Innovus Database Navigation

head.

: Root/Head of the database

allCells: List of pointers to cells of all types (**library cells** and **design cells**)

libCells: List of pointers to **library cells**

topCells: List of pointers to **top design cells**

ptnCells: List of pointers to partition cells

vCells: List of pointers to **Verilog cells**

dbGet head.topCells == dbGet top

topCell.

:Top cell, container for flattened connectivity

insts: List of pointers to **instances in the cell**

hInst: Pointer to the top level hierarhical instance

```
innovus #> dbGet head.topCells.
0x7fdb4b8e78c0
innovus #> dbGet top.
0x7fdb4b8e78c0
```

dbGet head.topCells. == dbGet top.

hInst.

: Hierarchical instance (derived from netlist)

allInsts: List of pointers to all **instances** and **hierarchical instances** in the **level** referred to by the hInst

hInsts: List of pointers to all **hierarchical instances** in the **level** referred to by the hInst

insts: List of pointers to all **instances** in the **level** referred to by the hInst

allTreeInsts: List of pointers to all **instances** and **hierarchical instances** in the hInst

treeHInsts: List of pointers to all **hierarchical instances** in the hInst

treeInsts: List of pointers to all **instances** in the hInst

```
innovus #> dbGet top.hInst.treeHInsts.objType -u
    hInst
innovus #> dbGet top.hInst.treeInsts.objType -u
    inst
innovus #> dbGet top.hInst.allTreeInsts.objType -u
    hInst inst
innovus #> expr [llength [dbGet top.hInst.treeHInsts]] + [llength [dbGet top.hInst.treeInsts]]
    6007
innovus #> llength [dbGet top.hInst.allTreeInsts]
    6007
```

cell: Pointer to module cell(vCell) that corresponds to the hInst

parent: Pointer to the parent hInst object from the current hInst
hInstTerms: List of pointers of the hierarchical terminals

> dbGet selected.objType
inst
> dbGet selected.hinst.name
usubB/usuba
> dbGet selected.hinst.hinstTerms.name
usubB/usuba/A **usubB/usuba**/B **usubB/usuba**/Z
> dbGet selected.hinst.hNets.name
usubB/usuba/n1 **usubB/usuba**/n0 **usubB/usuba**/Z **usubB/usuba**/B **usubB/usuba**/A

dbGet top.insts == dbGet top.hinst.treeInsts

> dbGet head.objType
head
> dbGet head.topCells.objType
topCell
> dbGet top.objType
topCell
> dbGet top.hinst.objType
hInst

inst.

: Instance - canonical (flat), equivalent to DEF COMPONENT. Points to a libCell or ptnCell
cell: Pointer to child cell master(libcell) or ptnCell
hInst: Pointer to the **parent hierarhical instance** of the current instance
instTerms: List of pointers to instance terminals
pgInstTerms: List of pgInstTerms for the inst.
pd: Pointer to the power domain that the instance should be placed within (equivalent to Design Browser locPD)

pgInstTerm.

: A power or ground instance terminal that connects to a net.
net: The net connected to this pgInstTerm.
term: The corresponding cell term for this pgInstTerm

```
innovus #> selectInst ahbmo_reg_3_hwdata_18/DFF
innovus #> dbGet selected.objType
    inst
innovus #> dbGet selected.pgInstTerms.name
    VSS VDD
innovus #> dbGet selected.pgInstTerms.net.name
    VSS VDD
```

```
innovus #> dbGet selected.pgInstTerms.term.name
    VSS VDD
innovus #> dbGet selected.instTerms.name
    ahbmo_reg_3_hwdata_18/DFF/CK ahbmo_reg_3_hwdata_18/DFF/D ahbmo_reg_3_hwdata_18/DFF/Q
    ahbmo_reg_3_hwdata_18/DFF/SE ahbmo_reg_3_hwdata_18/DFF/SI
innovus #> dbGet selected.instTerms.cellTerm.name
    CK D Q SE SI
```

net.

*: Canonical (flat) net (equivalent to **connectivity in DEF NETS and SPECIALNETS**)*

hNets: List of pointers to hNets which make up this net.

instTerms: List of pointers to instTerm connections

terms: List of pointers to top level term connections

term.

*: Terminal for **libCell**, **ptnCell**, or **topCell***

cell: Pointer to the cell (libCell, topCell, or ptnCell) that contains the terminal

direction: Specifies the direction of the power or ground terminal from liberty data. PG terminals with no liberty entry will have 'invalid'. Legal enum: bidi, input, internal, output, outputTriState, unknown

net: Pointer to canonical (flat) net connected to the terminal

pinShapes: All the individual physical pin shapes of the term.

pins: List of pointers to **ports (lef port)**

pin

*: **Lef Port***

allShapes: List of pointers to **layerShapes** and **shapeVias** that define the terminal pin geometries.

layerShapeShapes: List of pointers to **layerShapes** that define the terminal pin geometries.

