

# Innovus floorplan

<How is the chip utilization calculated in Genus? Frequently Asked Questions on Utilization in Genus.>

**Chip Level utilization (Core utilization)** : It is the ratio of the area of standard cells, macros and the pad cells with respect to area of chip.

Area (Standard Cells) + Area (Macros) + Area (Pad Cells)

---

Area (chip)

**Floorplan Utilization:** It is defined as the ratio of the area of standard cells, macros, and the pad cells to the area of the chip minus the area of the sub floorplan.

Area (Standard Cells) + Area (Macros) + Area (Pad Cells)

---

Area (Chip) – Area (sub floorplan)

**Standard Cell Utilization:** It is defined as the ratio of the area of the standard Cells to the area of the chip minus the area of the macros and area of blockages.

Area (Standard Cells)

---

Area (Chip) - Area (Macro) – Area (Region Blockages)

**deleteAllFPObjets**

**clear All Floorplan Objects**

Removes all floorplan objects. The density screen, obstruction, power strips, and route guides objects are deleted. The guides and block guides are cleared from the design area. The physical instance groups are also cleared from the core area. Use this command after importing the design.

**deleteSelectedFromFPlan**

Deletes the currently selected floorplan objects. You can use this command at any point in the design flow.

**create\_relative\_floorplan**

**-place obj\_name\_list**

Specifies the **target** object to place. It can be hInst, inst, group, power\_domain, pin\_guide, blockages, or port.

Data\_type: string, optional

**-ref ref\_obj\_name\_list**

Specifies the **reference** object name. It can be hInst, inst, group, power\_domain, pin\_guide, blockages, port, or pin. It cannot be used with -ref\_type core\_boundary or -ref\_type die\_boundary.

Data\_type: string, optional

**-ref\_type {Specifies the type of reference.**

Default: object

Data\_type: enum\_list, optionalcore\_boundary die\_boundary}

-horizontal\_edge\_separate {ref\_edge\_horizontal y\_offset obj\_edge\_horizontal}

Specifies the vertical spacing between the horizontal edge of the target and reference objects.

Data\_type: string, optional

-vertical\_edge\_separate {ref\_edge\_vertical x\_offset obj\_edge\_vertical}

Specifies the horizontal spacing between the vertical edge of the target and reference objects.

Data\_type: string, optional

## delete\_relative\_floorplan

Removes the relative floorplan information from the database. It deletes all relative floorplan constraints, all relative constraints for the specified object(s), or removes the last executed relative floorplan action from the database. Use this command after running create\_relative\_floorplan.

-all

Delete all relative floorplan constraints.

Data\_type: bool, optional

obj\_name\_list

Deletes all relative constraints for the specified object(s).

Note: If you do not specify obj\_name\_list, this command removes the last executed relative floorplan action from the database.

Data\_type: string, optional

## checkFPlan

[-help]

[-outFile fileName]

[-reportUtil]

Checks the quality of the floorplan to detect potential problems before the design is passed on to other tools. This highlights the cells in the design display area with violation markers, where applicable. Use this command after specifying the floorplan data or running an initial floorplan. Run checkFPlan on your final floorplan file.

## placeInstance

Places a leaf instance in the core box. Instance is snapped to the site whose lower left corner is nearest to the location specified. If the location specified is outside of the core box, then the instance is snapped to site whose lower left corner is nearest to the location specified. Instance is not checked for legality—it can overlap other boxes, either placed or fixed, and it can be placed within placement blockages. If no orientation is specified, or if an illegal orientation is specified, placeInstance will orient the instance with the default orientation of the site in which the instance is being placed. You can check the legality of the placed instance by the checkPlace command.

Note: If you place a master instance blackbox to non-R0 orientation, the new non-R0 orientation will be converted to R0 orientation.

placeInstance coreinst/ks\_core1/amba\_dsp1/mcore0/proc0/cmemp0/itags0/u0/id0 {385 937.87} R0 -fixed

placeInstance coreinst/ks\_core1/amba\_dsp1/mcore0/proc0/cmemp0/ddata0/u0/id0 {385 1082.085} R0 -fixed

placeInstance coreinst/ks\_core1/amba\_dsp1/mcore0/proc0/cmemp0/idata0/u0/id0 {385 1226.3} R0 -fixed

placeInstance coreinst/ks\_core1/amba\_dsp1/mcore0/proc0/cmemp0/dtags0/u0/id0 {385 1370.515} R0 -fixed

