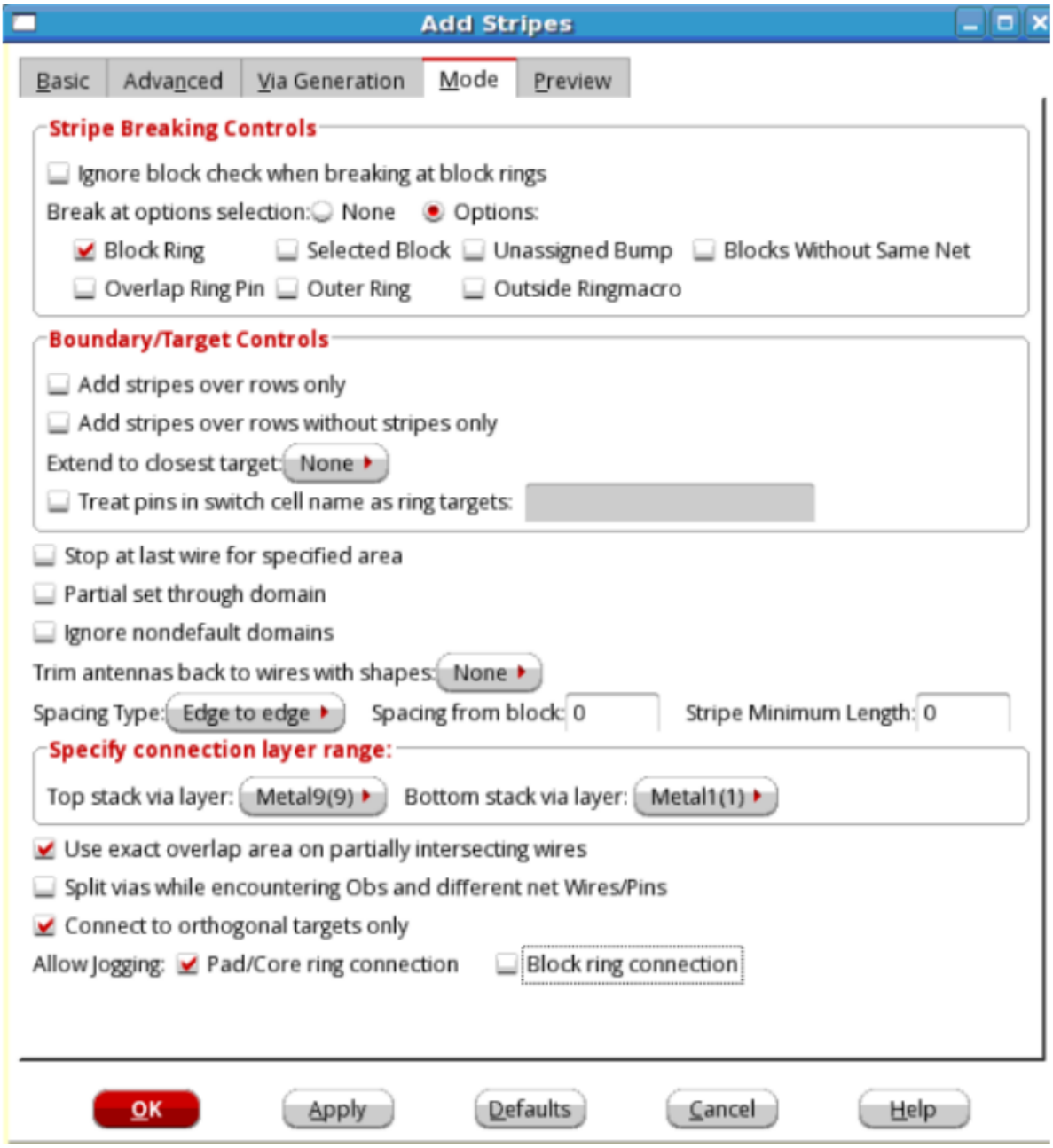
Specify - **Start**: __ **Stop**: __



In the **Mode** tab:

Under **break at options selection**→ select radio button **options** → select **Block ring**. [This will break stripes so they do not go over the blocks.]

Under **Allow jogging** → Deselect **Block Ring Connection**



Click **Apply.**

Only **horizontal power and ground stripes** created for the design core.