shapeViaShapes: List of pointers to **shapeVias** that define the terminal pin geometries

layerShape

: layerShape

shapeVia

: layer Shape via

Layer Geometries

```
{ LAYER layerName
    [EXCEPTPGNET]
    [SPACING minSpacing | DESIGNRULEWIDTH value] ;
  [WIDTH width ;]
  { PATH [MASK maskNum] pt ... ;
  | PATH [MASK maskNum] ITERATE pt ... stepPattern ;
  | RECT [MASK maskNum] pt pt ;
  | RECT [MASK maskNum] ITERATE pt pt stepPattern ;
  | POLYGON [MASK maskNum] pt pt pt pt ... ;
  | POLYGON [MASK maskNum] ITERATE pt pt pt pt ... stepPattern ;
  } ...
| VIA [MASK viaMaskNum] pt viaName ;
| VIA ITERATE [MASK viaMaskNum] pt viaName stepPattern ;
} ...
```

Used in the macro obstruction (OBS) and pin port (PIN) statements to define layer geometries in the design.

How to get pin shape of cell

```
innovus #> selectPin DTMF_INST/RESULTS_CONV_INST/i_9296/Y
innovus #> dbGet selected.objType
    instTerm
innovus #> dbGet selected.cellTerm.cell.name
INVX1
innovus #> dbGet selected.cellTerm.pins.objType
    pin
innovus #> dbGet selected.cellTerm.pins.allShapes.shapes.rect
    {0.8 2.37 0.835 3.56} {0.835 1.35 1.065 3.56} {1.065 2.37 1.18 3.56}
innovus #> dbGet selected.cellTerm.pins.layerShapeShapes.shapes.rect
    {0.8 2.37 0.835 3.56} {0.835 1.35 1.065 3.56} {1.065 2.37 1.18 3.56}
innovus #> dbGet selected.cellTerm.pins.shapeViaShapes.
    0x0
```

tech lef

```
MACRO INVX1
  CLASS CORE ;
  FOREIGN INVX1 0 0 ;
  ORIGIN 0 0 ;
  SIZE 1.32 BY 5.04 ;
  SYMMETRY X Y ;
  SITE tsm3site ;
  LEQ INVXL ;

  PIN Y
  DIRECTION OUTPUT ;
  ANTENNADIFFAREA 0.7845 ;
  ANTENNAPARTIALMETALAREA 0.6868 LAYER Metal1 ;
  ANTENNAPARTIALMETALSIDEAREA 2.7454 LAYER Metal1 ;
    PORT
     LAYER Metal1 ;
     RECT 1.065 2.37 1.18 3.56 ;
     RECT 0.835 1.35 1.065 3.56 ;
     RECT 0.8 2.37 0.835 3.56 ;
    END
  END Y
```

hInstTerm.

*: **Hierarchical instance** terminal (seen from **level above**)*

hInst: Pointer to the hierarchical instance

hTerm: Pointer to the hierarchical terminal

net: Pointer to the assoicated canonical (flat) net

term: Pointer to terminal

hterm.

*: Hierarchical terminal (seen from **level below**), “” hinst’s term” ”*

name: Terminal name (**local**)

hInst: Pointer to hInst containing the hTerm

downHNet: Pointer to hierarchical net below

upHNet: Pointer to hierarchical net above

term: Pointer to terminal

> select_obj UsubB/Usuba

> dbGet selected.objType

hInst

> dbGet selected.hIn

hInstTerms hInsts

> dbGet selected.hInstTerms.name

UsubB/Usuba/INA0 UsubB/Usuba/INA1 UsubB/Usuba/OUTA

> dbGet selected.hInstTerms.hTerm.name

INA0 INA1 OUTA

instTerm

:Instance terminal (used in flattened connectivity)