## dbPlaceInst

instPtr dbLocX dbLocY orient

instPtr

Address of the instance.

dbLocX

X coordinate.

dbLocY

Y coordinate.

orient

Orientation.

e.g.

innovus #> dbPlaceInst \$objPtr 500.0 500.0 R90

dbInstPlacementStatus

Returns the placement status for the specified instance. The possible states are:

instPtr

Address of instance

e.g.

dbInstPlacementStatus [dbGetInstByName DTMF\_INST/i\_10048]

Returns:

dbcUnplaced

dbSetInstPlacementStatus

Sets the placement status for the specified instance. The possible states are:

dbcUnplaced

dbcPlaced

dbcFixed

dbcCovered

instPtr

Address of instance.

status

Status

e.g.

innovus #> dbSetInstPlacementStatus [dbGetInstByName DTMF\_INST/i\_10048] dbcUnplaced

setInstancePlacementStatus

[-help]

{-allHardMacros | -allPtnBlks | -allBlackBoxes | -name instName(s)}

[-status {unplaced | fixed | placed | cover | softFixed}]

Changes the placement attributes status for all hard macros, all partition blocks, all blackboxes, or selected instances. This allows you to fix all blocks so they are not moved by place\_design.

innovus #> dbGet [dbGetInstByName coreinst/ks\_core1/amba\_dsp1/ram2p\_78kx32/ram0/ram0].pStatus placed

innovus #> dbSet [dbGetInstByName coreinst/ks\_core1/amba\_dsp1/ram2p\_78kx32/ram0/ram0].pStatus fixed

innovus #> dbGet [dbGetInstByName coreinst/ks\_core1/amba\_dsp1/ram2p\_78kx32/ram0/ram0].pStatus fixed

```
innovus #> setInstancePlacementStatus -name coreinst/ks_core1/amba_dsp1/ram2p_78kx32/ram0/ram0 -status placed
```

```
innovus #> dbGet [dbGetInstByName coreinst/ks_core1/amba_dsp1/ram2p_78kx32/ram0/ram0].pStatus placed
```

## createPlaceBlockage

Creates cell placement blockages that can be placed. A placement blockage is a floorplan object used to block standard cell placements. Use this command after importing the design.

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 placement, but can be used during in-place optimization, clock tree synthesis, ECO placement or placement legalization (`refinePlace`).
- **Macro-Only** - Enables `planDesign` to keep macros out of the placement blockage; however, it enables standard cells to be placed inside the box if no blockage is present.

## setFinishFPlanMode

### finishFloorplan

```
createPGPin -selected -onDie
```

## addHaloToBlock

## create\_Bump

## power and ground net

```
9 # Version 1.1
10 #
11
12 set init_assign_buffer {1}
13 set init_design_setup 0
14 set init_pwr_net {VDD}
15 set init_gnd_net {VSS}
16 set init_import_mode { -keepEmptyModule 1 -treatUndefinedCellAsBbox 0 -useLefDef56 1}
17 set init_io_file {DATA/asic.entity.io}
18 set init_lef_file {libs/lef/gsclib045.fixed2.lef libs/lef/pdkIO.lef libs/lef/MEM2_4096X32.lef libs/lef/MEM2_2048X32.lef libs/lef/MEM2_1024X32.lef libs/lef/MEM2_512X32.lef libs/lef/MEM2_136X32.lef libs/lef/MEM2_128X32.lef libs/lef/MEM2_128X16.lef libs/lef/MEM1_4096X32.lef libs/lef/MEM1_1024X32.lef libs/lef/MEM1_256X32.lef DATA/leon.partition.lef DATA/periph1.partition.lef}
19 set init_mmmc_file {DATA/viewDefinition.tcl}
20 #set init_top_cell {asic entity}
21 "DATA/asic.entity.globals" 21 lines --95--
```

```
innovus #> dbGet top.pgNets.name
```

VDD VSS

```
innovus #> dbGet top.pgTerms.name
```

0x0

```
innovus #> dbGet top.pgNets.name
```

VDD VSS

```
innovus #> dbGet top.pgNets.isPwrOrGnd
```

1 1

```
innovus #> dbGet top.pgNets.isPwr
```

1 0

```
innovus #> dbGet top.pgNets.isGnd
```

## place hard macrons:

- 1) design with a small number of blocks
  - a. relative floorplaning
  - b. manual movement
- 2) design with a hundred or more blocks
  - a. automatic floorplan synthesis
  - b. prototyping foundation flow

