

PICO-RV DESIGN REPORT

Srujan Reddy Annayyagari

RTL-GDS II Training and Internship by VLSI EXPERT.

1. Introduction

- Overview of the Task Environment
- Opening the Design in ICC2

2. Floorplanning

- **2.1 Floorplan Task 1-----page no. 04**
 - Die Boundary Control
 - Pin Placement Strategy
 - Core Utilization and Offset
 - Standard Cell Placement
 - Saving the Floorplan
- **2.2 Floorplan Task 2-----page no. 12**
 - Rectilinear Floorplan Initialization
 - Core Boundary and Utilization
 - Pin Placement by Side
 - Pin Placement Distance
 - Saving the Floorplan

3. Power Planning

- **3.1 Powerplan Task 1-----page no. 19**
 - U-Shaped Floorplan Initialization
 - Input/Output Pin Placement
 - Hard Blockage Specification
 - Ring Pattern Creation (M7, M8)
 - Mesh Pattern Creation (M6, M7)
 - Standard Cell Power Rails (M1)
 - PG Strategy Setup
 - Saving the Power Plan
- **3.2 Powerplan Task 2-----page no. 29**
 - T-Shaped Floorplan Initialization
 - Soft Blockage Specification
 - Ring Pattern Creation (M8, M9)
 - Mesh Pattern Creation (M7, M8)
 - Standard Cell Power Rails (M1)
 - PG Strategy Setup
 - Saving the Power Plan

1. Introduction

Overview of the Task Environment

The following tasks are carried out in the **ICC2 (IC Compiler II) Shell** environment, a leading physical implementation tool from Synopsys, widely used for place-and-route in digital design flows. All tasks are performed on the **picorv32** block, which is part of the **picorv_lib** design library.

Each task in this documentation builds on the **picorv32** block and focuses on mastering floorplanning and power planning techniques such as shaping die boundaries, pin placement strategies, creating ring and mesh power grids, and implementing standard cell power rail strategies.

Opening the design in ICC2

Before starting any of the tasks, ensure that the design library and block are opened correctly in the ICC2 shell. The following commands are used to load the project and launch the GUI:

open_lib PICORV_LIB/

open_block picorv32

start_gui Displays the picorv32 design in the GUI

2. FLOORPLANNING

2.1 Floorplan Task 1

Objective:

To define and initialize a floorplan for the picorv32 design using **ICC2** shell with the following requirements:

- **Die Shape:** Rectangular, where the vertical sides are 4x longer than the horizontal sides.
 - **Core Utilization:** 90% (0.9)
 - **Core Offset:** 10µm horizontally and 20µm vertically.
 - **Input Pins:** All input pins except `clk` must be placed on sides **3** and **4**.
 - **Output Pins:** All output pins must be placed on sides **1** and **2**.
 - **Clock Pin (`clk`):** Can be placed on **any side**; here, it is assigned to **side 2**.
 - **Standard Cell Area:** Must lie **within the core**.
 - **Floorplan Save Name:** `fp_task1`
-

Steps with Commands and Description

Step 1: Open the Library and Design Block

```
open_lib picorv_lib/  
open_block picorv32  
start_gui
```

- Opens the **design library** `picorv_lib` and loads the **block** `picorv32` into ICC2.
 - `start_gui` launches the graphical interface.
-

Step 2: Link the Block

```
link_block
```

- Links the design block to resolve all design elements before editing.
-

Step 3: Initialize Floorplan

```
initialize_floorplan -control_type die -side_length {100 400} -core_offset {10 20} -core_utilization 0.9
```

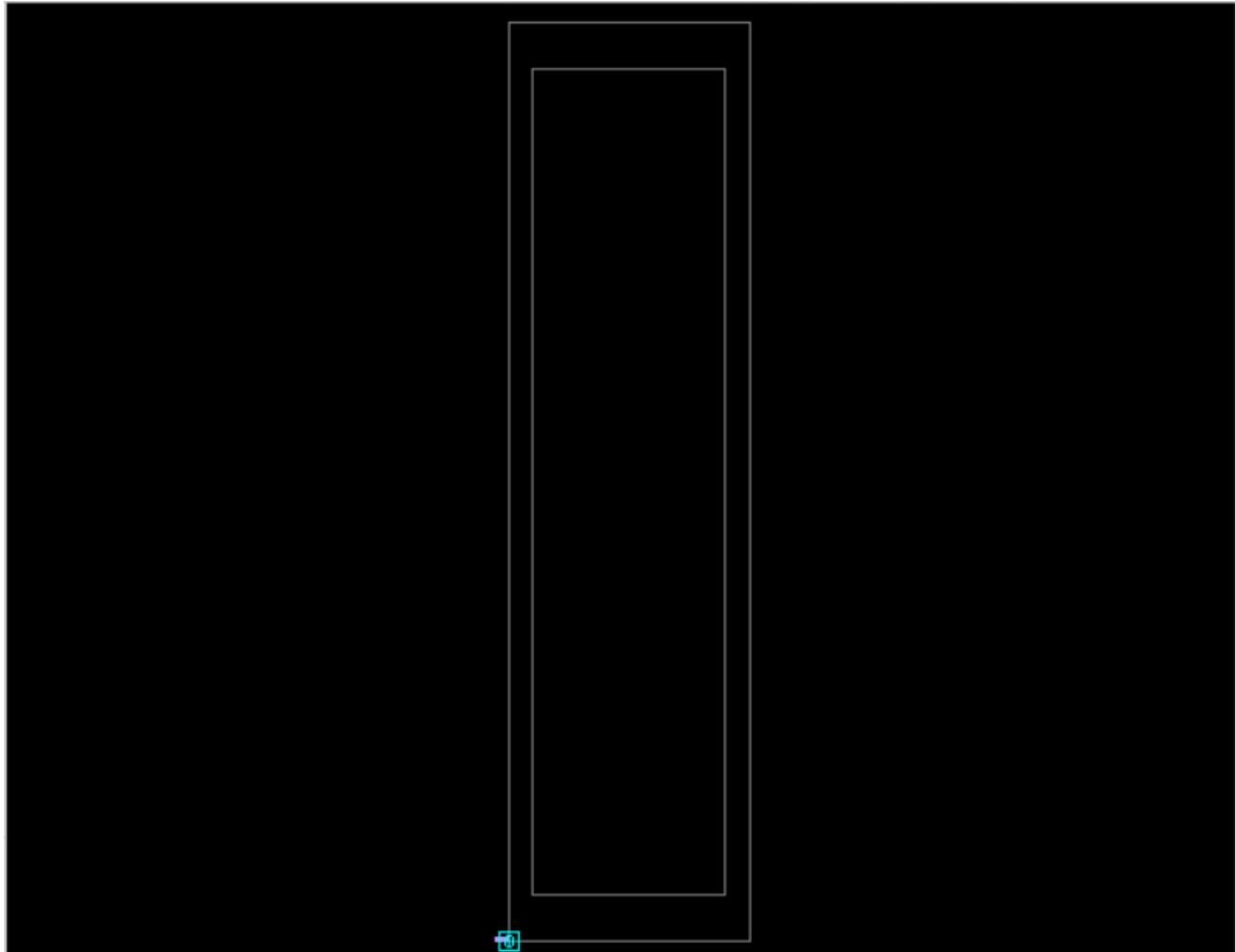
- Initializes the **floorplan with explicit die dimensions** of 100 μm × 400 μm .
- Sets **core offset** (10 μm horizontal, 20 μm vertical).
- **Core utilization** of 90%.

Optional Alternative Command (if side lengths are not fixed):

```
initialize_floorplan -side_ratio {1 4} -core_utilization 0.9 -control_type die -core_offset {10 20}
```

- Automatically scales the floorplan while maintaining a 1:4 **aspect ratio** for die sides.
- Useful when dimensions can be flexible but proportion must be maintained.

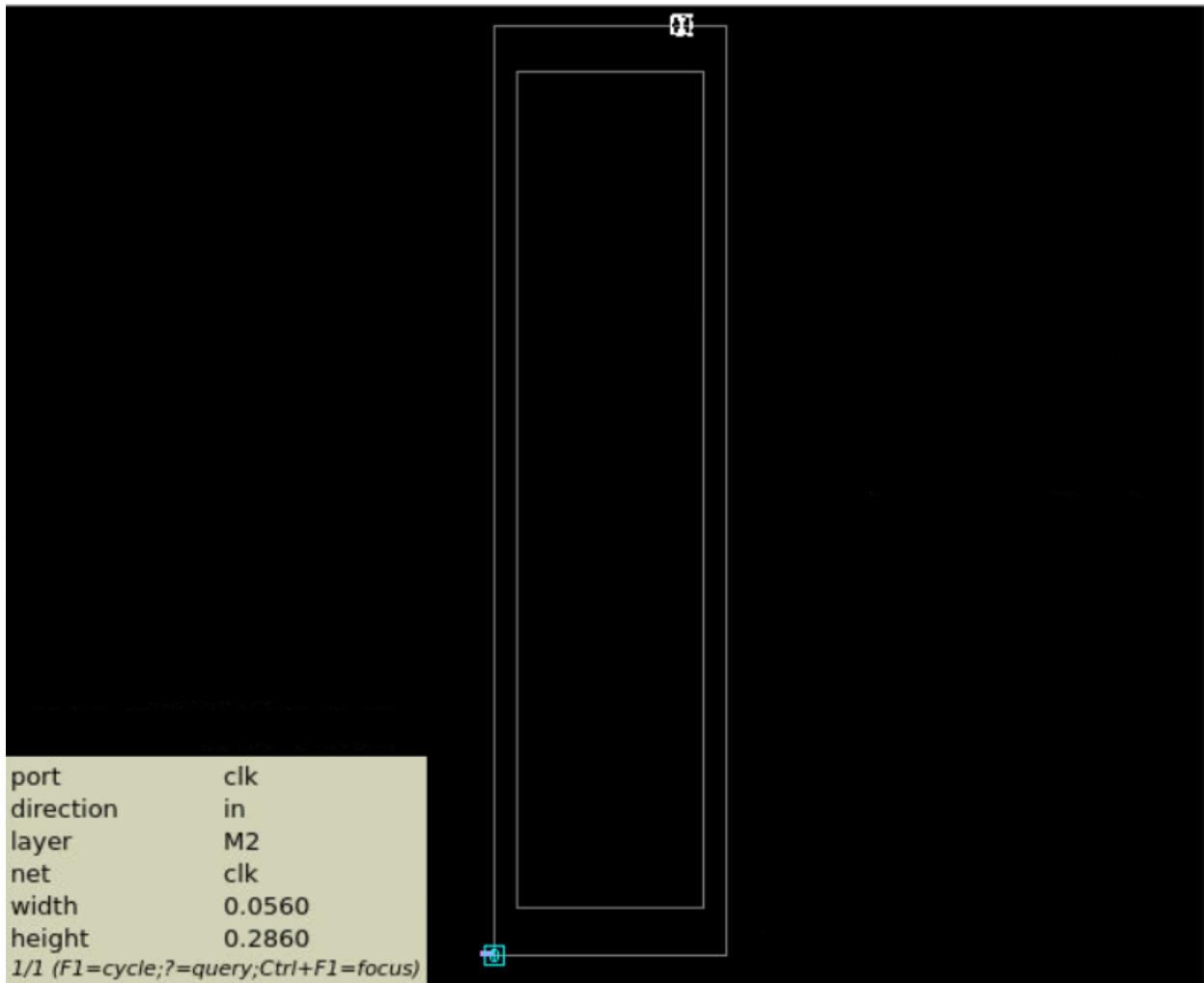
```
icc2_shell> initialize_floorplan -control_type die -side_length {100 400} -core_offset {10 20} -core_utilization 0.9
Warning: this option -core_utilization/-macro_utilization is only valid when you specify the -side_ratio option. (DPI-503)
Removing existing floorplan objects
Creating core...
Core utilization ratio = 84.04%
Unplacing all cells...
Creating site array...
Creating routing tracks...
Initializing floorplan completed.
icc2_shell> █
```



Step 4: Place Clock Pin on Side 2

```
set_individual_pin_constraints -ports [get_ports clk] -sides 2  
place_pins -ports clk
```

- Assigns and places the `clk` input pin on **side 2**.

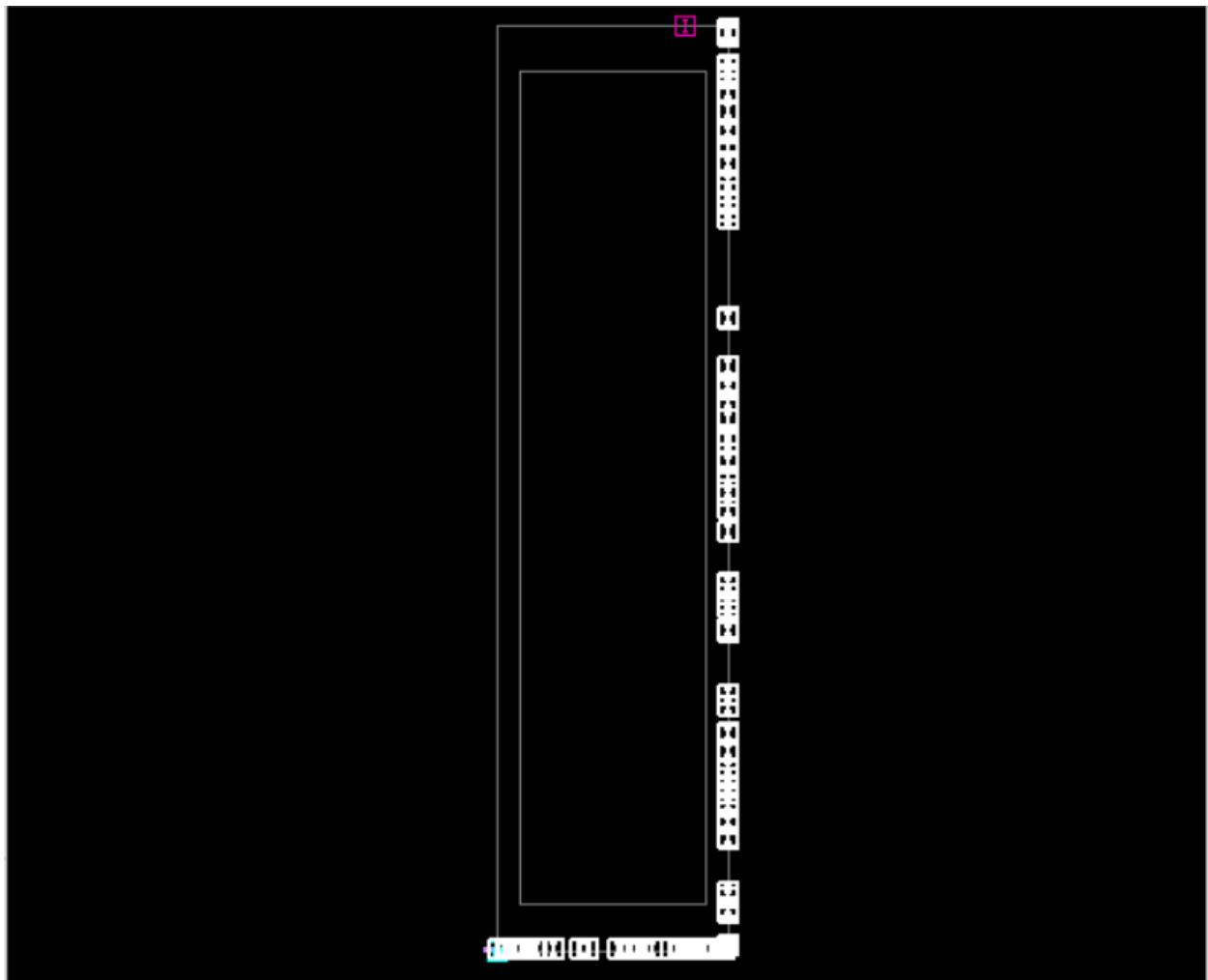


Step 5: Place Remaining Input Pins on Sides 3 and 4

```
set_individual_pin_constraints -ports [remove_from_collection [all_inputs] clk] -sides {3 4}  
place_pins -ports [remove_from_collection [all_inputs] clk]
```

- Filters out `clk` from input ports.