cellTerm: Pointer to equivalent cell terminal

inst: Pointer to Instance containing the instTerm

hNet: Pointer to the hierarchical net (hNet) connected to the instance term

net: Pointer to net connected to the instTerm

selectInst UsubB/Usuba/Unand

> dbGet selected.objType

Inst

> dbGet selected.instTerms.name

UsubB/Usuba/Unand/A1 UsubB/Usuba/Unand/A2 UsubB/Usuba/Unand/ZN

> dbGet selected.instTerms.cellTerm.name

A1 A2 ZN

> dbGet selected.cell.terms.name

A1 A2 ZN

> dbGet selected.cell.terms

0x7f3b3fc37908 0x7f3b3fc37940 0x7f3b3fc37978

> dbGet selected.instTerms.cellTerm

0x7f3b3fc37908 0x7f3b3fc37940 0x7f3b3fc37978

dbGet selected.instTerms.cellTerm == dbGet selected.cell.terms

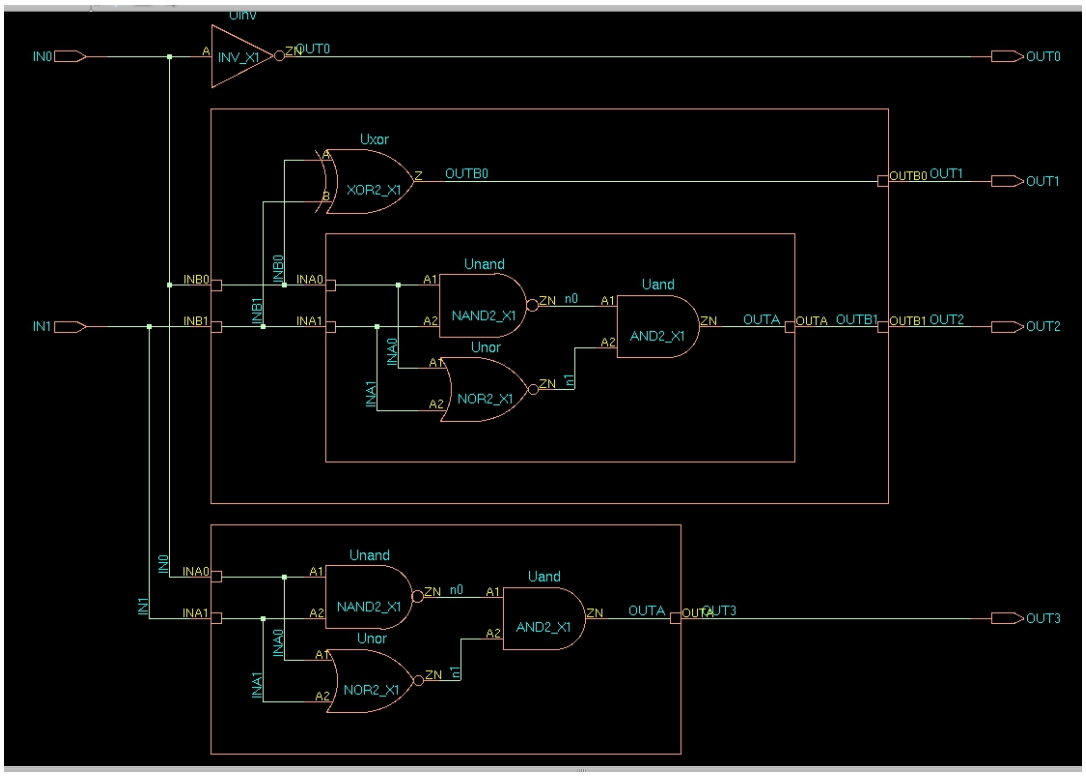
Attribute of libcell..., not design

> select_obj UsubB/Usuba

```
> dbGet selected.objType
    hInst
> dbGet selected.hIn
    hInstTerms  hInsts
> dbGet selected.hInstTerms.name
    UsubB/Usuba/INA0 UsubB/Usuba/INA1 UsubB/Usuba/OUTA
> dbGet selected.hInstTerms.hTerm.name
    INA0 INA1 OUTA

> dbGet [dbGet top.insts.cell.name AND2_X1 -p2].name
UsubB/Usuba/Uand Ua/Uand
> set ptrInsts [dbGet top.insts.cell.name AND2_X1 -p2]
0x7f4a4667a1c0 0x7f4a4667a310

> dbGet [lindex  $ptrInsts 0].cell.Terms
0x7f4a4c2eefa8 0x7f4a4c2eefe0 0x7f4a4c2ef018
> dbGet [lindex  $ptrInsts 1].cell.Terms
0x7f4a4c2eefa8 0x7f4a4c2eefe0 0x7f4a4c2ef018
> dbGet [lindex  $ptrInsts 0].instTerms.cellTerm
0x7f4a4c2eefa8 0x7f4a4c2eefe0 0x7f4a4c2ef018
> dbGet [lindex  $ptrInsts 1].instTerms.cellTerm
0x7f4a4c2eefa8 0x7f4a4c2eefe0 0x7f4a4c2ef018

> dbGet [lindex  $ptrInsts 0].instTerms.
0x7f4a44fd2210 0x7f4a44fd2240 0x7f4a44fd2270
> dbGet [lindex  $ptrInsts 1].instTerms.
0x7f4a44fd23c0 0x7f4a44fd23f0 0x7f4a44fd2420
```

Hterm is different with cellTerm of instTerms, Hterm is still in hierarchy(like, hinst, downHNet attribute).

```
> set ptrHinsts [dbGet top.hInst.treeHInsts.cell.name subA -p2]
0x7f4a46780250 0x7f4a46780378

> dbGet [lindex $ptrHinsts 0].hInstTerms.Hterm
0x7f4a46614290 0x7f4a466142d0 0x7f4a46614310
> dbGet [lindex $ptrHinsts 1].hInstTerms.Hterm
0x7f4a46614350 0x7f4a46614390 0x7f4a466143d0

> dbGet [lindex $ptrHinsts 0].hInstTerms
0x7f4a46614280 0x7f4a466142c0 0x7f4a46614300
> dbGet [lindex $ptrHinsts 1].hInstTerms
0x7f4a46614340 0x7f4a46614380 0x7f4a466143c0
```



get_cells

-filter expr

-hierarchical

If the flat name of an instance at a particular **hierarchical level** matches the patterns, it is included in the result.

leaf instance is included if name matching

e.g.