## 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.

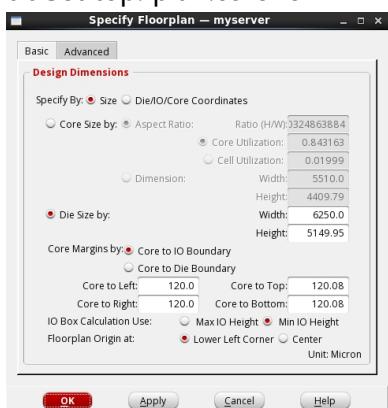
Owing to There are no fixed locations for "soft guide" , you dont need to place it in core area before "Soft Guide" setting.

## query floorplan data

die box: dbGet top.fPlan.box

core box:

dbGet top.fplan.coreBox



innovus #> dbGet top.fplan.box

```
{0.0 0.0 6250.0 5149.95}
```

```
innovus #> dbGet top.fplan.ioBox
```

```
{250.0 250.0 6000.0 4899.95}
```

```
innovus #> dbGet top.fplan.corebox
```

```
{370.0 370.08 5880.0 4779.87}
```

```
selectPhyPin
```

```
deleteSelectedFromFPlan
```

```
innovus #> selectPhyPin 45.0000 0.0000 53.0000 8.0000 8 VDD
```

```
innovus #> deleteSelectedFromFPlan
```

```
innovus #> selectPhyPin 54.2500 0.0000 62.2500 8.0000 8 VSS
```

```
innovus #> deleteSelectedFromFPlan
```

```
10 BUSBITCHARS "[]" ;
11 DESIGN leon ;
12 UNITS DISTANCE MICRONS 2000 ;
13 +-74025 lines: PROPERTYDEFINITIONS-----+
74038 - clk_out_mux MX2X4 + SOURCE TIMING + FIXED ( 1020400 809400 ) FS + WEIGHT 1
74039 ;
74040 - clk_div_out_mux MX2X4 + SOURCE TIMING + FIXED ( 354800 563160 ) FS + WEIGHT 1
74041 ;
74042 END COMPONENTS
74043
74044 PINS 121E ;
74045 - tm + NET tm + DIRECTION INPUT + USE SIGNAL
74046 + LAYER Metal5 ( -70 0 ) ( 70 580 )
74047 + PLACED ( 0 792490 ) E ;
74048 - clk_en + NET clk_en + DIRECTION INPUT + USE SIGNAL
74049 + LAYER Metal2 ( -70 0 ) ( 70 580 )
74050 + PLACED ( 1007800 1410020 ) S ;
74051 +-3621 lines: - resetn + NET resetn + DIRECTION INPUT + USE SIGNAL-----+
77672 - BG_scan_in_17 + NET BG_scan_in_17 + DIRECTION INPUT + USE SIGNAL
77673 + LAYER Metal2 ( -70 0 ) ( 70 580 )
77674 + PLACED ( 938600 0 ) N ;
77675 - BG_scan_out_13 + NET BG_scan_out_13 + DIRECTION OUTPUT + USE SIGNAL
77676 + LAYER Metal5 ( -70 0 ) ( 70 580 )
77677 + PLACED ( 0 675450 ) E ;
77678 END PINS
77679
77680 SPECIALNETS 2 :
77681 - VDD ( * VDD )
demo2.def                                     77681,16  0% demo.def                                     77687,16  0%
```

```
createPGPin
```

```
innovus #> dbGet top.pgTerms.name
```

```
0x0
```

```
innovus #> dbGet top.pgNets.name
```

```
VDD VSS
```

```
innovus #> selectWire 45.0000 13.1650 53.0000 691.8850 8 VDD
```

```
innovus #> editResize -direction y -offset -13.165 -side low -keep_center_line auto
```

```
innovus #> selectWire 45.0000 0.0000 53.0000 691.8850 8 VDD
```

```
innovus #> createPGPin -onDie -selected
```

```
innovus #> selectWire 54.2500 1.9150 62.2500 703.1350 8 VSS
```

```
innovus #> editResize -direction y -offset -1.915 -side low -keep_center_line auto
```

```
innovus #> selectWire 54.2500 0.0000 62.2500 703.1350 8 VSS
```

```
innovus #> createPGPin -onDie -selected
```

```
innovus #> dbGet top.pgTerms.name
```

```
VSS VDD
```

```
innovus #> dbGet top.pgNets.name
```