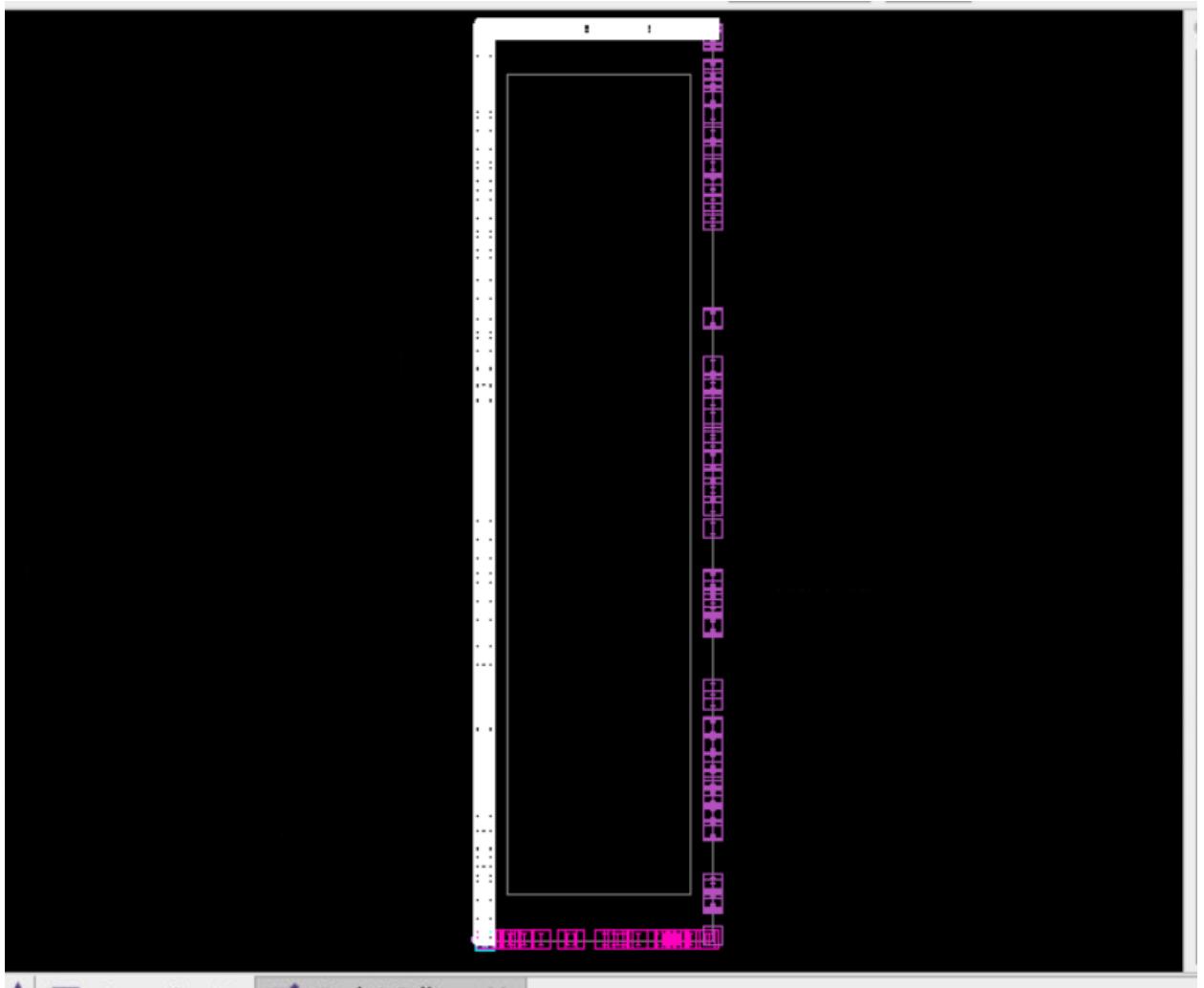
- Places all **other inputs** on **sides 3 and 4**.
- **change_selection [[remove_from_collection [all_inputs] clk]]**



Step 6: Place Output Pins on Sides 1 and 2

```
set_individual_pin_constraints -ports [all_outputs] -sides {1 2}
place_pins -ports [all_outputs]
change_selection [all_outputs]
```

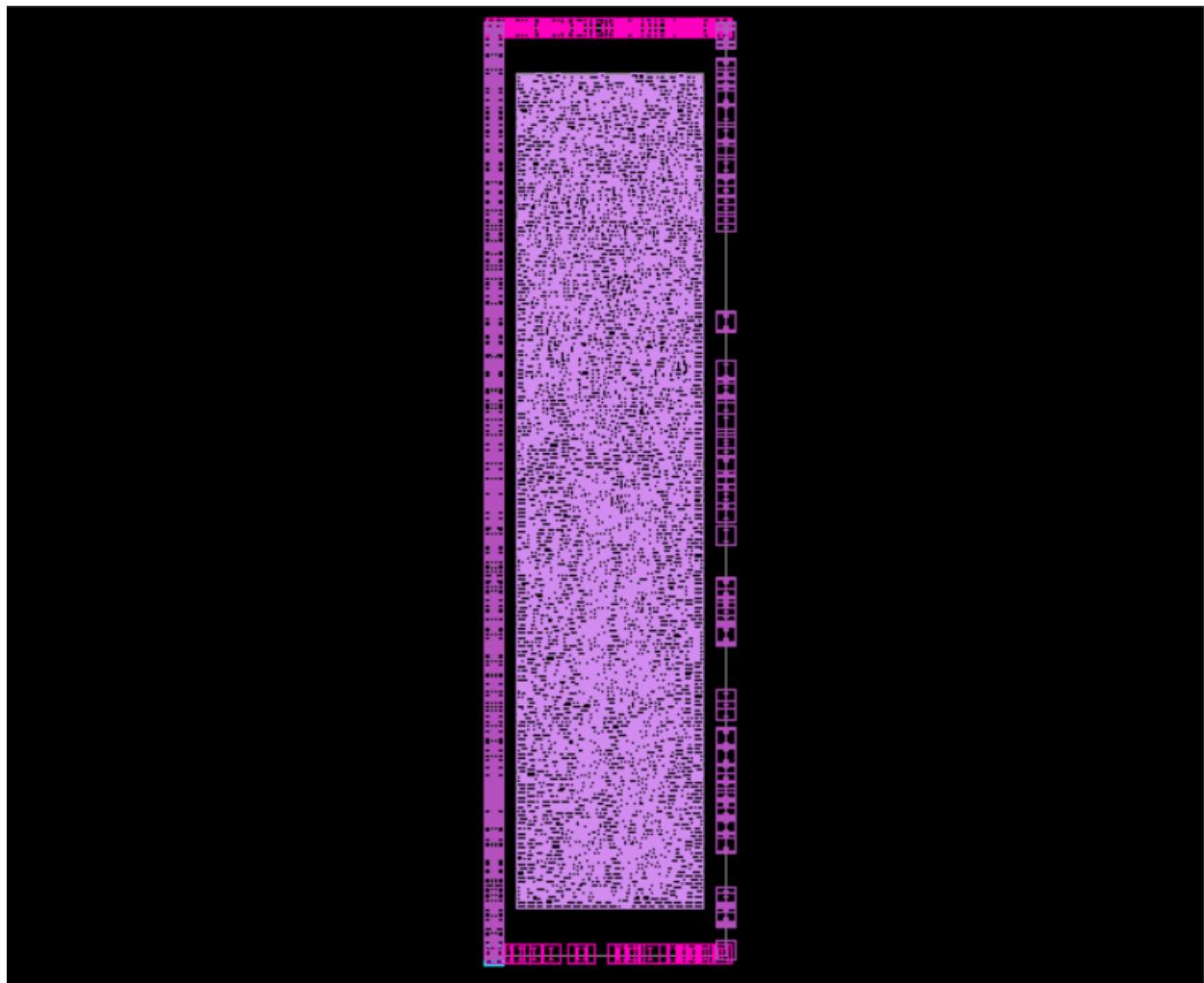
- Places all output ports on **sides 1 and 2**.



Step 7: Create Initial Standard Cell Placement

```
create_placement -floorplan
```

- Performs **initial placement** of cells inside the defined core area.



Step 8: Save the Floorplan

`save_block -as fp_task1`

- Saves the current floorplan and placement state under the name `fp_task1`.

Outcome:

The floorplan is successfully created with:

- A 1:4 aspect ratio
- 90% core utilization
- Input/output pin placement per specification
- Standard cell region defined and initialized

Reports :

Cell Instance Type	Count	Area
TOTAL LEAF CELLS	7071	24153.592
unit	7071	24153.592
Standard cells	7071	24153.592
unit	7071	24153.592
Hard macro cells	0	0.000
unit	7071	24153.592
Soft macro cells	0	0.000
unit	7071	24153.592
Always on cells	0	0.000
unit	7071	24153.592
Physical only	0	0.000
unit	7071	24153.592
Fixed cells	0	0.000
unit	7071	24153.592
Moveable cells	7071	24153.592
unit	7071	24153.592
Placed cells	0	0.000
unit	7071	24153.592
Sequential	1624	10640.247
unit	7071	24153.592
Buffer/inverter	340	461.017
unit	7071	24153.592
ICG cells	0	0.000
unit	7071	24153.592

Logic Hierarchies	: 0
Design Masters count	: 44
Total Flat nets count	: 7729
Total FloatingNets count	: 203
Total no of Ports	: 411
Number of Master Clocks in design	: 0
Number of Generated Clocks in design	: 0
Number of Path Groups in design	: 6 (0 of them Non Default)
Number of Scan Chains in design	: 0
List of Modes	: default
List of Corners	: default
List of Scenarios	: default
Core Area	: 34472.600
Chip Area	: 34472.600
Total Site Row Area	: 34472.600
Number of Blockages	: 0
Total area of Blockages	: 0.000
Number of Power Domains	: 1
Number of Voltage Areas	: 1
Number of Group Bounds	: 0
Number of Exclusive MoveBounds	: 0
Number of Hard or Soft MoveBounds	: 0
Number of Multibit Registers	: 0
Number of Multibit LS/ISO Cells	: 0
Number of Top Level RP Groups	: 0
Number of Tech Layers	: 71 (61 of them have unknown routing dir.)
Total wire length	: 0.00 micron
Total number of wires	: 0
Total number of contacts	: 0

2.2 Floorplan Task 2 – Detailed Steps

Objective:

The goal of this task is to create a rectilinear floorplan consisting of **8 sides**, shaped like a staircase, using ICC2. The floorplan should meet the following specifications:

- **Core Utilization:** 0.82
- **Core and Die boundary should not coincide**
- **First row of standard cells should not be flipped**
- **Core Offset:** 2.0 μm in both X and Y directions
- **Pin Placement:**
 - **Input pins** on sides 1 to 4, placed with a pitch of 3 μm
 - **Output pins** on sides 5 to 8, placed with a pitch of 4 μm
 - **Clock pin** can be placed on any side (e.g., side 2)
- Floorplan to be saved as: fp_task2

Commands & Description

Step 1: Open the library and block

```
open_lib picorv_lib  
open_block picorv32  
link_block
```

- Opens the existing design library and loads the block picorv32 for floorplanning.
-