innovus #> get_cells -hierarchical i_*

DTMF_INST/RAM_128x16_TEST_INST/**i_10074** DTMF_INST/RAM_128x16_TEST_INST/**i_14**

DTMF_INST/RAM_128x16_TEST_INST/i_13 DTMF_INST/RAM_128x16_TEST_INST/i_12

DTMF_INST/RAM_128x16_TEST_INST/i_11 DTMF_INST/RAM_128x16_TEST_INST/i_10

DTMF_INST/RAM_128x16_TEST_INST/i_9 DTMF_INST/RAM_128x16_TEST_INST/i_8

DTMF_INST/RAM_128x16_TEST_INST/i_7 DTMF_INST/RAM_128x16_TEST_INST/i_6

DTMF_INST/RAM_128x16_TEST_INST/i_5 DTMF_INST/RAM_128x16_TEST_INST/i_4

-leaf

Returns leaf cells only.

Note: The -leaf parameter can only be specified with the -of_objects parameter.

-of_objects object_list

Creates a collection of all cells associated with the specified pins, nets, library cells, or design cells.

-regexp

Treats the specified patterns as a regular expression patterns.When this parameter is specified, the given patterns start and end are assumed to be matching with the start and end part of the complete object name.

Leaf instance, follow two command is similar, but “get_cells” return collection, “dbGet” return list

> get_cells -hierarchical -filter "is_hierarchical==false"

UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand UsubB/Uxor Ua/Unand Ua/Unor Ua/Uand Uinv

```
> dbGet top.insts.name
Uinv UsubB/Uxor UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand Ua/Unand Ua/Unor Ua/Uand
```

Hier instance, one return collection, the other one return list of ptr

```
> get_cells * -hierarchical -filter "is_hierarchical==true"
UsubB/Usuba UsubB Ua
> dbGet top.hinst.treeHInsts.name
UsubB UsubB/Usuba Ua

> dbGet top.hInst.Hinsts.name
UsubB Ua
> get_cells * -filter "is_hierarchical==true"
UsubB Ua



> get_property [get_cells UsubB/Usuba] hierarchical_name
UsubB/Usuba
> get_property [get_cells UsubB/Usuba] name
Usuba
> get_property [get_cells UsubB/Usuba] ref_lib_cell_name
subA
> get_property [get_cells UsubB/Usuba] timing_model_type
NA
> get_property [get_cells UsubB/Usuba] is_hierarchical
true
> get_property [get_cells UsubB/Usuba/Uand] is_hierarchical
False
> get_property [get_cells UsubB/Usuba/Uand] instance_model
ECSM



> get_cells UsubB/*
UsubB/Uxor UsubB/Usuba
> dbGet [dbGetHInstByName    UsubB].allInsts.name
UsubB/Uxor UsubB/Usuba

> dbGet [dbGetHInstByName    UsubB].Insts.name
UsubB/Uxor
> get_cells UsubB/* -filter "is_hierarchical==false"
UsubB/Uxor

> dbGet [dbGetHInstByName    UsubB].HInsts.name
UsubB/Usuba
>    get_cells UsubB/* -filter "is_hierarchical==true"
UsubB/Usuba
```

> get_cells **-regexp** UsubB/**.***
UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand UsubB/Uxor UsubB/Usuba
> dbGet [dbGetHInstByName UsubB].allTreeInsts.name
UsubB/Usuba UsubB/Uxor UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand

> dbGet [dbGetHInstByName UsubB].treeInsts.name
UsubB/Uxor UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand
> get_cells -regexp UsubB/.* -filter "is_hierarchical==false"
UsubB/Usuba/Unand UsubB/Usuba/Unor UsubB/Usuba/Uand UsubB/Uxor

> dbGet [dbGetHInstByName UsubB].treeHInsts.name
UsubB/Usuba
> get_cells -regexp UsubB/.* -filter "is_hierarchical==true"
UsubB/Usuba

dbGet top.hInst.insts.name == get_cells * -filter "is_hierarchical==false"