To **Add Vertical Power and Ground Stripes only the following change has to be done in GUI;**

In Basic Tab → Choose the other **Metal** Layer and **Direction** as **Vertical**

All other settings are same as previous adding horizontal power and ground stripes.

Now only **Click OK**

Command to add Stripes:

#Adding Horizontal Stripes

```
addStripe -nets { VSS VDD } -layer M2 -width $swidth -direction Horizontal -spacing $pspace -xleft_offset $soffset -set_to_set_distance $sspace -block_ring_top_layer_limit M3 -block_ring_bottom_layer_limit M1 -padcore_ring_bottom_layer_limit M1 -padcore_ring_top_layer_limit M3 -stacked_via_top_layer M3 -stacked_via_bottom_layer M1 -max_same_layer_jog_length 3.0 -snap_wire_center_to_grid Grid -merge_stripes_value 1.5
```

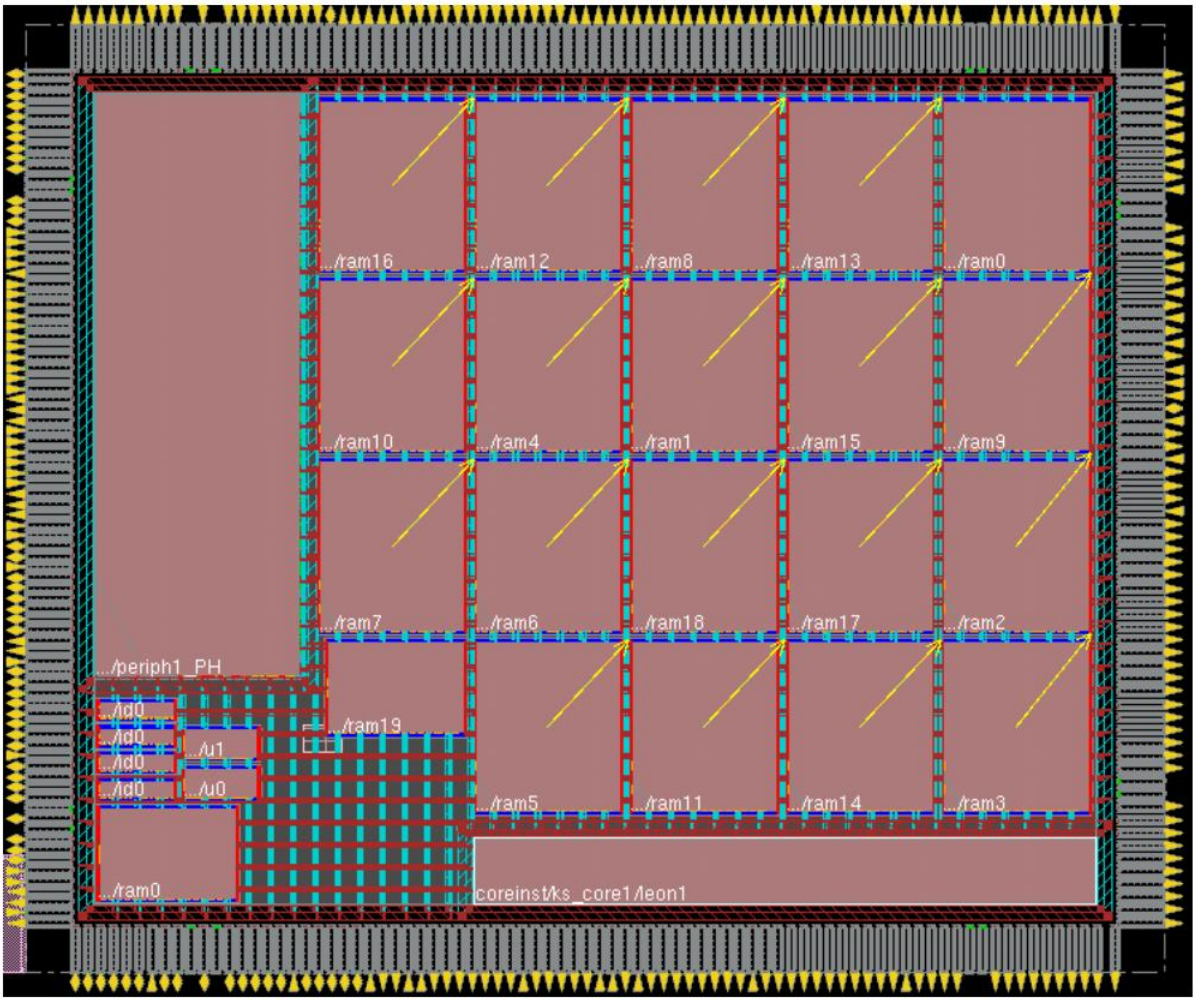
#Adding Vertical Stripes

```
addStripe -nets { VSS VDD } -layer M1 -width $swidth -direction Vertical -spacing $pspace -xleft_offset $soffset -set_to_set_distance $sspace -block_ring_top_layer_limit M3 -block_ring_bottom_layer_limit M1 -padcore_ring_bottom_layer_limit M1 -padcore_ring_top_layer_limit M3 -stacked_via_top_layer M3 -stacked_via_bottom_layer M1 -max_same_layer_jog_length 3.0 -snap_wire_center_to_grid Grid -merge_stripes_value 1.5
```