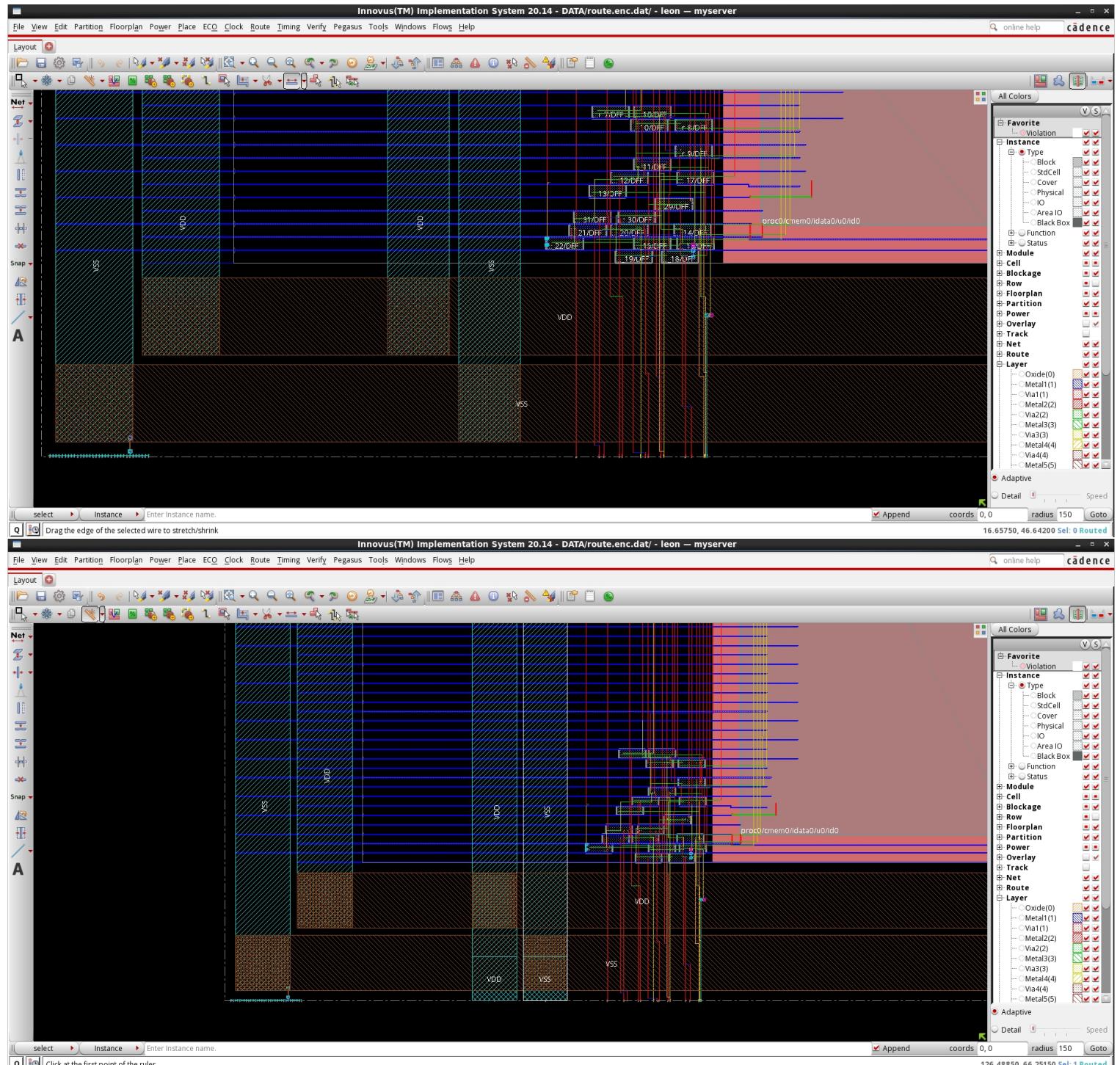
VDD VSS

```
innovus #> dbGet top.pgTerms.pinshapes.rect
```