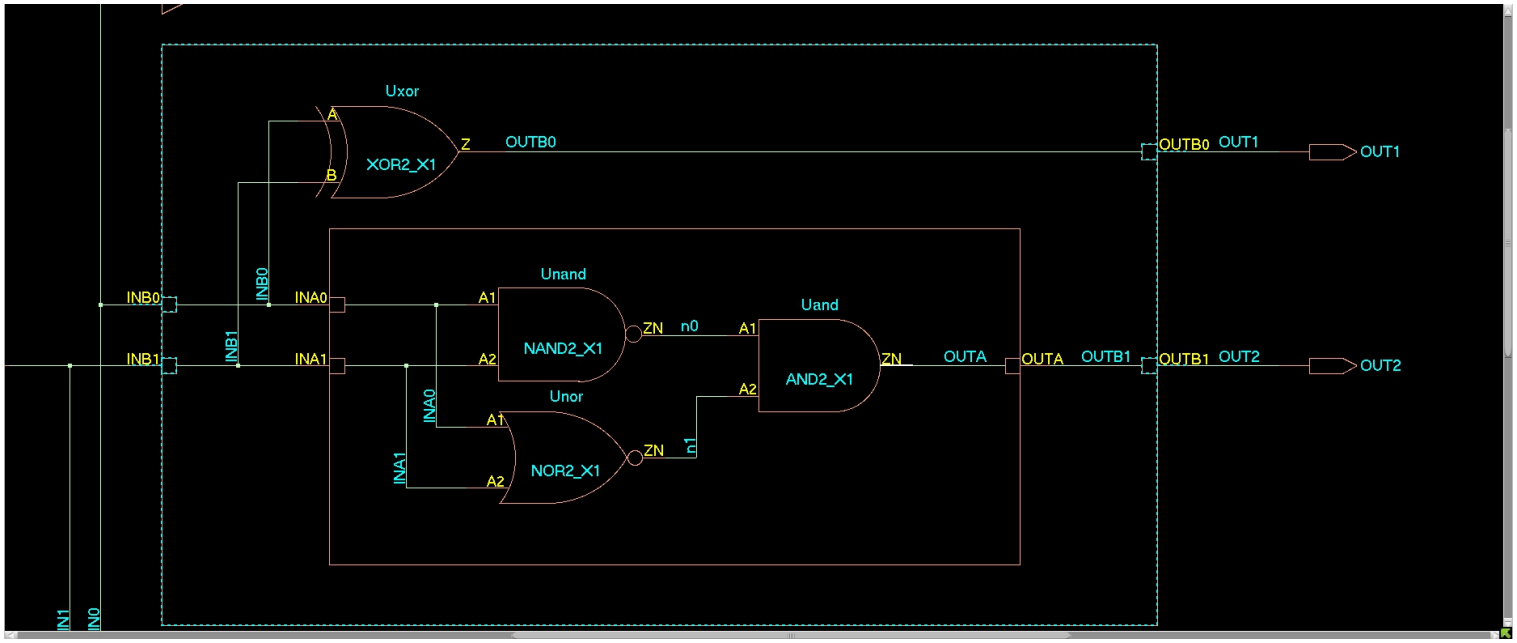
voltus #> dbGet top.hInst.insts.name
test_clk__L1_I0 ref_clk__L1_I0 clk__L2_I0 clk__L1_I0 FE_OFC1_bypass FE_OFC0_bypass shutdown_I0 isolate_I0 u0 g4 g19
voltus #> get_cells * -filter "is_hierarchical==false"
test_clk__L1_I0 ref_clk__L1_I0 clk__L2_I0 clk__L1_I0 FE_OFC1_bypass FE_OFC0_bypass shutdown_I0 isolate_I0 u0 g4 g19

dbGet top.hInst.hinsts.name == get_cells * -filter "is_hierarchical==true"

voltus #> dbGet top.hInst.hinsts.name
AO1 AO2 column external ring RC_CG_DECLONE_HIER_INST RC_CG_DECLONE_HIER_INST80
ISO_RULE_ISO_HIER_INST_0 ISO_RULE_ISO_HIER_INST_1 ISO_RULE_ISO_HIER_INST_2 ISO_RULE_ISO_HIER_INST_3
ISO_RULE_ISO_HIER_INST_4 ISO_RULE_ISO_HIER_INST_5 ISO_RULE_ISO_HIER_INST_6 ISO_RULE_ISO_HIER_INST_7
ISO_RULE_ISO_HIER_INST_8 ISO_RULE_ISO_HIER_INST_9 ISO_RULE_ISO_HIER_INST_10 ISO_RULE_ISO_HIER_INST_11
ISO_RULE_ISO_HIER_INST_12 ISO_RULE_ISO_HIER_INST_13 ISO_RULE_ISO_HIER_INST_14 ISO_RULE_ISO_HIER_INST_15
ISO_RULE_ISO_HIER_INST_16 ISO_RULE_ISO_HIER_INST_17 ISO_RULE_ISO_HIER_INST_18 ISO_RULE_ISO_HIER_INST_19
ISO_RULE_ISO_HIER_INST_20
voltus #> get_cells * -filter "is_hierarchical==true"
AO1 AO2 column external ring RC_CG_DECLONE_HIER_INST RC_CG_DECLONE_HIER_INST80
ISO_RULE_ISO_HIER_INST_0 ISO_RULE_ISO_HIER_INST_1 ISO_RULE_ISO_HIER_INST_2 ISO_RULE_ISO_HIER_INST_3
ISO_RULE_ISO_HIER_INST_4 ISO_RULE_ISO_HIER_INST_5 ISO_RULE_ISO_HIER_INST_6 ISO_RULE_ISO_HIER_INST_7
ISO_RULE_ISO_HIER_INST_8 ISO_RULE_ISO_HIER_INST_9 ISO_RULE_ISO_HIER_INST_10 ISO_RULE_ISO_HIER_INST_11
ISO_RULE_ISO_HIER_INST_12 ISO_RULE_ISO_HIER_INST_13 ISO_RULE_ISO_HIER_INST_14 ISO_RULE_ISO_HIER_INST_15
ISO_RULE_ISO_HIER_INST_16 ISO_RULE_ISO_HIER_INST_17 ISO_RULE_ISO_HIER_INST_18 ISO_RULE_ISO_HIER_INST_19
ISO_RULE_ISO_HIER_INST_20

```
innovus #> set init_pwr_net "VDD"
innovus #> set init_gnd_net "VSS"
innovus #> globalNetConnect VDD -pin VDD -type pgpin -instanceBasename *
innovus #> globalNetConnect VSS -pin VSS -type pgpin -instanceBasename *
innovus #> globalNetConnect VDD -pin avdd! -type pgpin -instanceBasename *
innovus #> globalNetConnect VSS -pin avss! -type pgpin -instanceBasename *
innovus #> selectInst PLLCLK_INST
innovus #> dbGet selected.pgInstTerms.name
    avdd! agnd!
innovus #> dbGet selected.pgInstTerms.net.name
    VDD VSS
innovus #> clearGlobalNets
innovus #> dbGet selected.pgInstTerms.name
    avdd! agnd!
innovus #> dbGet selected.pgInstTerms.net.name
    0x0
```