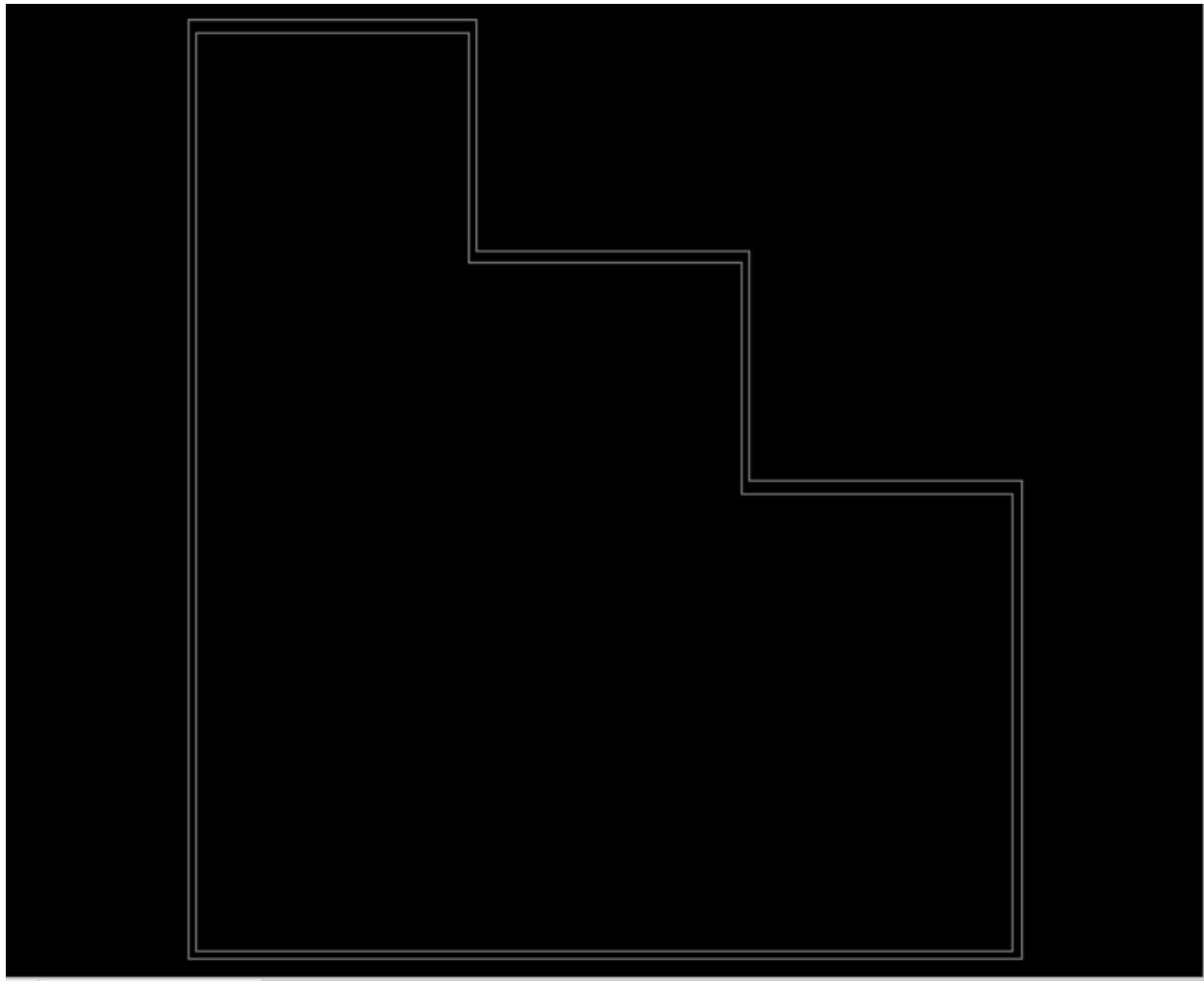
Step 2: Initialize an 8-sided staircase-style floorplan

```
initialize_floorplan \  
-core_utilization 0.82 \  
-coincident_boundary false \  
-flip_first_row false \  
-core_offset {2.0 2.0} \  
-boundary {{0 0} {0 220} {65 220} {65 165} {130 165} {130 110} {195 110} {195 0}}
```

- This creates the die and core boundary based on the coordinates forming a staircase shape. The coincident_boundary is set to false to ensure die and core edges do not overlap. First row flipping is disabled.

```
icc2_shell> report_utilization
*****
Report : report_utilization
Design : picorv32
Version: V-2023.12-SP4
Date   : Mon Apr 21 03:59:17 2025
*****
Utilization Ratio:          0.7565
Utilization options:
  - Area calculation based on:      site_row of block
  - Categories of objects excluded: hard_macros macro
Total Area:                  31929.6356
Total Capacity Area:         31929.6356
Total Area of cells:         24153.5916
Area of excluded objects:
  - hard_macros      :        0.0000
  - macro_keepouts   :        0.0000
  - soft_macros      :        0.0000
  - io_cells         :        0.0000
  - hard_blockages   :        0.0000
Total Area of excluded objects: 0.0000
Ratio of excluded objects:    0.0000

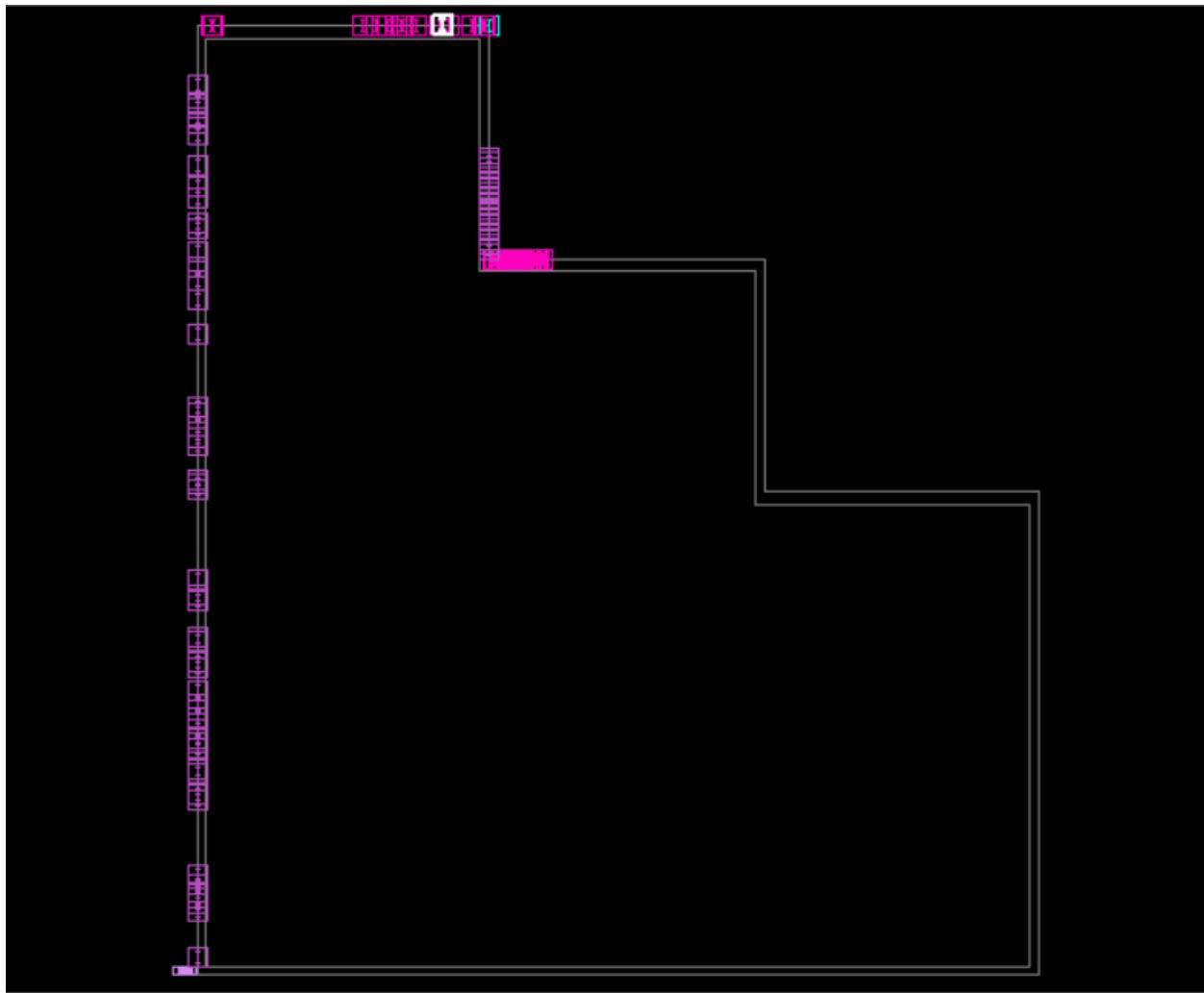
Utilization of site-rows with:
  - Site 'unit':          0.7565
0.7565
```



Step 3: Place input pins on sides 1 to 4 with 3µm pitch

```
set_pin_constraints -pin_spacing 3 -pin_layer 2 -sides {1 2 3 4}  
place_pins -ports [remove_from_collection [all_inputs] clk]
```

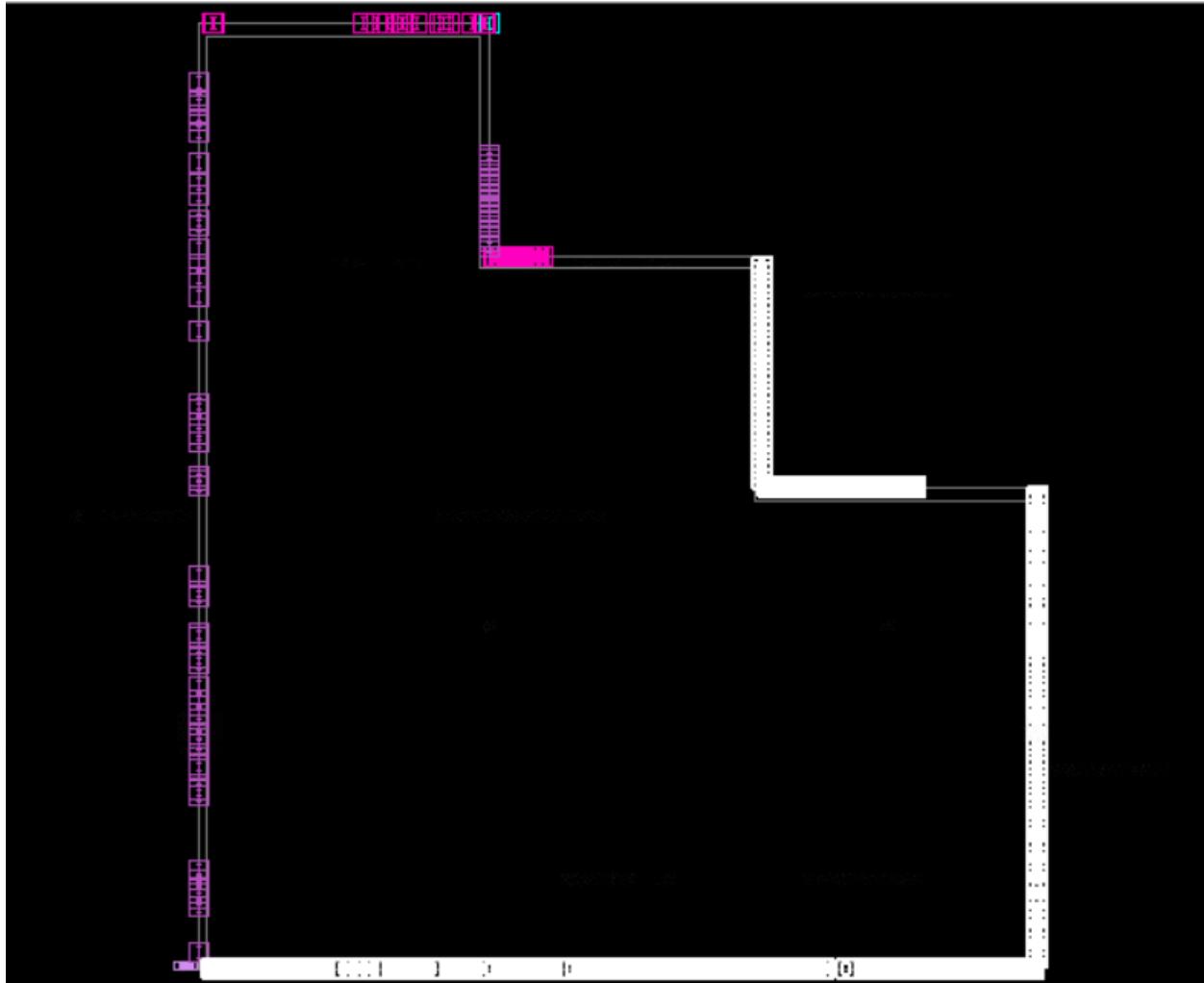
- This ensures all input pins (except clock) are placed on the first four sides with proper spacing.



Step 4: Place output pins on sides 5 to 8 with 4 μ m pitch

```
set_pin_constraints -pin_spacing 4 -pin_layer 2 -sides {5 6 7 8}  
place_pins -ports [all_outputs]
```

- This places the output pins evenly across the remaining four sides of the die.



Step 5: Place the clock pin (can be on any side; here, side 2)

```
set_individual_pin_constraints -ports [get_ports clk] -sides 2
```

```
place_pins -ports clk
```

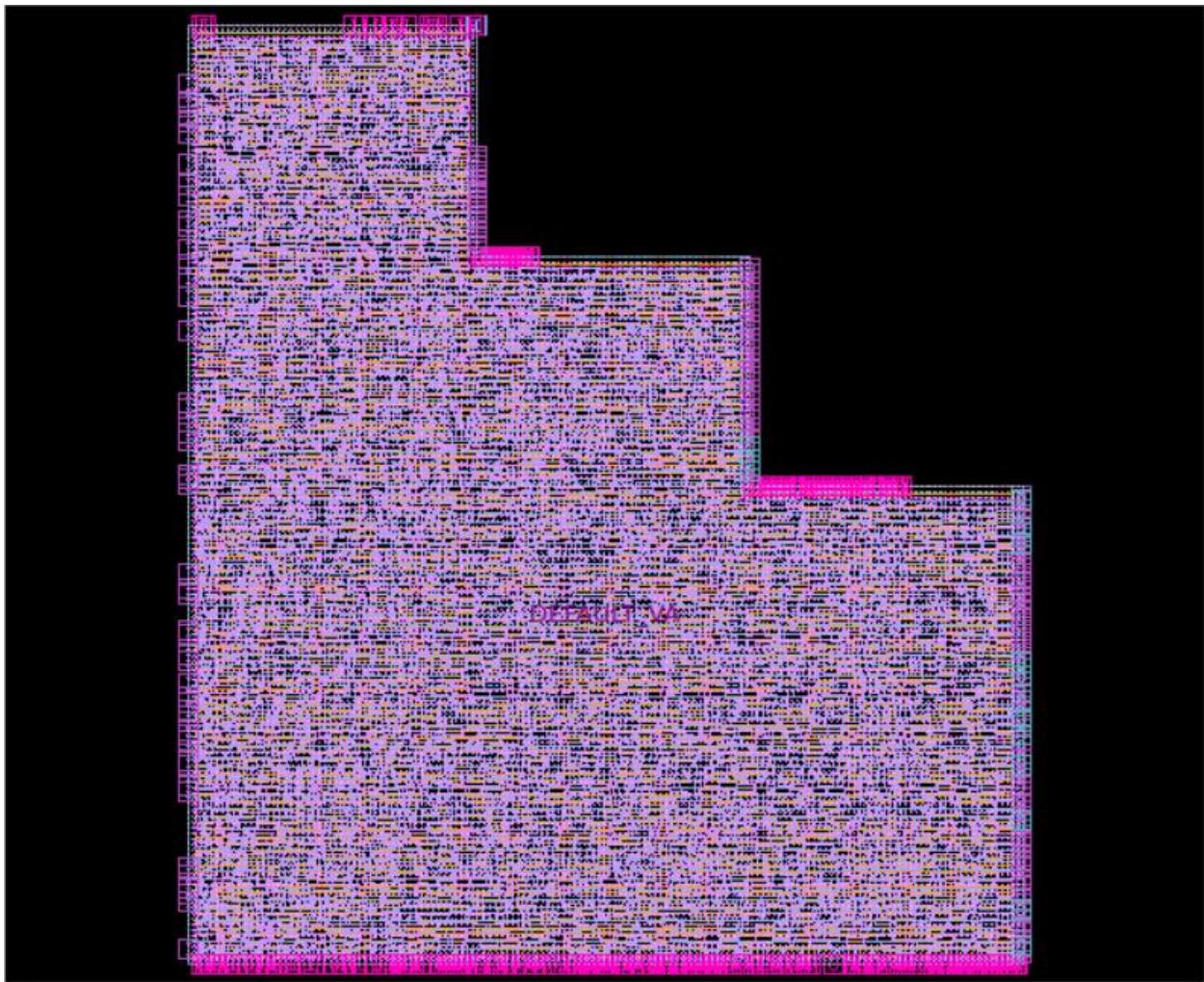
- Clock pin is handled separately and placed individually to allow optimal clock tree design later.