Note:

For larger designs, use multi-threading to generate stripes.

Run "`SetMultiCpuUsage`" Command to specify number of CPUS to use and then Run to add Stripes



9. Power and Ground Routing

Special Routing (or) SRoute is used to route the Power and Ground Nets.

SRoute routes **block pins, pad pins, pad rings and standard cell pins.**

Select **Route → Special Route**

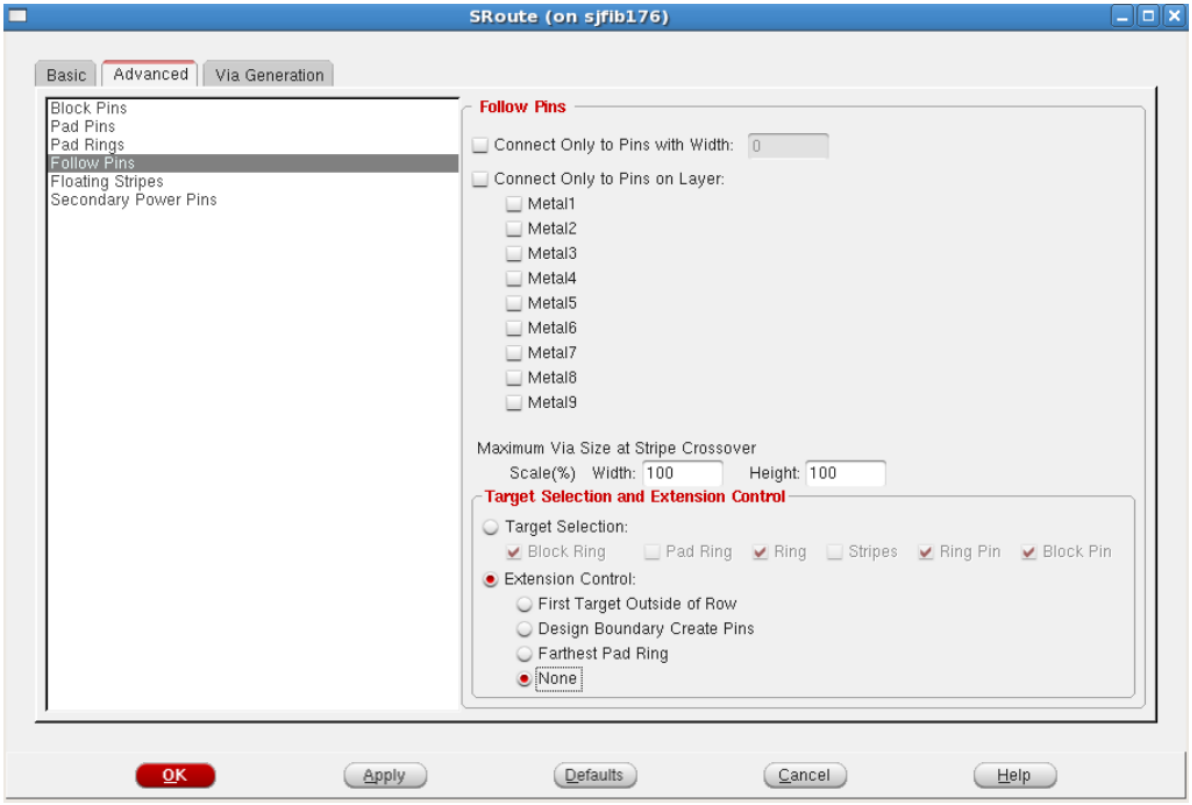
In **Basic Tab;**

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

Deselect **Pad Rings and Floating Stripes**

In **Advance** tab:

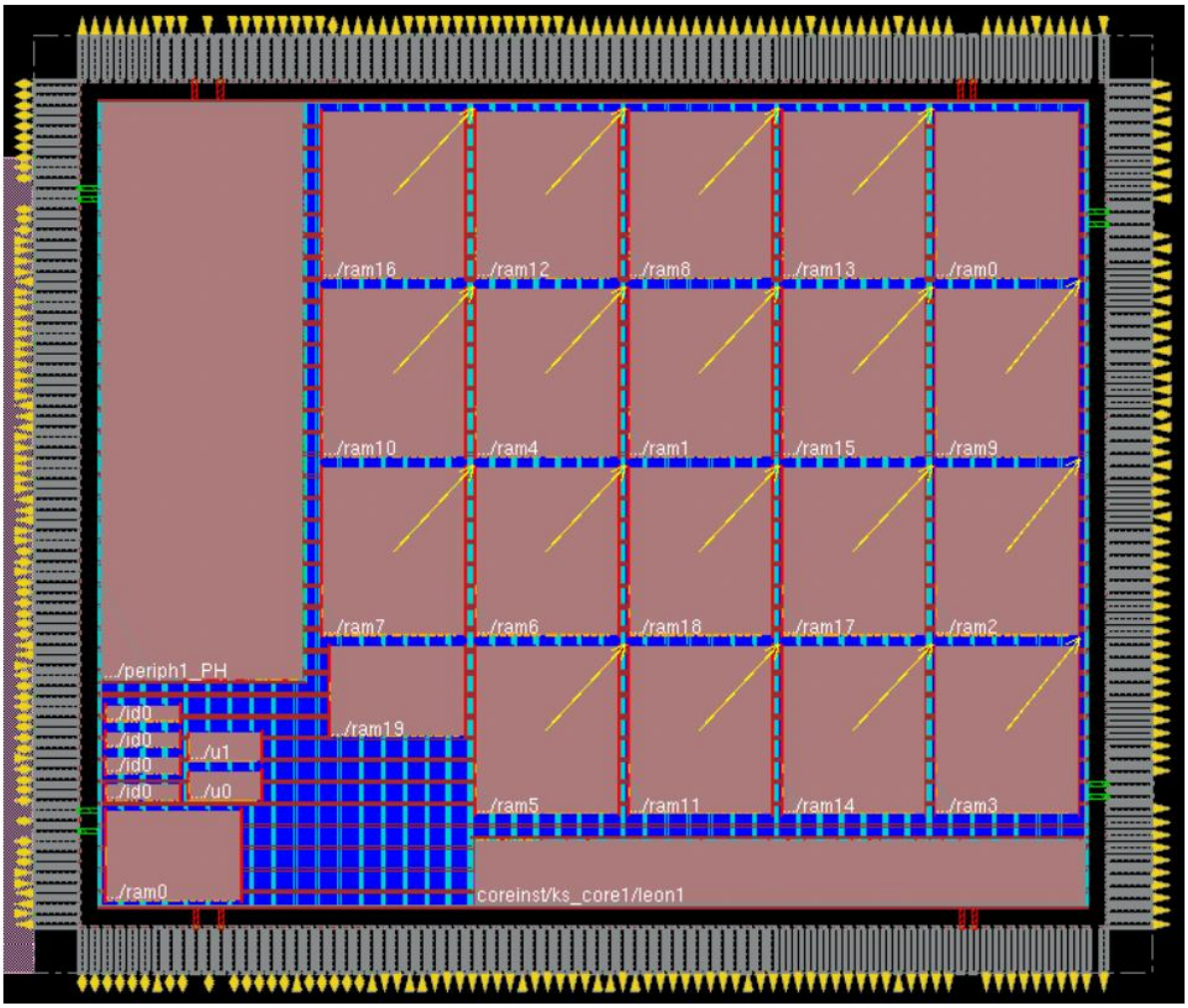
Select **follow pins** in left column → In **Extension Control** → Select **none** → **OK**



This creates the standard cell power rails on Metal1 horizontally across the core.