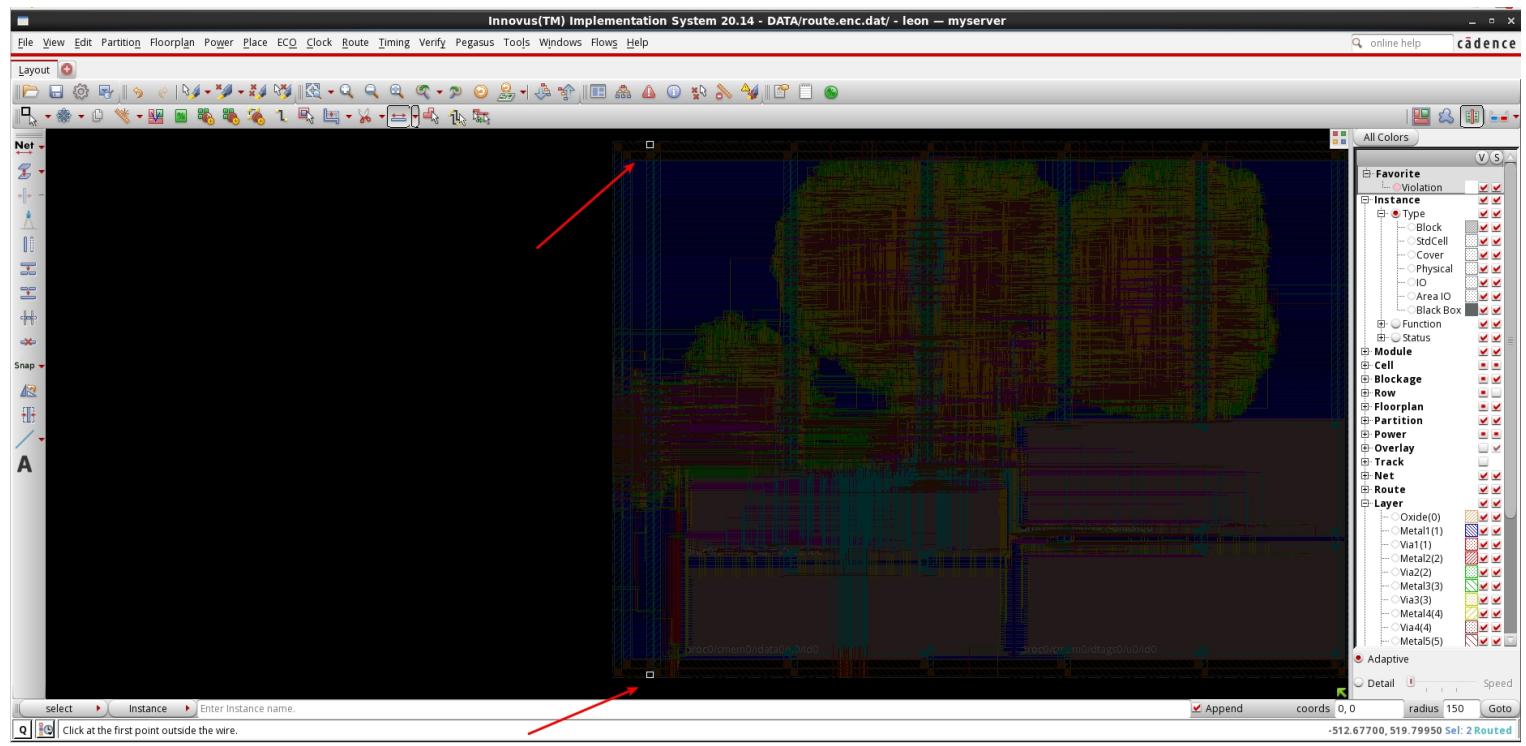
\*\*physical pin-shape \*\*

```
innovus #> dbGet top.pgTerms.pinshapes.layer.name
```

\*\* pin layer\*\*



**createPGPin -onDie -selected** create pin shape where the stripe touch design boundary



Option **{pgPinName [-geom]}** and **{-onDie [-selected] [-width] [-length]}** are mutually exclusive

```
innovus #> selectPGPin -net VDD
innovus #> deletePGPin -selected
```

#### createPhysicalPin

Adds a new physical pin-shape to a top-level term (DEF PIN). You can use this command to add **more PG** (power/ground) pin-shapes to an existing PG term. It will also create a new PG term if there is not already a PG term.

Note: It can add shapes to top-level signal terms, but this is not recommended. Multiple shapes for a signal term may cause flow problems (the router will not connect the different pin shapes, etc.) and should only be done with expert knowledge of the flow implications. It will not create a new signal term (You must use ECO commands to add a new signal term).

**-allowOutsideBoundary**

Do not check whether the specified rectangle is inside the design boundary.