Step 6: Create initial standard cell placement inside the core

```
create_placement -floorplan
```

- This places all the standard cells inside the defined core region according to the initialized floorplan.



Step 7: Save the floorplan

```
save_block -as fp_task2
```

- The design state is saved under the name fp_task2 for future reference and continuation.

3. Power Planning

3.1 Powerplan Task 1 :

Objective:

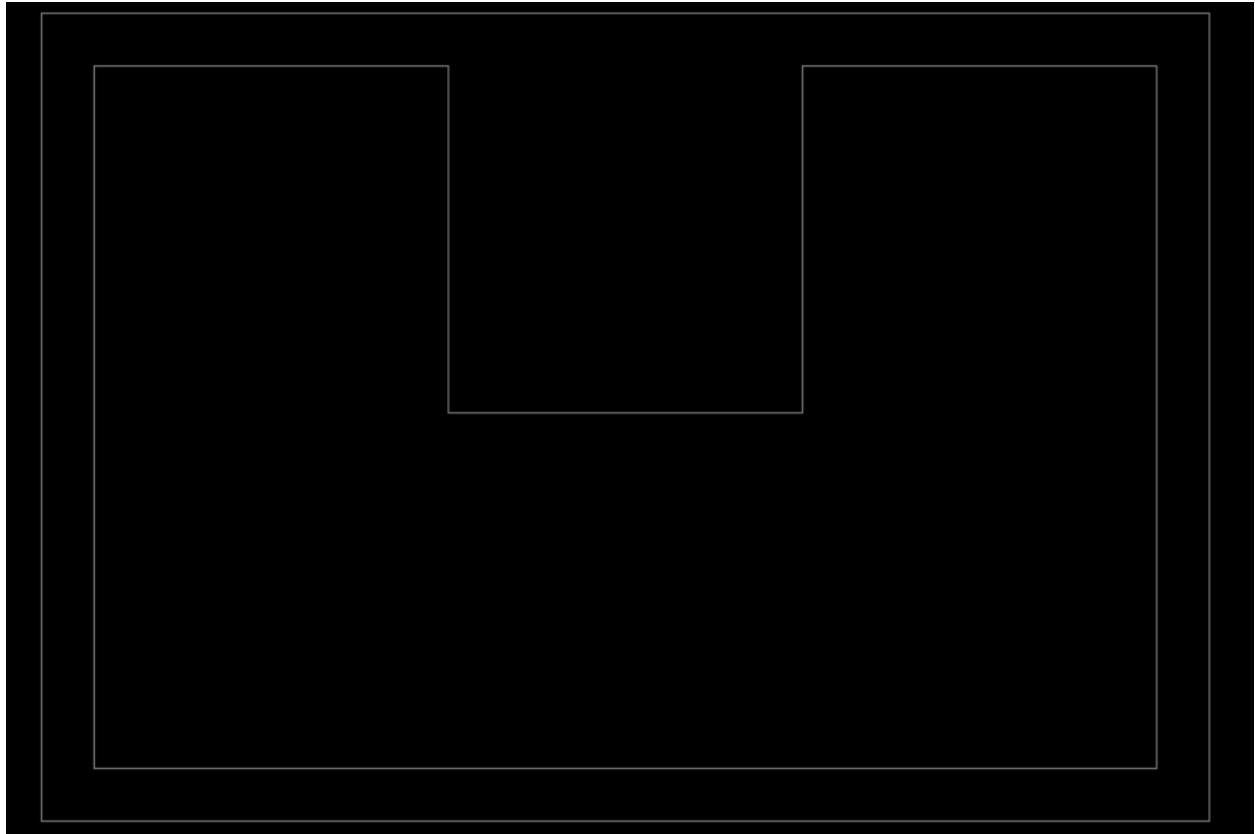
To perform a power planning flow by defining the **floorplan**, **pin placement**, **blockages**, and **power grid strategies** (ring, mesh, and standard cell rails), using appropriate metal layers and constraints as per foundry rules.

Step 1: Floorplan Initialization

- **Shape:** U-shaped core area
- **Core utilization:** 0.75 (75%)
- **Offset:** Maintained to ensure routing space between core and boundary
- **Command:**

```
initialize_floorplan -core_offset 12 -shape U -core_utilization 0.75 -coincident_boundary false
*****
Report : report_utilization
Design : picorv32
Version: V-2023.12-SP4
Date   : Mon Apr 21 11:09:22 2025
*****
Utilization Ratio:          0.7800
Utilization options:
- Area calculation based on: site_row of block pp_task2_completed
- Categories of objects excluded: hard_macros macro_keepouts soft_macros io_cells hard_blockages
Total Area:                 31876.2653
Total Capacity Area:        30964.6508
Total Area of cells:        24153.5916
Area of excluded objects:
- hard_macros      : 0.0000
- macro_keepouts   : 0.0000
- soft_macros      : 0.0000
- io_cells         : 0.0000
- hard_blockages   : 911.6145
Total Area of excluded objects: 911.6145
Ratio of excluded objects:    0.0286

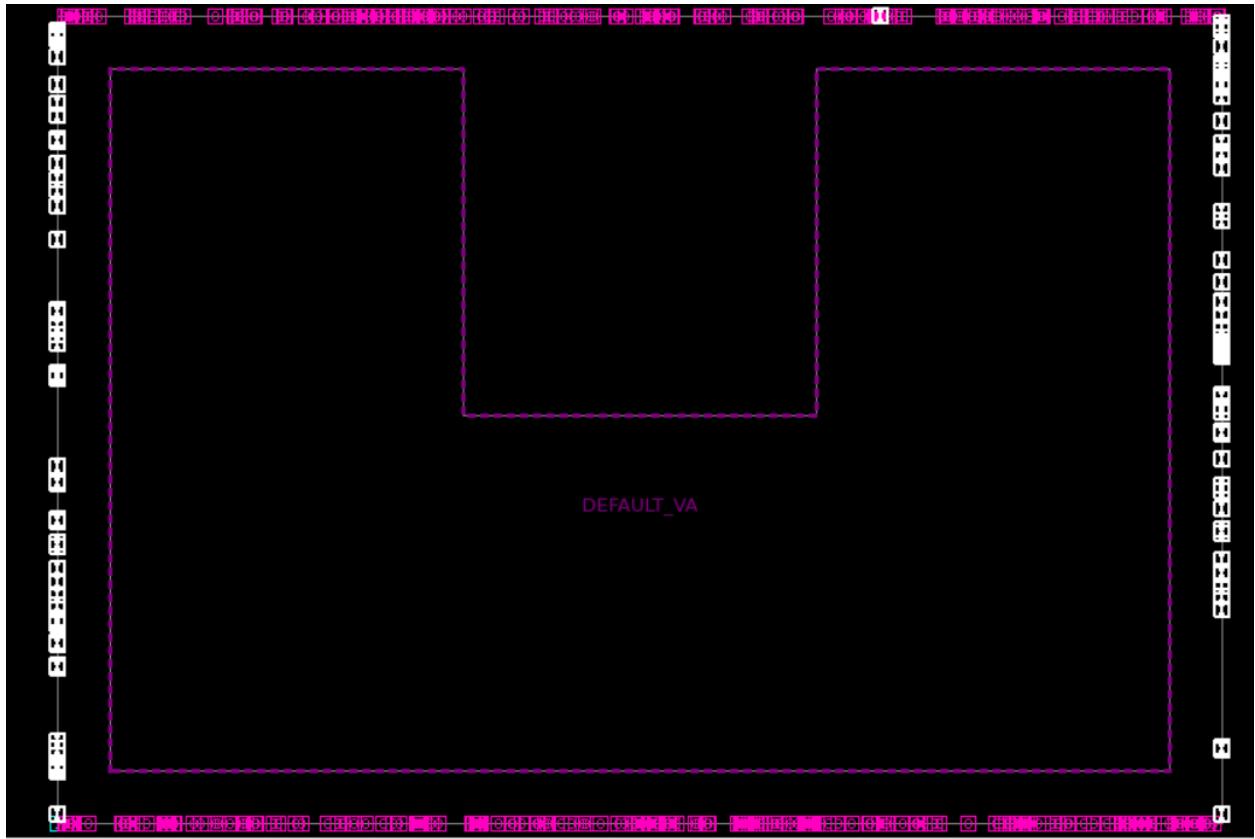
Utilization of site-rows with:
- Site 'unit':           0.7800
0.7800
```



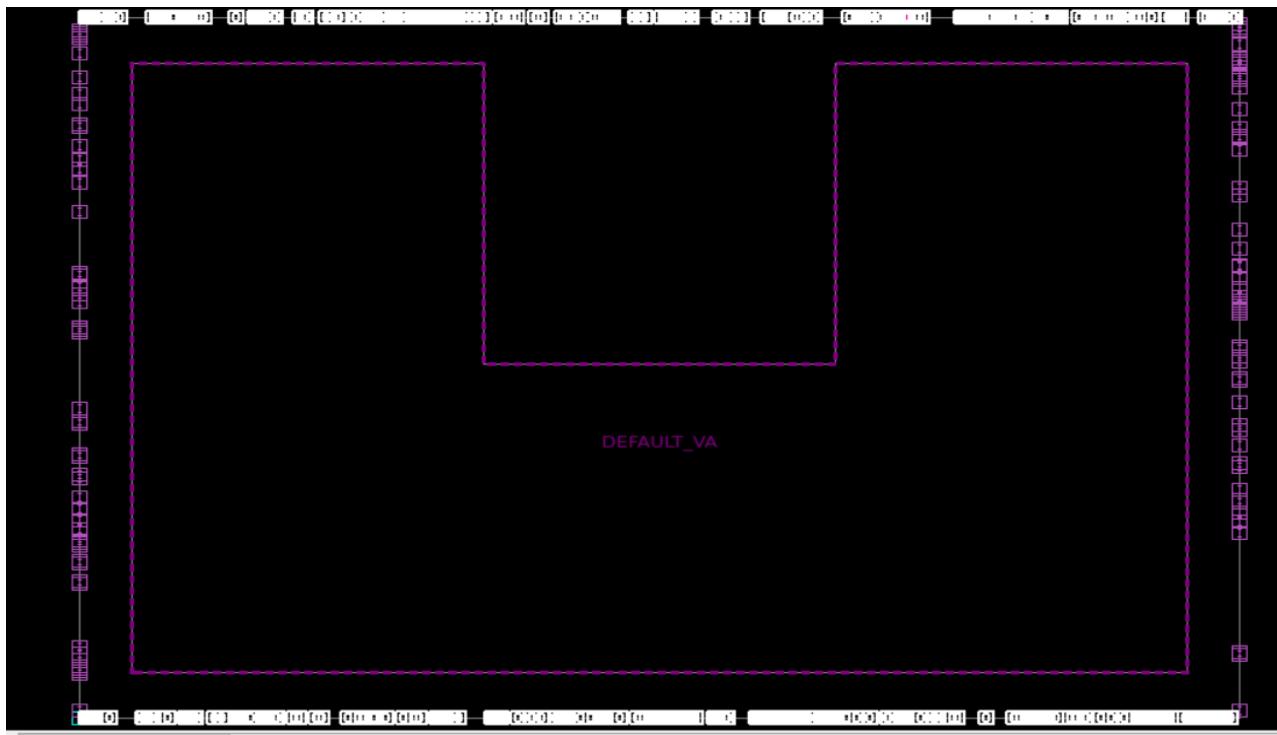
Step 2: Pin Placement

- **Inputs:** Placed on vertical sides (1 and 3)
- **Outputs:** Placed on horizontal sides (2 and 4)
- **Clock:** Can be placed on any side (handled by default)
- **Commands:**

```
set_individual_pin_constraints -ports [all_inputs] -sides {1 3}
place_pins -ports [all_inputs]
```



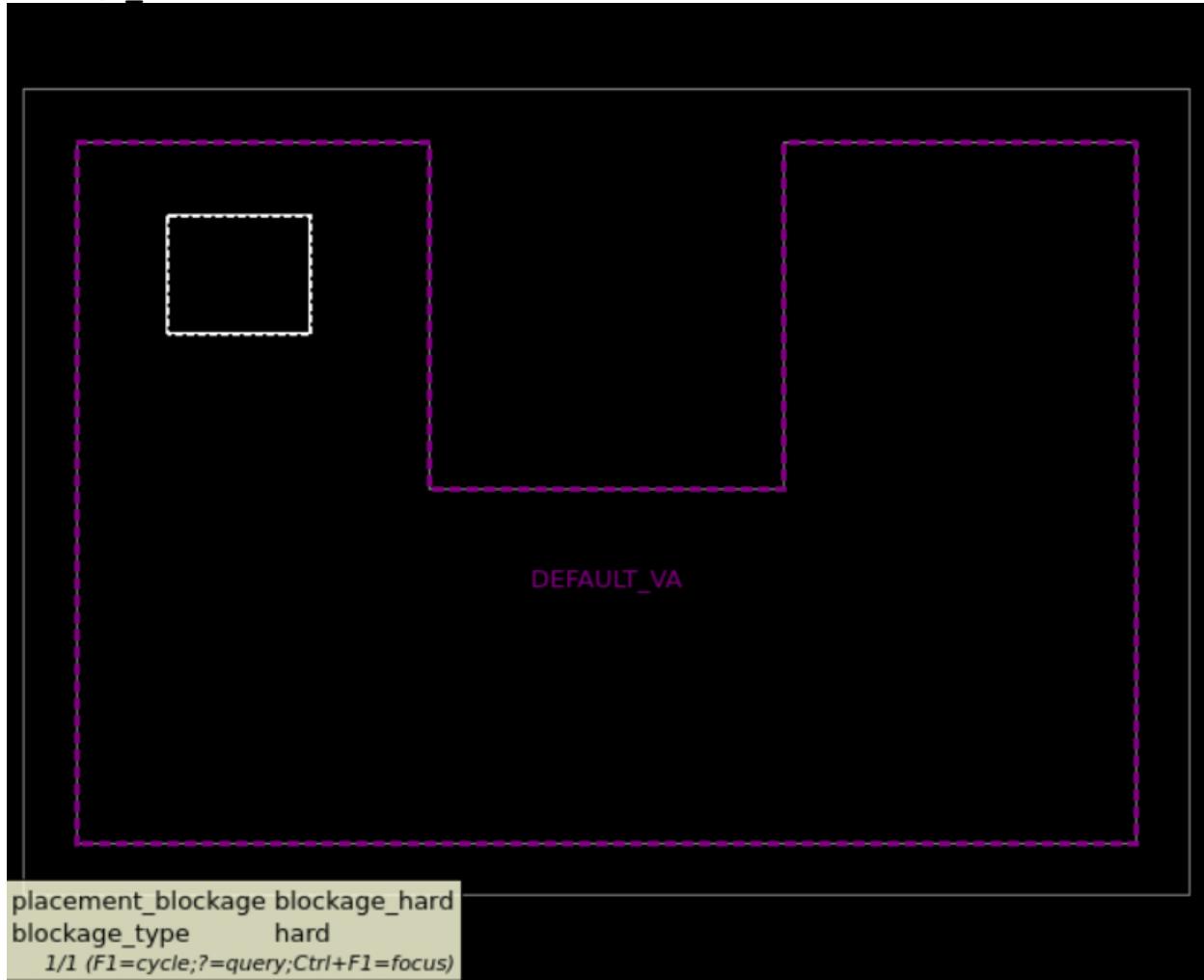
```
set_individual_pin_constraints -ports [all_outputs] -sides {2 4}  
place_pins -ports [all_outputs]
```