net vs hNet

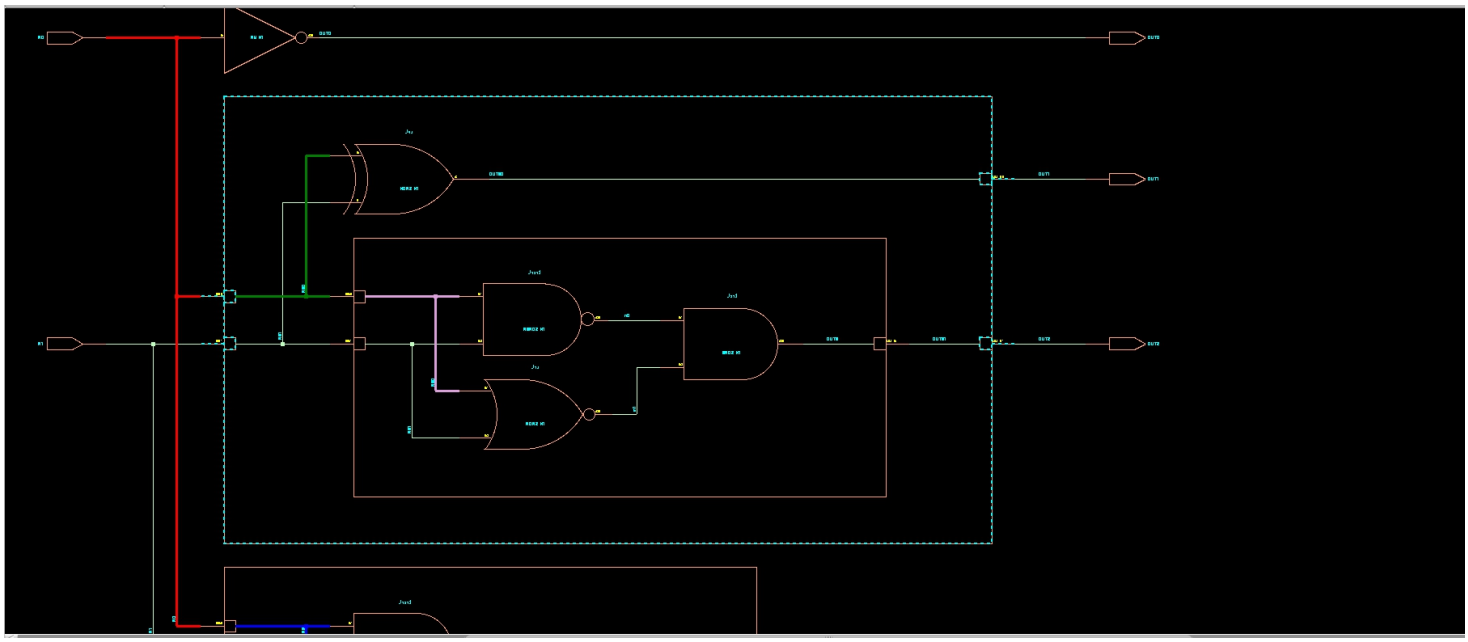



```
    innovus #> selectPin UsubB/Usuba/Uand/ZN
    innovus #> dbGet selected.hNet.name
        UsubB/Usuba/OUTA
    innovus #> dbGet selected.net.name
        OUT2
    innovus #> deselectAll
    innovus #> selectNet OUT2
    innovus #> dbGet selected.objType
        net
    innovus #> dbGet selected.hNets.name
        UsubB/Usuba/OUTA UsubB/OUTB1 OUT2
```

```
innovus #> selectModule UsubB/Usuba
innovus #> dbGet selected.objType
    hInst
```