**Data\_type:** bool, optional

**-layer layerName**

Specifies the LEF (or OA) layer name, or the routing layer index value. For example, 1 for first routing layer, 2 for second routing layer.

**Data\_type:** string, optional

**-net netName**

Specifies the name of the net. This is ignored if a top-level term named pinName already exists.

**Data\_type:** string, optional

**pinName**

Specifies the **name** of the **top-level term (DEF PIN)** to add shapes to. If it does not exist, then –net netName is used to find a top-level term instead. If netName does not already have a top-level term, and it is a PG net, then a new term named pinName will be created for netName.

**Data\_type:** string, required

**-polygon {x1 y1 x2 y2 ... xn yn}**

Specifies the coordinates of each vertex of the shape.

**Data\_type:** (point)+, optional

**-rect {x1 y1 x2 y2}**

Specifies the coordinates of the rectangular shape.

x1 Specifies the lower-left x coordinate, in microns, of the physical pin.

y1: Specifies the lower-left y coordinate, in microns, of the physical pin.

x2: Specifies the upper-right x coordinate, in microns, of the physical pin.

y2: Specifies the upper-right y coordinate, in microns, of the physical pin.

**Data\_type:** rect, optional

**-samePort**

Specifies that the physical pin shapes will be added to the first DEF PORT. If this parameter is not specified, each pin shape is created on a separate DEF port. Normally each PG shape should be on a separate port so each one is connected separately to the PG mesh at the next level up.

**Data\_type:** bool, optional

innovus #> createPhysicalPin -layer Metal8 -rect {45.0 0.0 53.0 8.0} VDD

**SAME WITH**

innovus #> selectWire 45.0000 0.0000 53.0000 691.8850 8 VDD

innovus #> createPGPin -onDie -selected

**-samePort demystify**

The screenshot shows two terminal windows side-by-side. Both windows have tabs labeled 'multiport.def : singleport.def' and 'Browse...'. The left window contains the following DEF code:

```
77675 - BG scan out 13 + NET BG scan out 13 + DIRECTION OUTPUT + USE SIGNAL
77676 + LAYER Metal5 ( -70 0 ) ( 70 580 )
77677 + PLACED ( 0 675450 ) E ;
77678 - VDD + NET VDD + SPECIAL + DIRECTION INOUT + USE POWER
+ PORT
77680 + LAYER Metal8 ( -8000 0 ) ( 8000 16000 )
+ FIXED ( 1179590 0 ) N
+ PORT
+ LAYER Metal8 ( -8000 0 ) ( 8000 16000 )
+ FIXED ( 458530 0 ) N
+ PORT
+ LAYER Metal8 ( -8000 0 ) ( 8000 16000 )
+ FIXED ( 98000 0 ) N
;
77689 END PINS
77690
```

The right window contains the following DEF code:

```
77669 - BG scan out 12 + NET BG scan out 12 + DIRECTION OUTPUT + USE SIGNAL
77670 + LAYER Metal2 ( -70 0 ) ( 70 580 )
77671 + PLACED ( 1191800 1410020 ) S ;
77672 - BG scan in 17 + NET BG scan in 17 + DIRECTION INPUT + USE SIGNAL
77673 + LAYER Metal2 ( -70 0 ) ( 70 580 )
77674 + PLACED ( 938600 0 ) N ;
77675 - BG scan out 13 + NET BG scan out 13 + DIRECTION OUTPUT + USE SIGNAL
77676 + LAYER Metal5 ( -70 0 ) ( 70 580 )
77677 + PLACED ( 0 675450 ) E ;
77678 - VDD + NET VDD + SPECIAL + DIRECTION INOUT + USE POWER
77679 + LAYER Metal8 ( -8000 0 ) ( 8000 16000 )
+ LAYER Metal8 ( -729060 0 ) ( -713060 16000 )
+ LAYER Metal8 ( -1089590 0 ) ( -1073590 16000 )
+ FIXED ( 1179590 0 ) N ;
77683 END PINS
77684
```

Arrows from the left window point to specific lines in the right window, indicating how the 'samePort' parameter is implemented.

**deleteFPObject**

**deleteSelectedFromFPlan**

**deleteAllFPObjets**

**editPowerVia**

**Modifies** or **deletes** existing power vias or **adds** new power vias to the design.

**-modify\_vias**

Modifies vias in the design.

**-delete\_vias {0 | 1}**

Deletes power vias from your design.

Default: 0

**-add\_vias {0 | 1}**

When set to 1, adds power vias to the design.