Step 3: Blockage Creation

- **Blockage Type:** Hard blockage
- **Location:** On top of the core (coordinates defined below)
- **Command:**

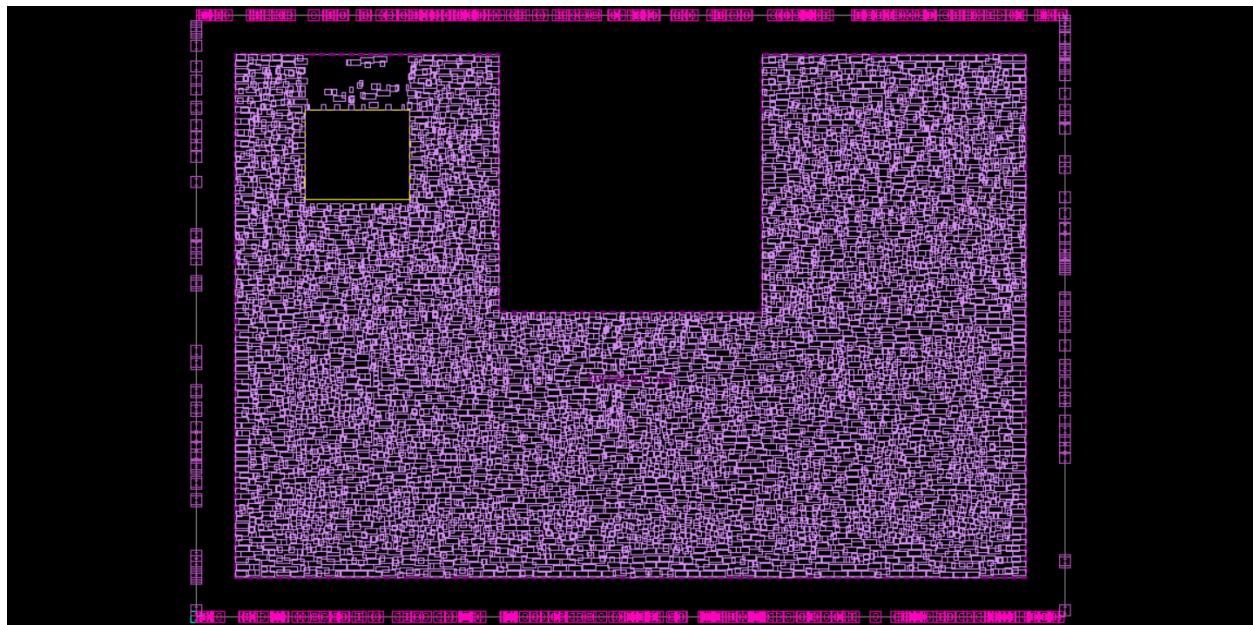
```
create_placement_blockage -type hard -boundary {{33 127} {65 154}} -name  
blockage_hard
```



Step 4: Initial Placement

- **Command:**

```
create_placement -floorplan
```



Step 5: Power Ring Strategy

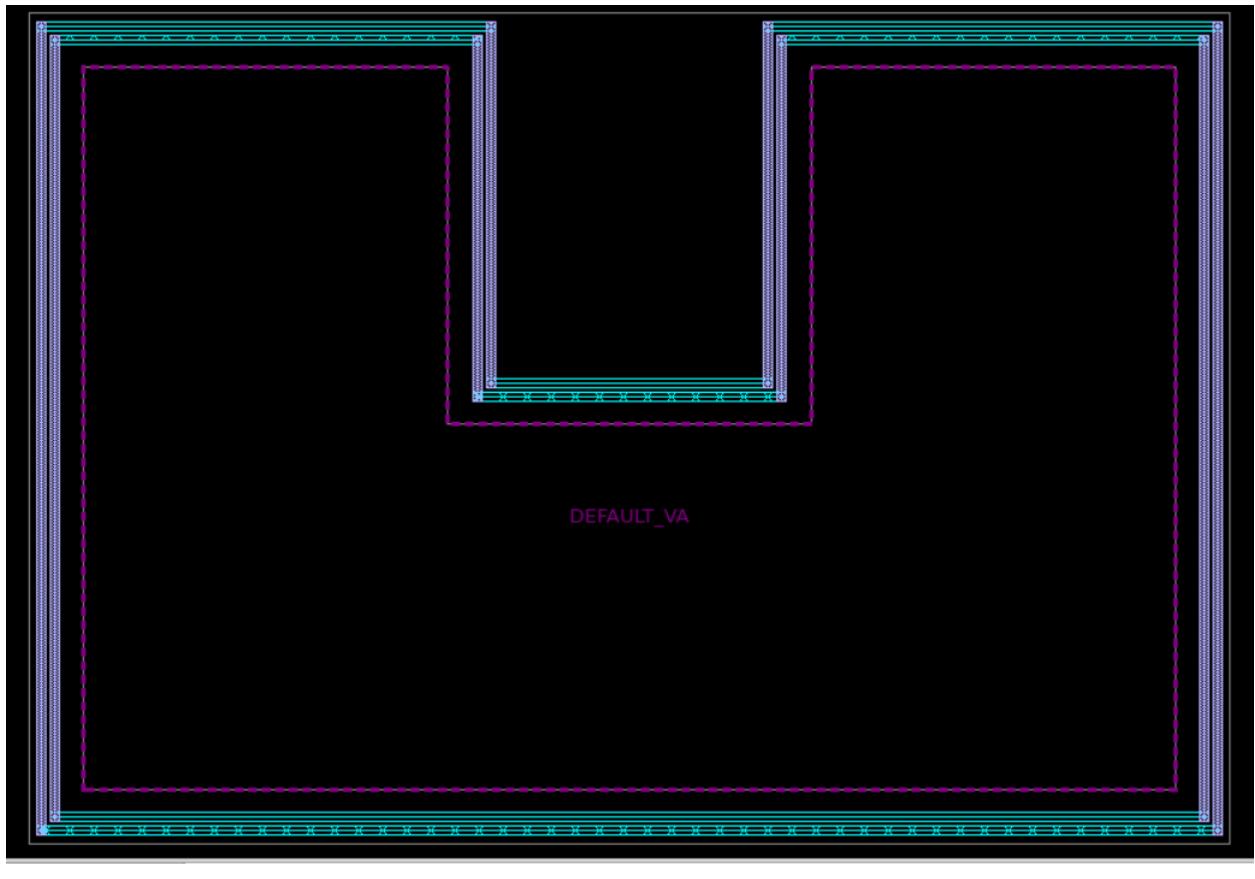
- **Layers Used:**
 - Horizontal: M7 (width = 2)
 - Vertical: M8 (width = 2)
- **Spacing:** 1 (greater than minimum required spacing)
- **Offset:** Maintained ($X > Y \rightarrow 5.5 > 5$)
- **Commands:**

```
connect_pg_net -pg -automatic

create_pg_ring_pattern core_ring \
    -horizontal_layer M7 -horizontal_width 2 -horizontal_spacing 1 \
    -vertical_layer M8 -vertical_width 2 -vertical_spacing 1

set_pg_strategy ring_strategy -core \
    -pattern {{name:core_ring} {nets:{VDD VSS}} {offset:{5.5 5}}}

compile_pg -strategies ring_strategy
```



Step 6: Power Mesh Strategy

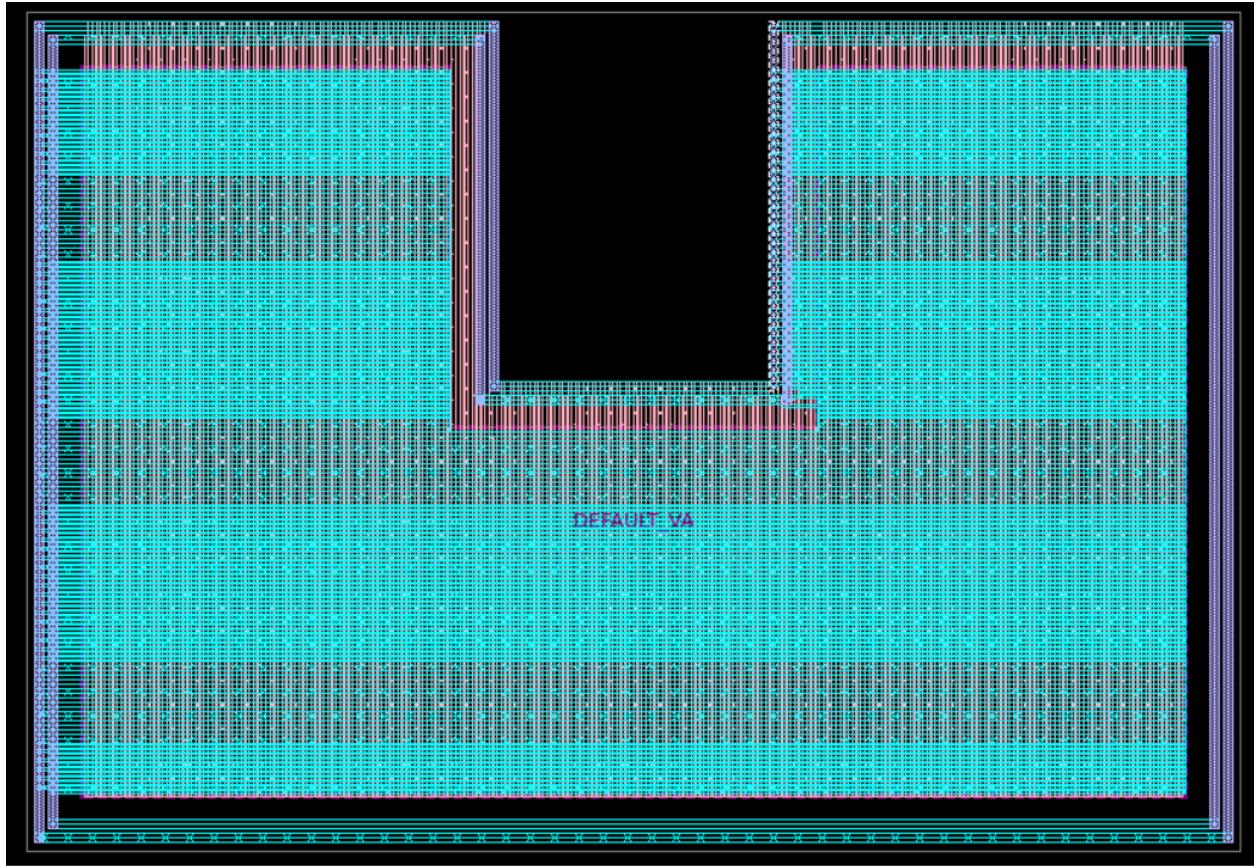
- **Layers Used:**
 - Vertical: M6 (width = 0.77, offset = 0.86)
 - Horizontal: M7 (width = 0.78, offset = 0.96)
- **Spacing:** interleaving
- **Pitch:** Maintained as per foundry rules
- **Extension Direction:** Top and Left only
- **Stop:** At innermost ring
- **Commands:**

```

create_pg_mesh_pattern mesh \
-layers { \
    {{vertical_layer:M6}    {width:0.77}    {spacing:interleaving}    {pitch:1.8} \
{offset:0.86}} \
    {{horizontal_layer:M7}   {width:0.78}    {spacing:interleaving}    {pitch:2} \
{offset:0.96}} \
}

set_pg_strategy core_mesh -pattern {{pattern:mesh} {nets:VDD VSS}} \
-extension {{direction: T L } {stop:innermost_ring}} -core
compile_pg -strategies core_mesh

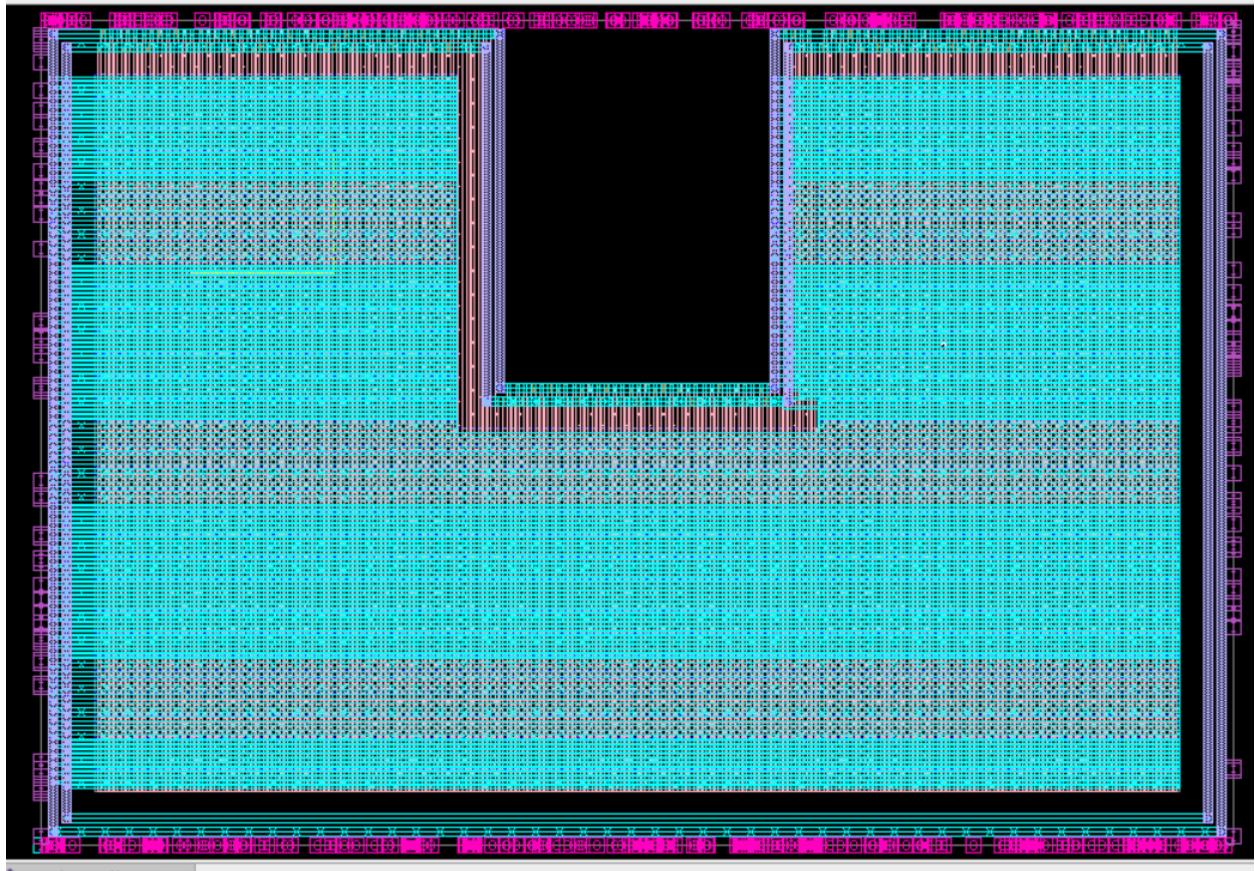
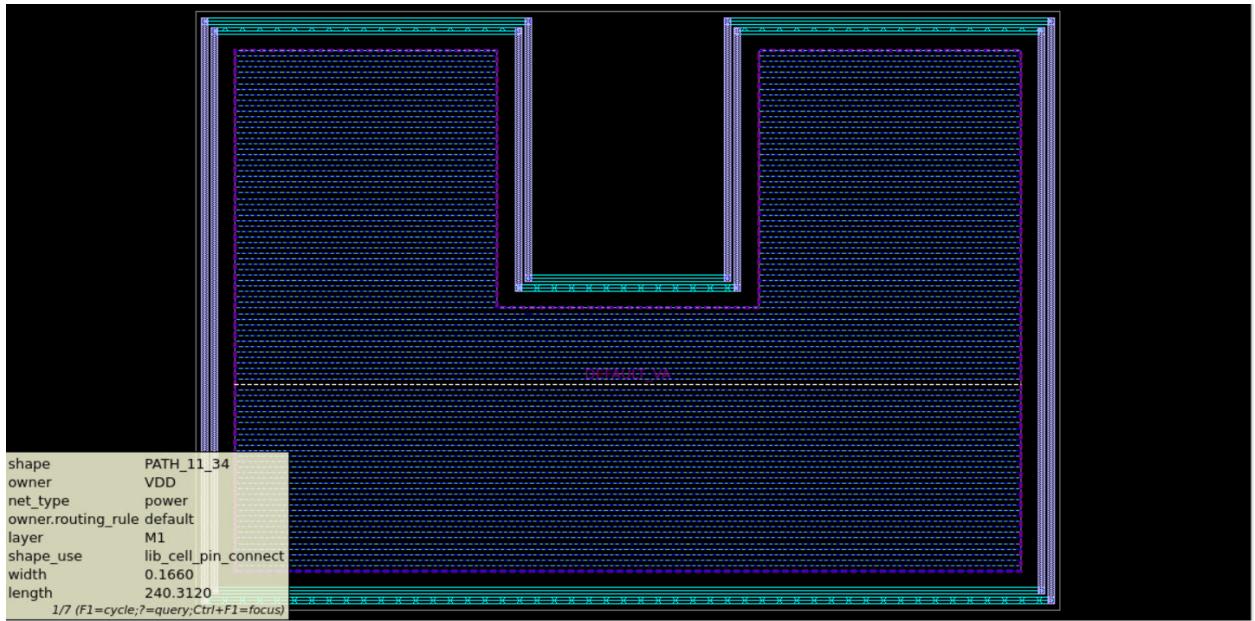
```