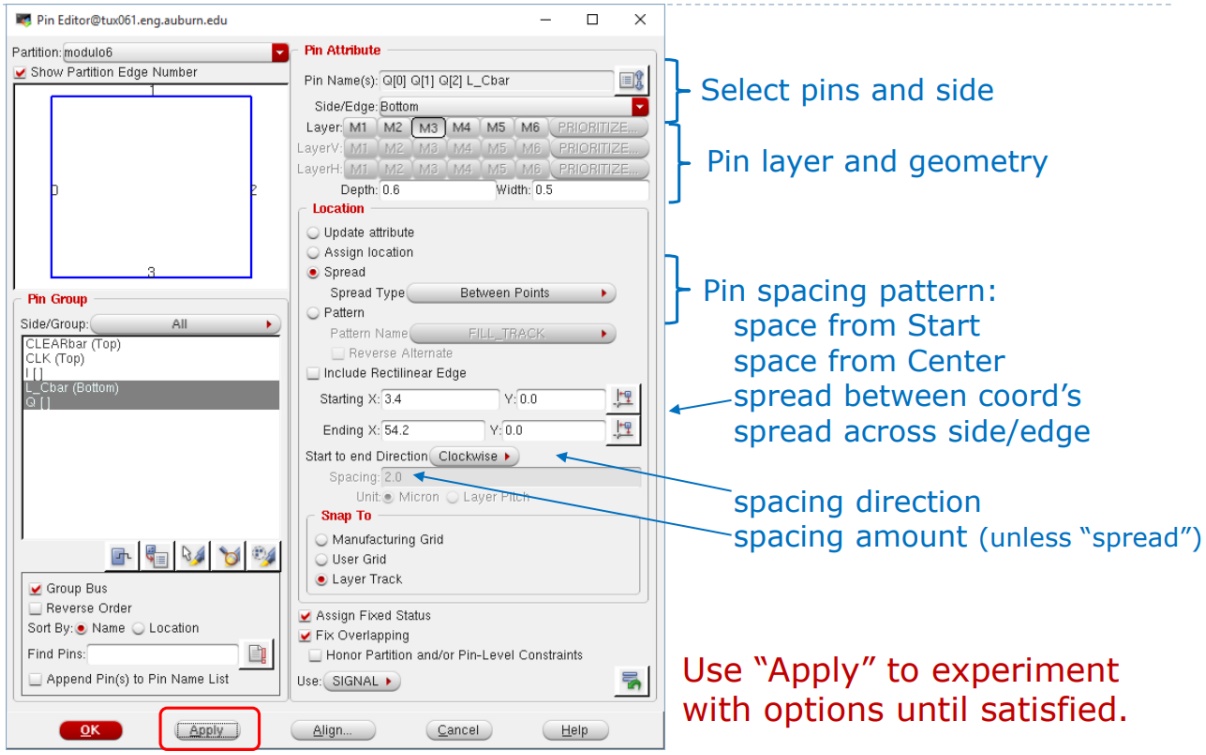
Command to Route Power and Ground:

```
sroute –connect {blockPin padRing corePin } –allowJogging true –allowLayerChange true –blockPin useLef –targetViaLayerRange {M1 AM }
```



Editing the Pins

If there is a necessity in editing the placement of pins, it can be done by **Pin Editor Form or Command**



In Command;

```
editPin -side TOP -layer MetalLayer -fixedPin 1 -spreadType CENTER
-spacing 6 -pin { List if Pins }
```

(or)

```
editPin -side BOTTOM -layer MetalLayer -fixedPin 1 -spreadType
RANGE -start {10 0} -end {40 0} -spreadDirection CounterClockwise
-pin { List if Pins }
```