Default: 0

## -between\_selected\_wires {0 | 1}

When set to 1, adds, modifies, or deletes power vias between selected wires only.

You can select the wires interactively in the design display area or by using the [select\\_obj](#) command.

Default: 0

[editSelect](#) is valid too.



How to add power  
vias to specifi...

## -orthogonal\_only {0 | 1}

When set to 1, creates vias only for orthogonal wires and pin intersections.

Default: If you do not specify this parameter or if you specify it with a value of 1, the software creates vias for orthogonal wires and pin intersections.

## -top\_layer layername

Specifies the highest layer to which vias can connect.

Default: The highest metal layer in the design.

## -bottom\_layer layername

Specifies the lowest layer to which vias can connect.

Default: The lowest metal layer in the design.

## -nets net\_list

Specifies the list of nets on which to add, delete, or edit vias.

## -selected\_vias {0 | 1}

When set to 1, modifies or deletes the selected vias only. You can select the wires interactively in the design display area or by using the [select\\_obj](#) command.

## -selected\_wires {0 | 1}

When set to 1, drops vias between selected wires and all same net objects across the selected wires. You can select the wires interactively in the design display area or by using the [select\\_obj](#) command.

## -split\_long\_via {threshold step offset length}

Splits vias longer than the specified length (threshold, in micrometers) into smaller vias with specified center-to-center spacing (step, in micrometers), left/bottom end offset (offset, in micrometers) and vertical/horizontal length (length, in micrometers) values . This is independent of the option [-split\\_via](#).

[Default: If you do not specify this parameter, the software does not split vias.](#)

Note: If the offset is given as negative, the vias will be placed symmetrically on both ends. The default value of offset is -1.

## setAddRingMode

## setViaGenMode

### -optimize\_via\_on\_routing\_track {true | false}

Free up routing tracks by shift or modify the vias minimally in intermediate layers and not applied to layers where the wires or pins exist. This option only supports generated vias and not predefined vias.

The default is false.

-snap\_via\_center\_to\_grid {{ layer1 {none | grid | half\_grid | either } [hard]} layer2 {{none | grid | half\_grid | either } [hard]} ... }

grid

Snaps via to the full grid layer track.

-optimize\_cross\_via {true | false}

Generates vias with the minimum enclosure value on each layer.

-viarule\_preference {default | predefined | generated | list\_of\_via\_rule/cell\_names | file\_name}

Sets the preferred via-rule usage. Use this parameter before running the froute, sroute, or power planning commands.

## setAddStripeMode

-use\_exact\_spacing {true | false}

If set to true, the power stripe breaks using the exact spacing rule from the obstructing blockage, pin, or wire to avoid a design rule violation. If set to false, the power stripe breaks at 1.5 times of the maximum range rule spacing of all layers.

Default:

true: For advance/N7 nodes

false: For non-advance/non-N7 nodes

-ignore\_nondefault\_domains {true | false}

Creates global stripes over domains without breaking. If you do not want the stripes to break at the non-default domains, set this variable to true before issuing the addStripe command.

Default: false

-partial\_set\_thru\_domain {true | false}

Specifies whether to go over the power domain if domain contains the specified net.

Default: false

## setFillerMode

-fitGap {true | false}

Fills a gap between cells by adding a combination of cells, instead of by adding the single largest cell that fits, if doing so avoids leaving an unfilled single-width gap.

Default: true

-vertical\_stack\_max\_length instance\_length

Specifies the instance max length (in micron) aligned in the vertical direction.

Default: 50

Min: 0

Max: 20000

-vertical\_stack\_repair\_cell {fillerCell ...}

Specifies the list of filler cells to fix the horizontal max length violation.

Default: ""

## deleteFiller

Removes physical cell instances, such as **filler cells**, **end-cap cells**, and **well-tap cells**, from standard cell rows. This command can be called if there are filler cells, end-cap cells, or well-tap cells added in the design.

**Physical cells should not be removed with the <Delete> key.** If you delete a cell using the <Delete> key, the cell will become unplaced. The cell will not be displayed in the layout, but it will still be part of the design. You should remove a physical cell using the **deleteFiller** command.

#### -cell fillerCellList

Specifies a cell name or list of cells to delete. Separate cell names with spaces.

Note: Cells specified by this parameter override the list of cells provided by the **-core** parameter of **setFillerMode**.

#### -keepFixed

Controls the command to keep the End-cap and Well-tap cells that are marked **FIXED**. By default, those cells will be deleted if this option is not specified.