Step 7: Standard Cell Power Rails

- **Layer:** M1
- **Rail Width:** 0.166 (slightly more than default width)
- **Commands:**

```
create_pg_std_cell_conn_pattern std_cell_rail -layers {M1} -rail_width 0.166
set_pg_strategy rail_strategy -core \
    -pattern {{name:std_cell_rail} {nets:VDD VSS}}
compile_pg -strategies rail_strategy
```



Save Power Plan

You can optionally save this plan under the name `pp_task1` using: `save block -as pp_task1`

Report:

```
icc2_shell> check_pg_missing_vias
Information: The command 'check_pg_missing_vias' cleared the undo history. (UNDO-016)
Check net VDD vias...
Number of missing vias: 0
Checking net VDD vias took 0 seconds.
Check net VSS vias...
Number of missing vias: 0
Checking net VSS vias took 0 seconds.
Overall runtime: 0 seconds.
icc2_shell> █
```

```
Information: The command 'check_pg_connectivity' cleared the undo history. (UNDO-016)
Checking secondary net through power switch is enabled.
Secondary net will be checked together from primary net. They will be treated as the same net
Primary Net : VDD      Secondary Net:
Primary Net : VSS      Secondary Net:
Loading cell instances...
Number of Standard Cells: 7071
Number of Macro Cells: 0
Number of IO Pad Cells: 0
Number of Blocks: 0
Loading P/G wires and vias...
Number of VDD Wires: 329
Number of VDD Vias: 35193
Number of VDD Terminals: 0
*****Verify net VDD connectivity*****
    Number of floating wires: 0
    Number of floating vias: 0
    Number of floating std cells: 4774
    Number of floating hard macros: 0
    Number of floating I/O pads: 0
    Number of floating terminals: 0
    Number of floating hierarchical blocks: 0
*****
Loading cell instances...
Loading P/G wires and vias...
Number of VSS Wires: 332
Number of VSS Vias: 35617
Number of VSS Terminals: 0
*****Verify net VSS connectivity*****
    Number of floating wires: 0
    Number of floating vias: 0
    Number of floating std cells: 5516
    Number of floating hard macros: 0
    Number of floating I/O pads: 0
    Number of floating terminals: 0
    Number of floating hierarchical blocks: 0
*****
Overall runtime: 0 seconds.
█
```

3.2 Power Plan Task 2 :

Objective:

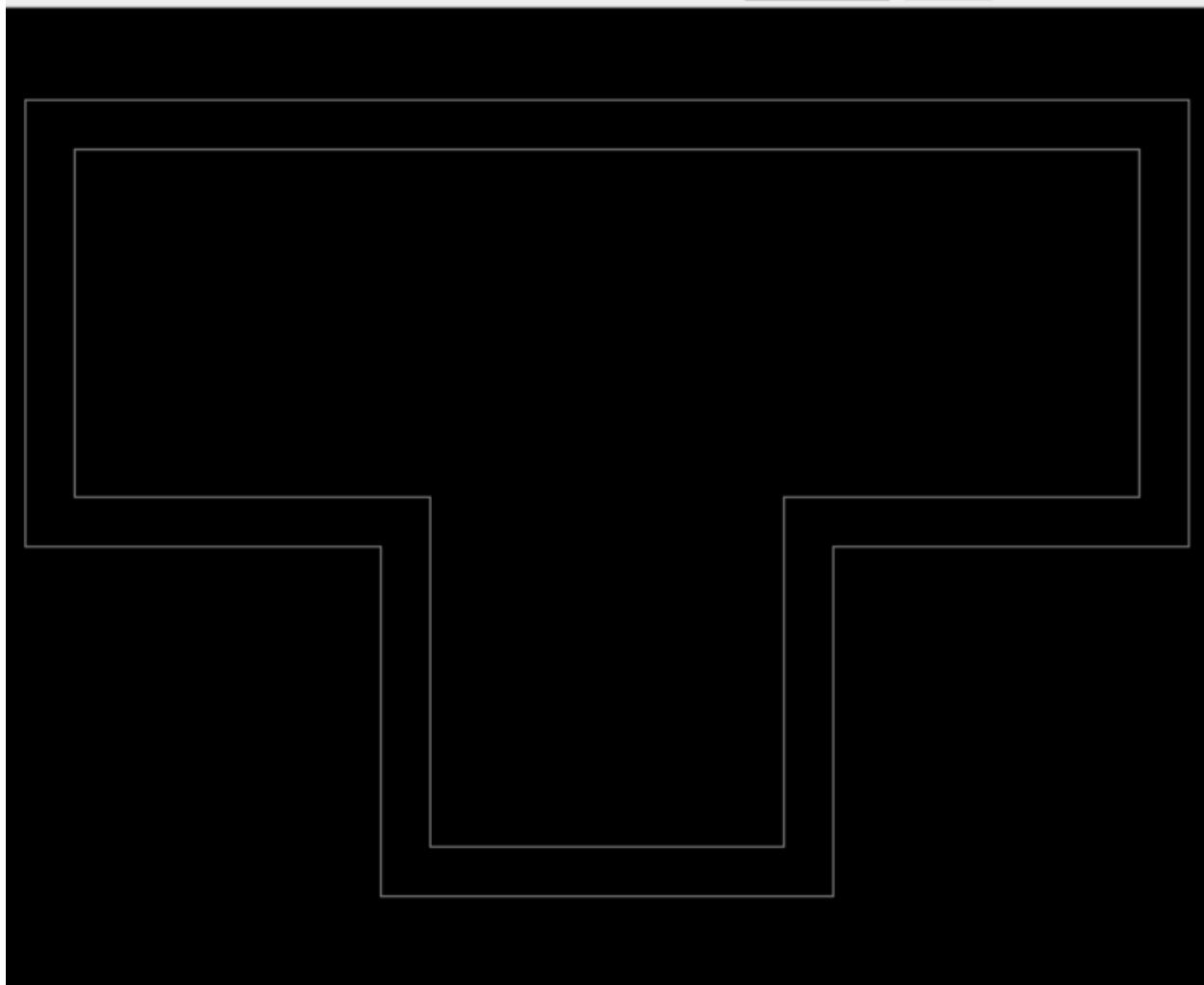
To create a T-shaped power plan with a rectilinear boundary and customized PG (Power Ground) network including ring, mesh, and standard cell rail strategies.

1. Floorplan Initialization

- **Shape:** T-shaped
- **Core Utilization:** 0.80
- **Core Offset:** 12 units
- **Boundary Type:** Rectilinear
- **Command:**

```
initialize_floorplan -shape T -coincident_boundary true -  
core_utilization 0.8 -core_offset 12
```

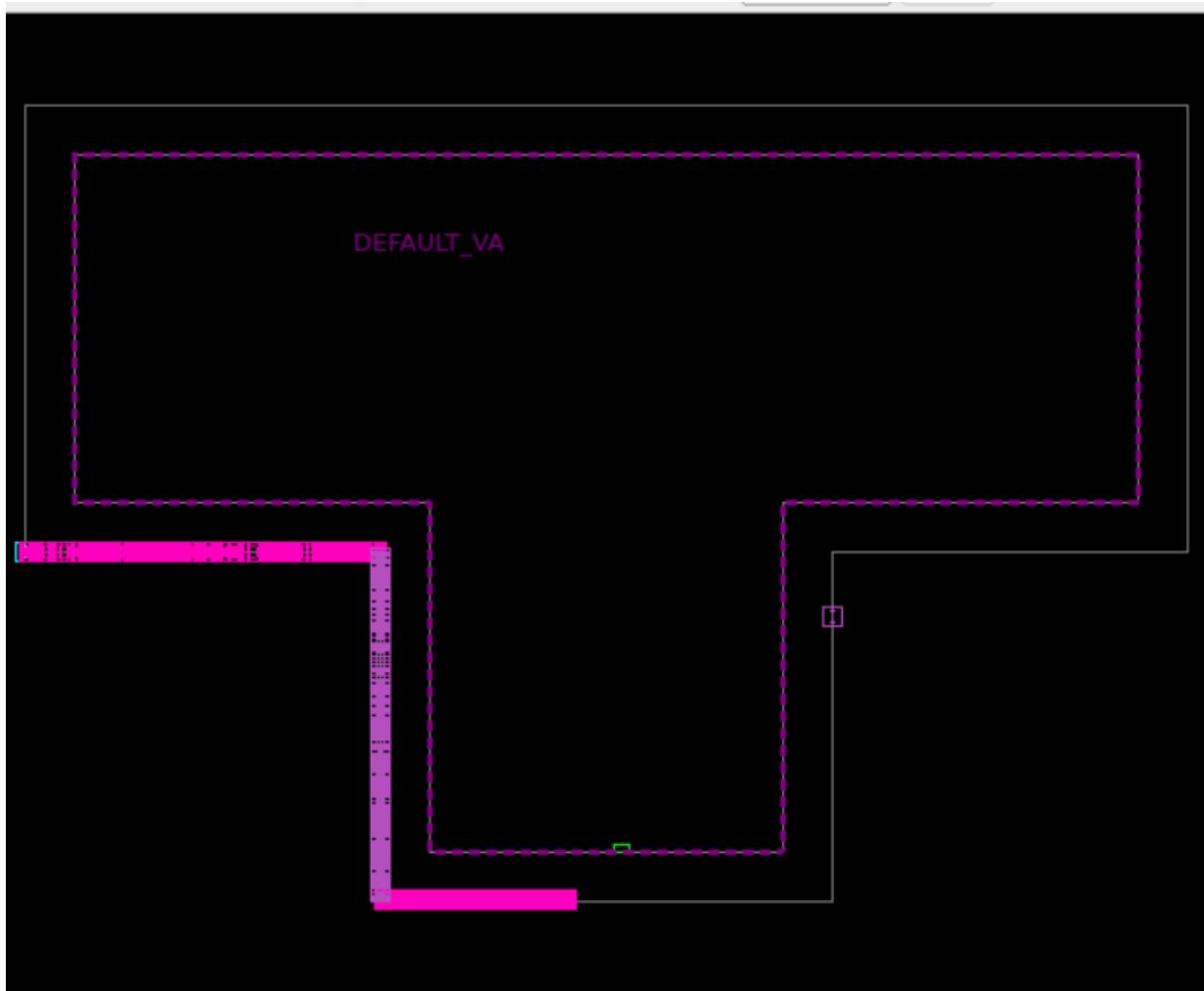
```
*****  
Report : report_utilization  
Design : picorv32  
Version: V-2023.12-SP4  
Date   : Mon Apr 21 12:14:50 2025  
*****  
Utilization Ratio:          0.8159  
Utilization options:  
  - Area calculation based on:      site_row of block PLACEMENT_Coarse  
  - Categories of objects excluded: hard_macros macro_keepouts soft_macros io_cells hard_blockages  
Total Area:                29603.7097  
Total Capacity Area:        29603.7097  
Total Area of cells:        24153.5916  
Area of excluded objects:  
  - hard_macros           : 0.0000  
  - macro_keepouts         : 0.0000  
  - soft_macros            : 0.0000  
  - io_cells               : 0.0000  
  - hard_blockages         : 0.0000  
Total Area of excluded objects: 0.0000  
Ratio of excluded objects:    0.0000  
  
Utilization of site-rows with:  
  - Site 'unit':          0.8159  
  
0.8159  
icc2_shells
```