Save the Design;

```
saveDesign *.enc
```

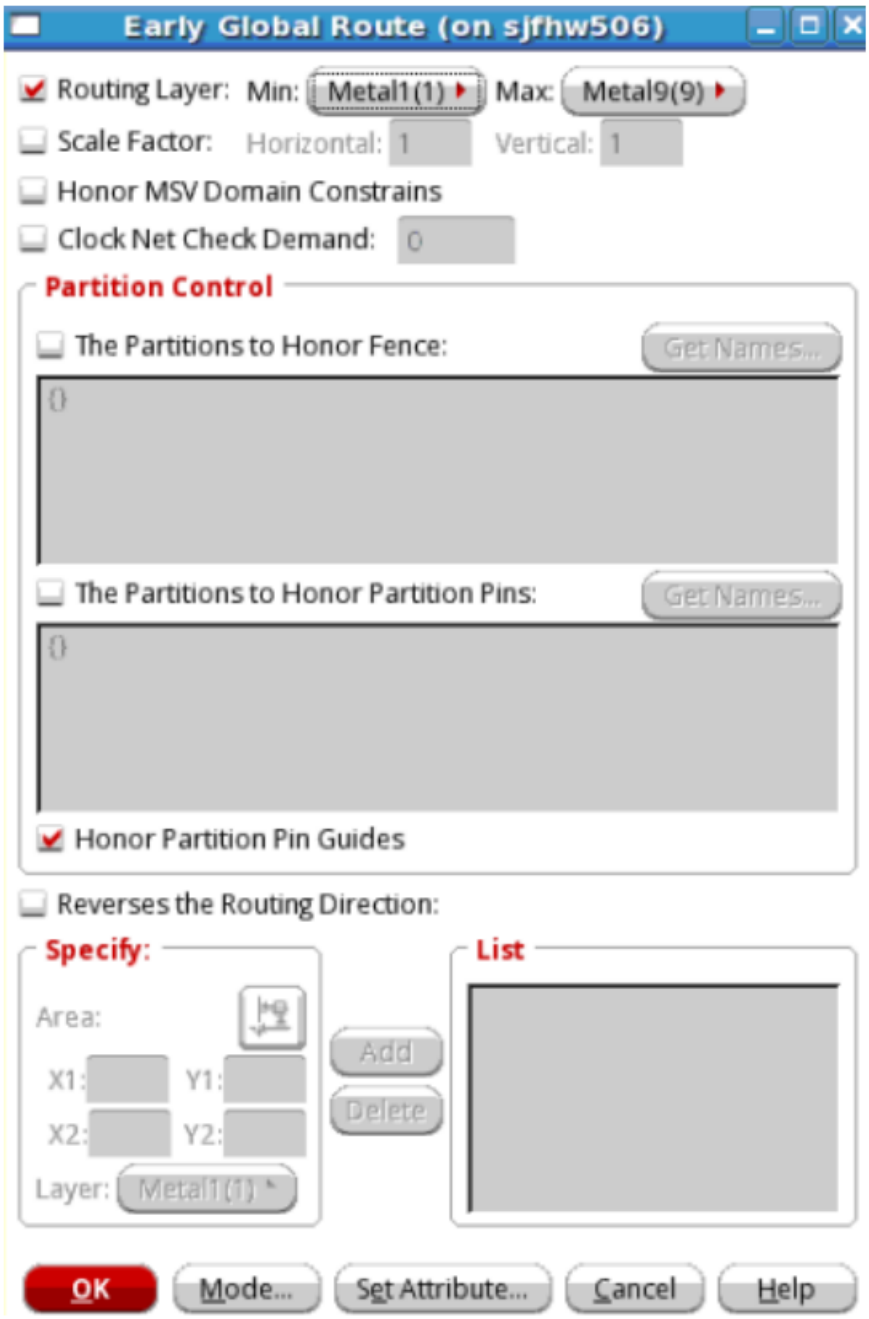
Place the Design:

Place → Place Standard Cell → OK

Early Global Routing:

Click**, Route → Early Global Route**

Select Range of **Routing Layers (Min & Max) → OK**



Adjusting Floorplan:

The channel width between the objects is adjusted in order to **avoid or minimize the Routing congestion.**

Command:

```
adjustFPlanChannel
```

Checks to be done after Floorplan:

```
checkDesign –all
```

```
checkFPlan -reportUtil

report_power

report_constraint

#Fix Macros

setBlockPlacementStatus -allHardMacros -status fixed
```

Save the Floorplan Design

```
saveFPlan *.fp
```

Proceed to the Following if you find your floor plan is well finished by carrying out the above steps:

Placement

Clock Tree Synthesis

Routing

Acknowledgements

https://www.cadence.com [Floor planning]

Dr. Adam Teman, EnICS Labs

https://www.edn.com/floorplanning-concept-challenges-and-closure/

https://en.wikipedia.org/wiki/LEON