(net.hNets: List of pointers to hNets which make up this net.)

```
innovus #> dbGetHTermByInstTermName UsubB/Usuba/OUTA
    0x7fc33dff6300
innovus #> dbGet selected.HInstTerms.name
    UsubB/Usuba/INA0 UsubB/Usuba/INA1 UsubB/Usuba/OUTA
innovus #> dbGet selected.HInstTerms
    0x7fc33dff6280 0x7fc33dff62c0 0x7fc33dff6300
innovus #> dbGet [dbGetHTermByInstTermName UsubB/Usuba/OUTA].net.name
    OUT2
innovus #> dbGet [dbGetHTermByInstTermName UsubB/Usuba/OUTA].HTerm.name
    OUTA
innovus #> dbGet [dbGetHTermByInstTermName UsubB/Usuba/OUTA].net.HNets.name
    UsubB/Usuba/OUTA UsubB/OUTB1 OUT2
```



```
innovus #> selectNet IN0
innovus #> dbGet selected.objType
    net
innovus #> dbGet selected.hNets.name
    Ua/INA0 UsubB/Usuba/INA0 UsubB/INB0 IN0
innovus #> llength [dbGet selected.hNets.name]
    4

innovus #> deselectAll
innovus #> dbGet top.insts.objType -u
    inst
innovus #> dbGet top.hInst.treeInsts.objType -u
    inst
```

innovus #> llength [dbGet top.insts. -e]
8
innovus #> llength [dbGet top.hInst.treeInsts. -e]
8

selectInst
leaf instances
dbGetInstByName
innovus #> selectInst UsubB/Usuba/Unor
innovus #> dbGet selected.
0x7fc33e1a6150
innovus #> deselectAll
innovus #> dbGetInstByName UsubB/Usuba/Unor
0x7fc33e1a6150

selectModule
hierarchical instances
dbGetHInstByName
innovus #> dbGet top.hInst.treeHInsts.name ahbsi_out_reg_5_hsize_2 -p
0x7fd657eebfe0
innovus #> dbGetHInstByName ahbsi_out_reg_5_hsize_2
0x7fd657eebfe0

innovus 39> deselectAll
innovus 40> selectObject Module DTMF_INST/TDSP_CORE_INST
innovus 41> dbGet selected.
0x7fb12d468b90
innovus 39> deselectAll
innovus 43> selectModule DTMF_INST/TDSP_CORE_INST
innovus 44> dbGet selected.
0x7fb12d468b90
innovus 46> dbGet top.hinst.treeHInsts.name *TDSP_CORE_INST
DTMF_INST/TDSP_CORE_INST
innovus 47> dbGet top.hinst.treeHInsts.name *TDSP_CORE_INST -p
0x7fb12d468b90

selectNet
dbGetNetByName
innovus #> selectNet UsubB/Usuba/n0
innovus #> dbGet selected.
0x7fc33e10a240
innovus #> dbGetNetByName UsubB/Usuba/n0
0x7fc33e10a240

	db + ptr	set get + name
place instance	dbPlaceInst	placeInstance
get place status	dbInstPlacementStatus	
set place status	dbSetInstPlacementStatus	setInstancePlacementStatus

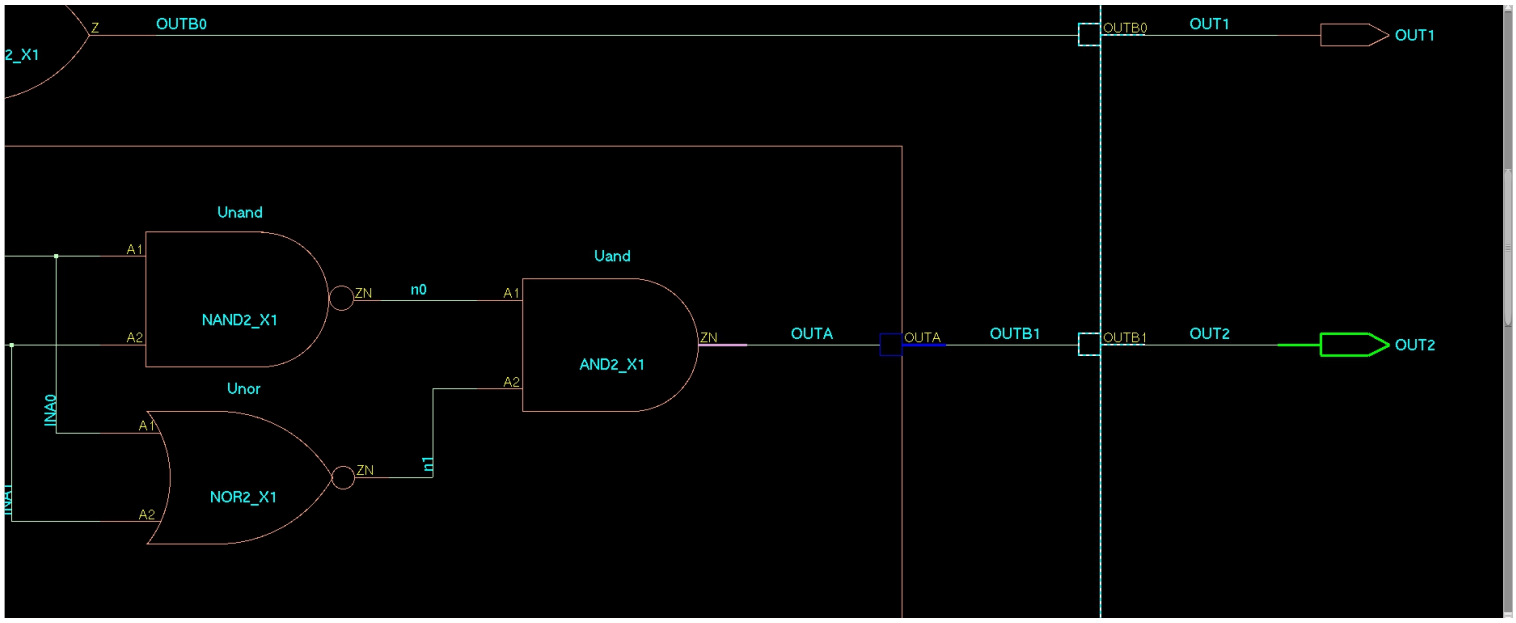
Database Objects	
wire	Wire (symbolic wire type, equivalent to **DEF NETS** wiring). Represents a **wire segment**.
pWire	Wire patch (equivalent to **DEF NETS** wiring **RECT**).
sWire	Special wire (equivalent to **DEF SPECIALNETS** wiring)
vWire	Wire virtual connection (equivalent to **DEF NETS** wiring **VIRTUAL**).
shapeVia	layer Shape via
sViaInst	Special Via (equivalent to DEF SPECIALNETS via instances)
via	Via cell (equivalent to LEF VIA or DEF VIA)
viaInst	viaInst (equivalent to via in DEF NETS wiring)