2. Pin Placement

- **Inputs & Outputs:** Placed on the last three sides (6, 7, 8)
- **Clock Input:** Placed on vertical side number 5 (an odd-numbered side)
- **Commands:**

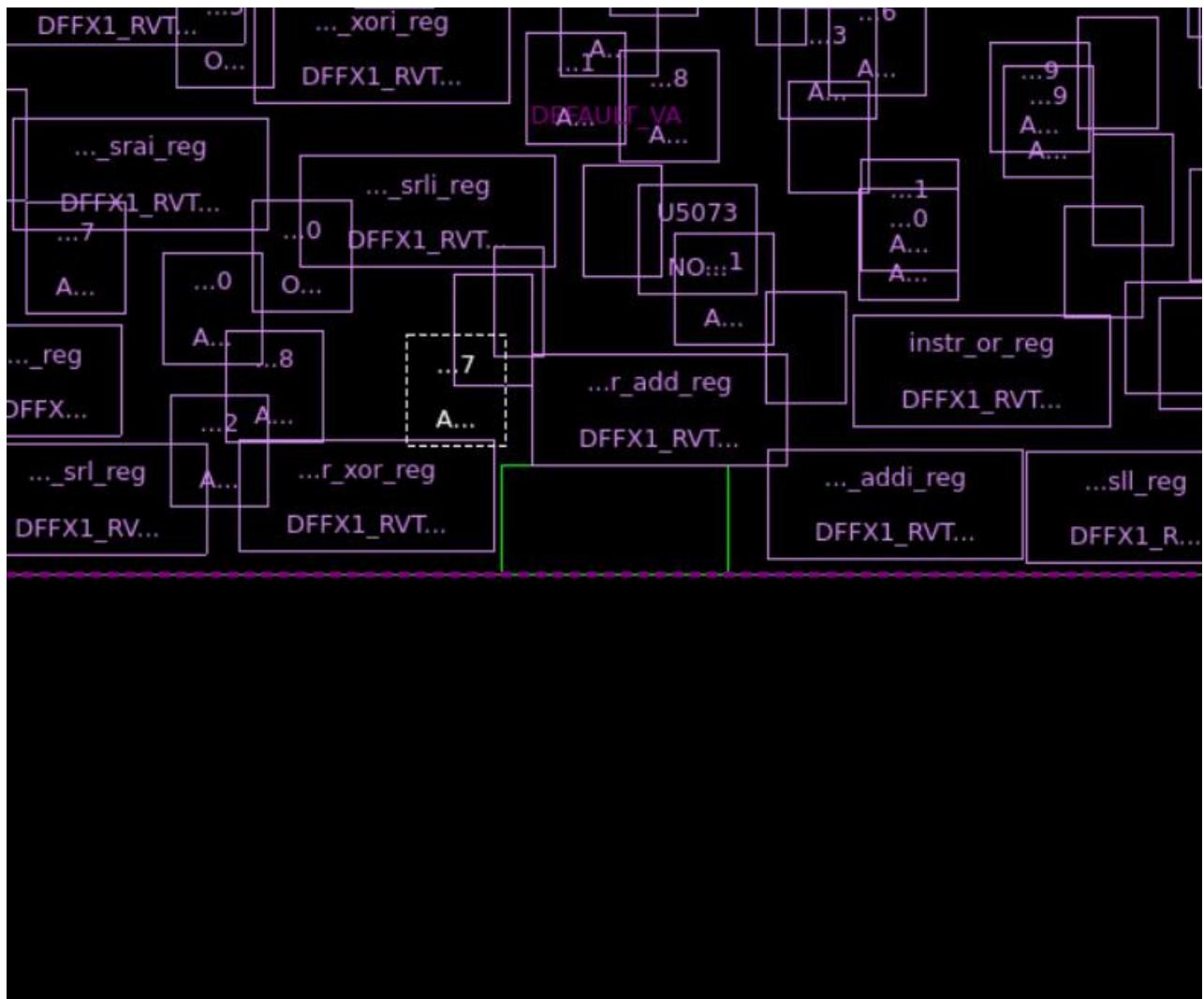
```
set_individual_pin_constraints      -ports      [remove_from_collection
[all_inputs] clk] -sides {6 7 8}
set_individual_pin_constraints -ports [all_outputs] -sides {6 7 8}
set_individual_pin_constraints -ports clk -sides {5}
place_pins -ports [all_outputs]
place_pins -ports [remove_from_collection [all_inputs] clk]
place_pins -ports [get_ports clk]
```



3. Blockage Specification

- **Type:** Soft Blockage
- **Region:** Bottom of the core, coordinates from {144 12} to {147 13}
- **Dimension:** Matches the cell row
- **Command:**

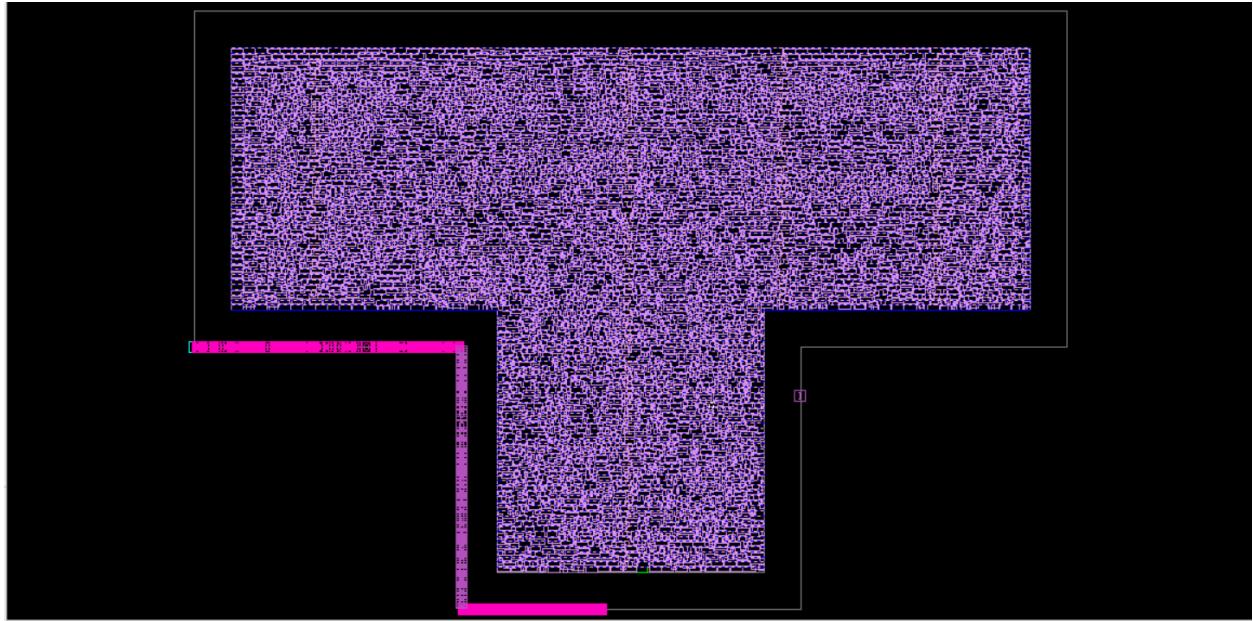
```
create_placement_blockage -type soft -boundary {{144 12} {147 13}} -  
name blockage_soft
```



4. Initial Cell Placement

- Command:

```
create_placement -floorplan
```

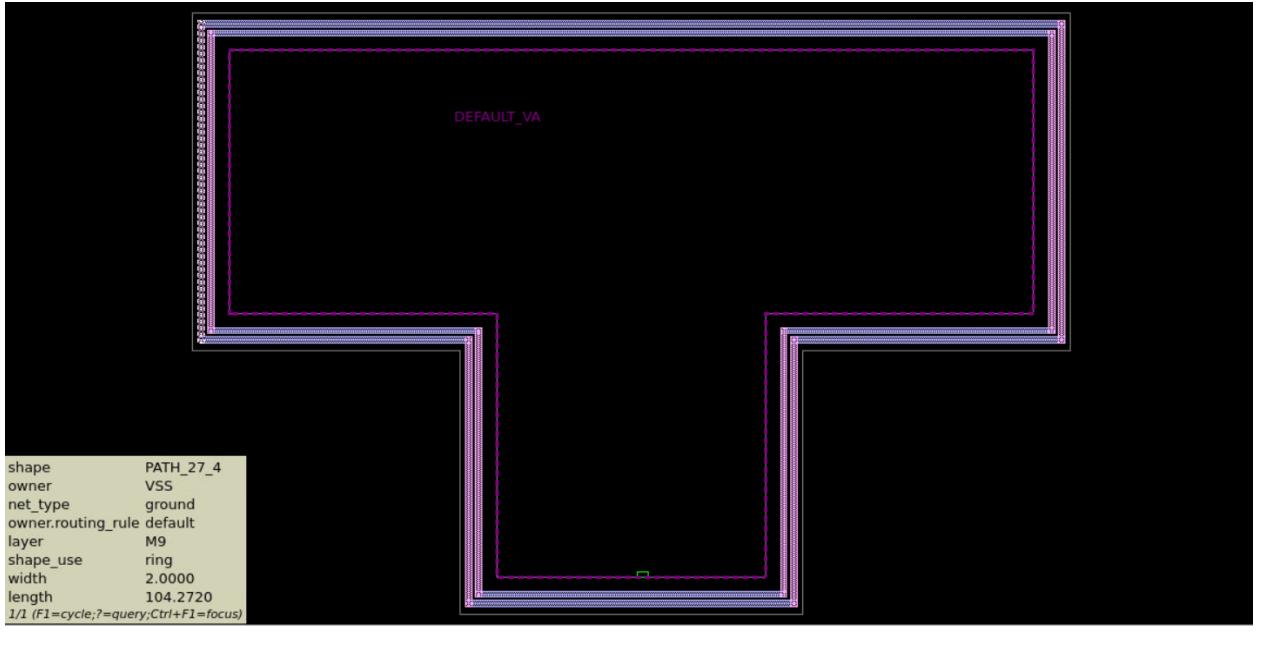


5. Power Ring Generation

- **Layers:**
 - Horizontal: M8 (Width: 2, Spacing: 1.0)
 - Vertical: M9 (Width: 2, Spacing: 1.2)
- **Offset:** {5 4.5} → Satisfying X > 2Y
- **Command:**
- ```
create_pg_ring_pattern core_ring \
 -horizontal_layer M8 -horizontal_width 2 -horizontal_spacing 1. \
 -vertical_layer M9 -vertical_width 2 -vertical_spacing 1.2

set_pg_strategy ring_strategy -core -pattern {{name:core_ring}
{nets:{VDD VSS}} {offset:{5 4.5}}}

compile_pg -strategies ring_strategy
```



## 6. Power Mesh Generation

- **Layers:**
  - Vertical: M7 (Width: 0.78, Spacing: interleaving, Offset: 0.7, Pitch: 2.1)
  - Horizontal: M8 (Width: 0.8, Spacing: interleaving, Offset: 0.8, Pitch: 2.5)
- **Extensions:** Toward **Top** and **Left**, stopping at innermost ring
- **Command:**

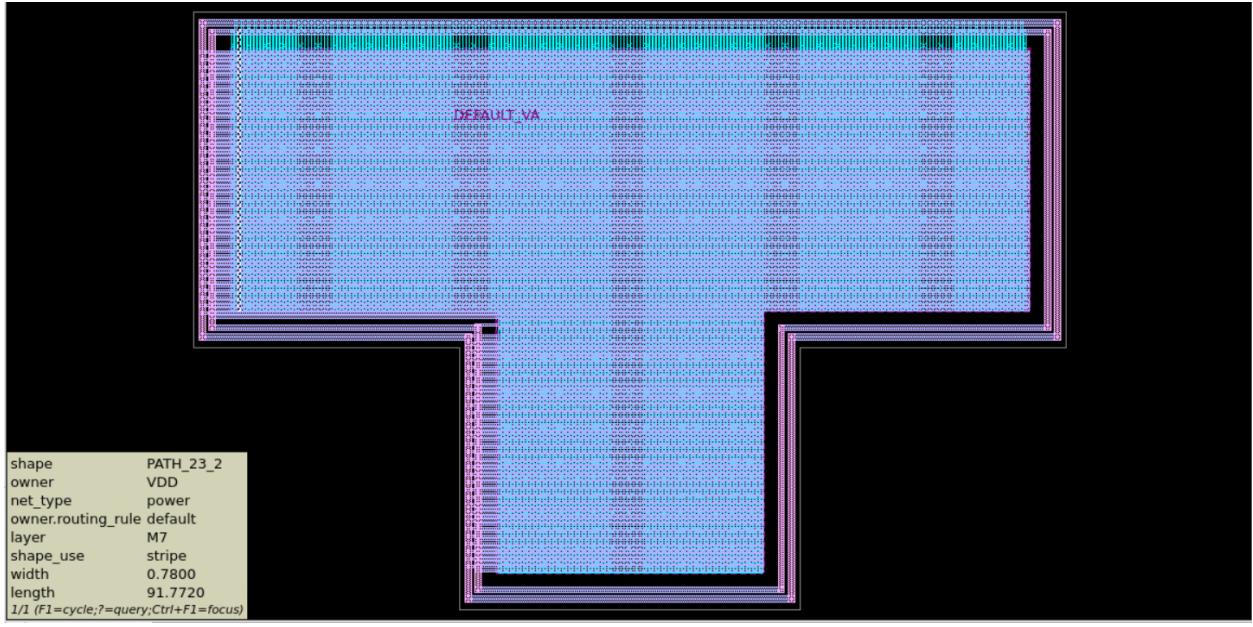
```

create_pg_mesh_pattern mesh -layers {{vertical_layer:M7} {width:0.78}
{spacing:interleaving} {pitch:2.1} {offset:0.7}} {{horizontal_layer:M8}
{width:0.8} {spacing:interleaving} {pitch:2.5} {offset:0.8}}}

set_pg_strategy core_mesh -pattern {{pattern:mesh} {nets:VDD VSS}} -
extension {{direction: T L} {stop:innermost_ring}} -core

compile_pg -strategies core_mesh