Deletes End-cap and Well-tap cells by default, though they are marked as **FIXED**.

Default: **true**

#### -prefix prefix

Specifies the prefix of the cell instances to delete.

For example, to delete all filler cell instances with the prefix FILL, type the following command:

**deleteFiller -prefix FILL**

The software will remove all physical instances whose name starts with FILL, including instances named FILL\_1, FILLTEMP\_1, and so on.

Default: **FILLER**

**deleteFiller only delete filler instances, the added instances by addInst wont be deleted**

#### addFiller

##### -cell filler\_cell\_list

Specifies the list of filler cells to add. Separate filler cell names with spaces and enclose the list with quotation marks ("") or curly braces ({}).

Note: Cells specified by this parameter override the list of cells provided by the **-core** parameter of **setFillerMode**.

##### -markFixed

Places newly added filler cells in **FIXED** status in the Innovus database.

Note: You can specify this parameter with **addFiller** only, and not with **setFillerMode**.

##### -prefix prefixName

Specifies the prefix for the placed instances.

Default: **FILLER**

e.g.

```
innovus #> addInst -cell FILL32 -inst anonFILL
innovus #> selectInst anonFILL
innovus #> dbGet selected.pStatus
      unplaced
innovus #> dbGet selected.isPhysOnly
      0
innovus #> addFiller -cell FILL32 -area {24.80650 27.35350 33.09600 24.61000}
innovus #> dbGet top.insts.name FILLER*
      FILLER_T_1_1
innovus #> selectInst FILLER_T_1_1
innovus #> dbGet selected.pStatus
```

placed

innovus #> dbGet selected.isPhysOnly

1

exportNdr

Replacing a Via in a Design

editChangeVia -from via4new -to via4 -at {2.717700e+02 8.405600e+02}

editChangeVia -from via4 -to via4new -at {2.717700e+02 8.405600e+02}

# techlef file excerpt

VIA **via4** DEFAULT

# (Worst case resistance model for via4 = 6.4 ohm/ct) = 6.4000e+00

RESISTANCE 6.4000e+00 ;

LAYER Metal4 ;

RECT -0.190 -0.140 0.190 0.140 ;

LAYER Via45 ;

RECT -0.130 -0.130 0.130 0.130 ;

LAYER Metal5 ;

RECT -0.190 -0.140 0.190 0.140 ;

END via4

VIA **via4new** DEFAULT

# (Worst case resistance model for via4 = 6.4 ohm/ct) = 6.4000e+00

RESISTANCE 6.4000e+00 ;

LAYER Metal4 ;

RECT -0.140 -0.190 0.140 0.190 ;

LAYER Via45 ;

RECT -0.130 -0.130 0.130 0.130 ;

LAYER Metal5 ;

RECT -0.140 -0.190 0.140 0.190 ;

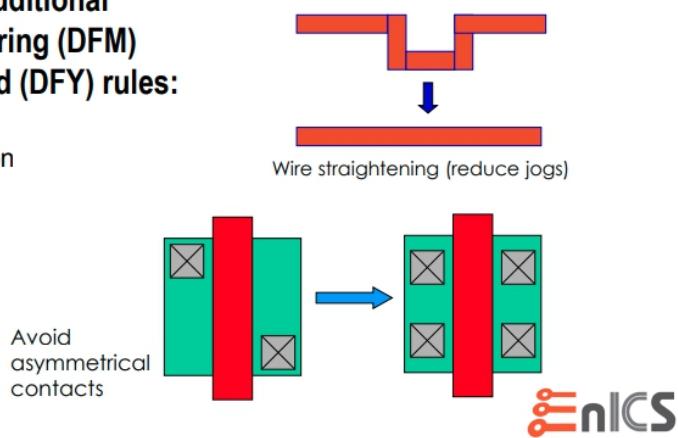
END via4new

setEditMode

jogs

# Design For Manufacturing

- During route, apply additional design for manufacturing (DFM) and/or design for yield (DFY) rules:
  - Via reduction
  - Redundant via insertion
  - Wire straightening
  - Wire spreading



28



Lecture-10-  
Routing.pdf

editPin

```
getPinAssignMode -pinEditInBatch -quiet  
setPinAssignMode -pinEditInBatch true  
editPin -fixOverlap 1 -global_location -side Top -layer 5 -assign 10.712 0.0015 -pin clk  
setPinAssignMode -pinEditInBatch false
```