```



---

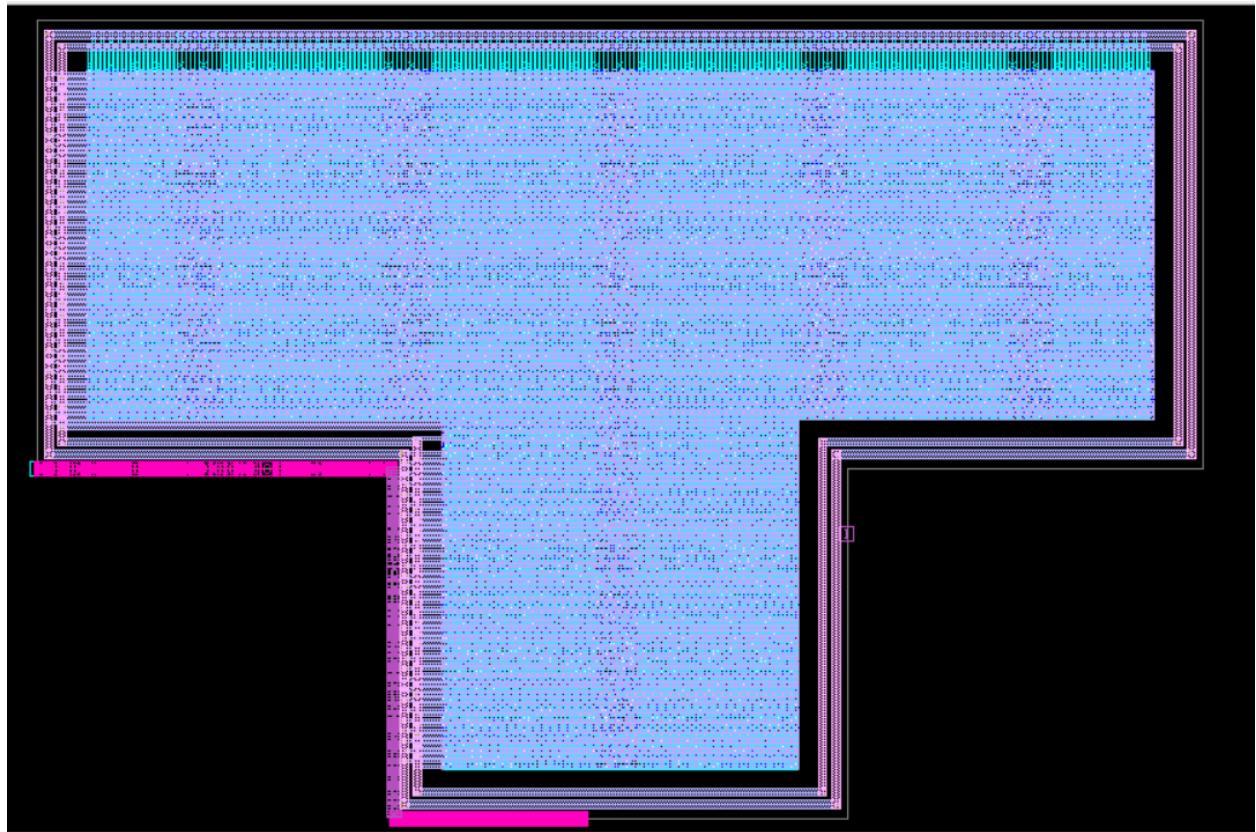
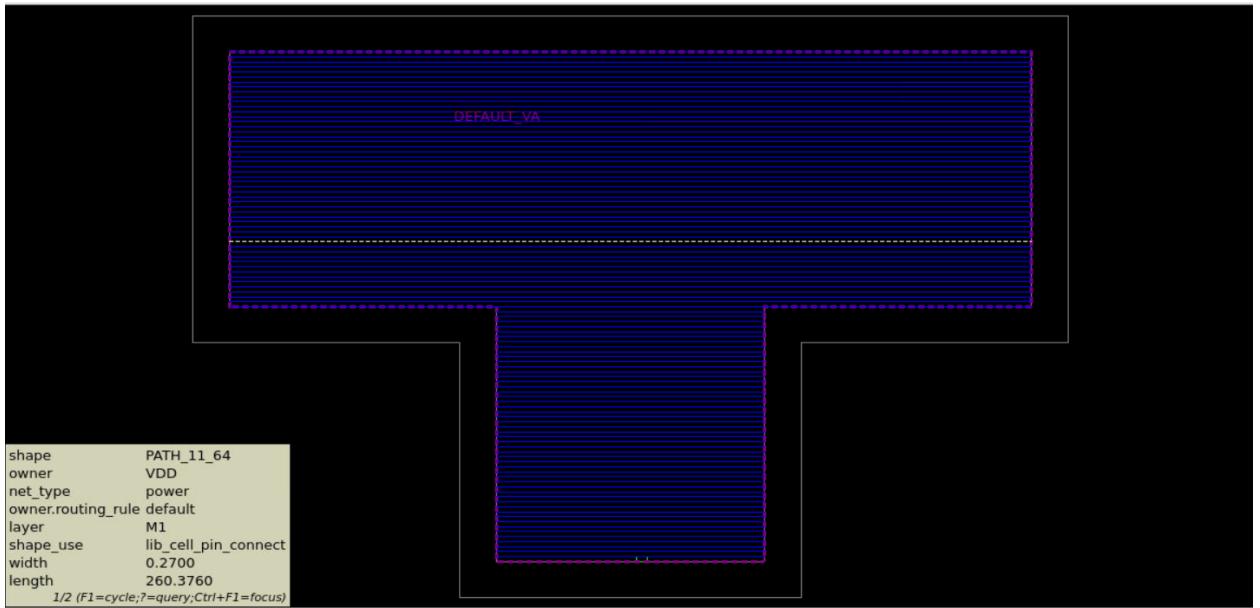
## 7. Standard Cell Power Rail

- **Layer:** M1
- **Rail Width:** 0.27 (slightly less than max width)
- **Command:**

```
create_pg_std_cell_conn_pattern std_cell_rail -layers {M1} -rail_width
0.27
```

```
set_pg_strategy rail_stratery -core -pattern {{name:std_cell_rail}
{nets:VDD VSS}}
```

```
compile_pg -strategies rail_stratery
```



---

## 8. Save Design

- Command : `save_block -as pp_task2`

Reports :

```
Pattern Name: core_ring_pattern
Type: Ring Pattern
Symbolic Nets:
Ring Width:
 {side: horizontal}{width: 2}
 {side: vertical}{width: 3}
Ring Layer:
 {side: horizontal}{layer: M8}
 {side: vertical}{layer: M9}
Ring Spacing:
 {side: horizontal}{spacing: 1.5}
 {side: vertical}{spacing: 1.5}
Corner Bridge: false
Via Rules:
 {intersection: all} {via_master : {default}}
Track Alignment: no alignment
Parameters:
#-----
Pattern Name: mesh
Type: Mesh Pattern
Parameters:
Layers:
{ {vertical_layer : M7} {width : {0.78 }} {spacing : interleaving} {trim : true} {track_alignment : none} {pitch : 2.1} {offset : 0.7} }
{ {horizontal_layer : M8} {width : {0.8 }} {spacing : interleaving} {trim : true} {track_alignment : none} {pitch : 2.5} {offset : 0.8} }
Via Rules:
{intersection: all} {via_master : {default}}
#-----
Pattern Name: mesh_pattern
Type: Mesh Pattern
Parameters:
Layers:
{ {vertical_layer : M8} {width : {minimum }} {spacing : minimum} {trim : true} {track_alignment : none} {pitch : 0} {offset : 0} }
Via Rules:
{intersection: all} {via_master : {default}}
#-----
Pattern Name: std_cell_rail
Type: Standard Cell Rail Pattern
Rail Width: 0.2700 0.2700
```

```
Information: The command 'check_pg_connectivity' cleared the undo history. (UNDO-016)
Checking secondary net through power switch is enabled.
Secondary net will be checked together from primary net. They will be treated as the same net
Primary Net : VDD Secondary Net:
Primary Net : VSS Secondary Net:
Loading cell instances...
Number of Standard Cells: 7071
Number of Macro Cells: 0
Number of IO Pad Cells: 0
Number of Blocks: 0
Loading P/G wires and vias...
Number of VDD Wires: 251
Number of VDD Vias: 31302
Number of VDD Terminals: 0
*****Verify net VDD connectivity*****
 Number of floating wires: 0
 Number of floating vias: 0
 Number of floating std cells: 4609
 Number of floating hard macros: 0
 Number of floating I/O pads: 0
 Number of floating terminals: 0
 Number of floating hierarchical blocks: 0

Loading cell instances...
Loading P/G wires and vias...
Number of VSS Wires: 250
Number of VSS Vias: 30618
Number of VSS Terminals: 0
*****Verify net VSS connectivity*****
 Number of floating wires: 0
 Number of floating vias: 0
 Number of floating std cells: 5164
 Number of floating hard macros: 0
 Number of floating I/O pads: 0
 Number of floating terminals: 0
 Number of floating hierarchical blocks: 0

Overall runtime: 0 seconds.
icc2_shell> █
```

```
icc2_shell> check_pg_missing_vias
Check net VDD vias...
Number of missing vias: 0
Checking net VDD vias took 1 seconds.
Check net VSS vias...
Number of missing vias: 0
Checking net VSS vias took 0 seconds.
Overall runtime: 1 seconds.
icc2_shell> █
```

# Overview of all Tasks

## Power Plan Tasks Summary & Debugging Insights

---

### □ Power Plan Task 1

*Mesh Layer Configuration (Initial) :*

- **Vertical Layer:** M6
  - **Width:** 0.77
  - **Pitch:** 1.8
  - **Offset:** 0.86
- **Horizontal Layer:** M7
  - **Width:** 0.78
  - **Pitch:** 2.0
  - **Offset:** 0.96

*Issues Faced :*

- **VIA6 not dropping** on standard cell rails.
- **2 VSS and 6 VDD** VIAs missing → resulted in **floating nets**.

*Resolution :*

- Initial **rail width** was **0.08**, too narrow for proper via coverage.
- Increased **standard cell rail width to 0.166**, which:
  - Ensured full via drop.
  - Eliminated **floating nets** and **missing VIAs**.
  - Resulted in a **clean, LVS-safe power grid**.

---

### □ Power Plan Task 2

*Mesh Layer Configuration (Initial)*

- **Vertical Layer:** M7
  - **Width:** 0.78
  - **Pitch:** 2.1
  - **Offset:** 0.7
- **Horizontal Layer:** M8

- **Width:** 0.8
- **Pitch:** 2.5
- **Offset:** 0.8

### ***Issues Faced***

- **Standard Cell Rails:**
  - **9 VIAs** were not dropping with initial **rail width = 1.47** (possibly over-constrained).
- **Mesh Grid:**
  - Encountered **7 floating nets** and **12 missing VIAs**.

### ***Resolution***

- Increased **rail width to 0.27**, which:
  - Resolved via drop issues on the rails.
- Adjusted **mesh pitch and width for M7 and M8**, ensuring:
  - **Minimum spacing rules for via insertion** were satisfied.
  - Mesh was **fully connected** with **no floating nets or missing VIAs**.

## **Conclusion**

The two floorplanning and power planning tasks provided an insightful, hands-on experience in physical design flow, testing both conceptual understanding and practical debugging skills.

---

### **□ Floorplan Tasks Summary**

- In **Task 1**, the floorplan was initialized in a **U-shape**, with **core utilization of 0.75** and a **rectangular boundary**. Input and output ports were constrained to specific sides, and a **hard blockage** was introduced on the top side of the core to restrict cell placement.
- In **Task 2**, a **T-shaped floorplan** was implemented with **core utilization of 0.80** and a **rectilinear boundary**. Input/output pins were strategically placed on the last three sides, with the clock input on an odd-numbered vertical side. A **soft blockage** was added to the bottom side to match the cell row dimensions.

These floorplanning tasks emphasized the importance of **core shape selection**, **pin constraint management**, and **blockage definition** for better placement optimization and routing efficiency.

---

## □ Power Plan Tasks Summary

- In **Task 1**, the **ring pattern** was created using **M7 (horizontal)** and **M8 (vertical)** layers. A **mesh** followed using **M6 and M7**, with carefully maintained **interleaving spacing** and pitch values to follow foundry rules. Standard cell rails were generated with **M1 width set to 0.08** initially, but **VIA6 issues** led to increasing the width to **0.166**, resolving **2 VSS and 6 VDD vias** not being dropped.
- In **Task 2**, the **ring pattern** used **M9 (horizontal)** and **M8 (vertical)** layers with a tighter offset strategy ( $X > 2Y$ ). The mesh used **M7 and M8**. Initially, standard cell rails had a width of **0.147**, which caused **9 VIAs** to not drop. Increasing the width to **0.27** fixed this. The mesh also had **7 floating nets and 12 missing VIAs**, which were resolved by adjusting the **width and pitch** of mesh layers.

From **Task 1**, the mesh used:

- **M6 (vertical)**: width = 0.77, offset = 0.86, pitch = 1.8
- **M7 (horizontal)**: width = 0.78, offset = 0.96, pitch = 2

---

## □ Key Takeaways

- **Rail width** must be compatible with **VIA dimensions** to ensure proper connectivity.
- **Mesh layer spacing, width, and pitch** must strictly follow foundry rules to avoid floating nets or missing VIAs.
- Customizing **PG strategies** and extending directions (like `Top` and `Left`) helps manage power connectivity efficiently.
- The alignment between floorplan constraints and power plan components is crucial for a **DRC-clean, LVS-friendly** design.