selectPin

<term|hInstTerm|instTerm>* # Pin name ((term|hInstTerm|instTerm)*, required)

```
innovus #> deselectAll
innovus #> selectPin UsubB/Usuba/Uand/ZN
innovus #> dbGet selected.objType
    instTerm
```

```
innovus #> deselectAll
innovus #> selectPin UsubB/Usuba/OUTA
innovus #> dbGet selected.objType
    hInstTerm
innovus #> deselectAll
innovus #> selectPin OUT2
innovus #> dbGet selected.objType
    term
```



deselectPin

    <term|hInstTerm|instTerm>* # Pin name ((term|hInstTerm|instTerm)*, required)

selectPGPin

selectIOPin

    Selects an I/O pin. You can use this command after importing the design.

deselectIOPin

highlight

    [-help]
    [<any_object>+]
    [-index indexValue | -auto_color | -original]
    [-color colorValue | -auto_color | -original]
    [-pattern patternValue | -original]
    Highlights selected or specified objects with the Highlight Set available through the View - Highlight Selected menu.

select_obj

    [-help]
    <any_object>+

Selects the object or the list of objects for putting into a set. The object type can be any object type selectable on the GUI such as inst and net.

<any_object>+

Specifies the list of objects.

Note: If the list contains the objects that cannot be selected, such as head, layer, and ptn, select_obj will not select them.

The objects that cannot be selected are head, layer, ptn, fPlan, prop, bus, vCell, viaRuleGenerate, layerRule, foreign, hNet, site, densityShape, shape, rackDef,and gCellGridDef.

Data_type: (any_object)+, required

e.g.

```
select_obj [dbGet –p top.insts.name i1]
```

WARN: (IMPDBTCL-200): The command dbSelectObj will be obsoleted in a future release. Please use **select_obj instead.

```
innovus #> deselectAll
innovus #> select_obj DTMF_INST/RAM_256x16_TEST_INST/i_2
innovus #> dbGet selected.
    0x7f20e9bf1ee0
innovus #> dbGet selected.objType
    inst
innovus #> deselectAll
innovus #> select_obj 0x7f20e9bf1ee0
innovus #> dbGet selected.
    0x7f20e9bf1ee0
innovus #> deselectAll
innovus #> selectInst DTMF_INST/RAM_256x16_TEST_INST/i_2
innovus #> dbGet selected.
    0x7f20e9bf1ee0
```

Both **name** and **pointer** are valid arguments

deselect_obj

Deselects the specified object or the list of objects from the Viewer. The object type can be any object type drawn on GUI such as instance wire, via instance.

-all

Deselects all objects.

Data_type: bool, optional

obj_list

Specifies the list of objects to be deselected.

Data_type: (any_object)+, optional

e.g.

```
deselect_obj [dbGet top.insts.name i1 -p]
```

selectNet

{netName | {list_of_netNames} | -allDefClock | -clock | -nonDefaultRule | -shield}

Selects a net and highlights it in the design display window. You can use this command after importing or restoring the design.

It only selects part of net, specifically only the parts which are **regular** wires.

editSelect

Selects **wires** and **vias** using the specified parameters. The wires are updated by subsequent wire editing commands.

-area {x1 y1 x2 y2}

Specifies the coordinates for the area from which to select wires and vias.

Default: Entire area

x1: Specifies the lower left x coordinate.

y1: Specifies the lower left y coordinate.

x2: Specifies the upper right x coordinate.

y2: Specifies the upper right y coordinate.

-direction {H | V | 45 | 135}

Specifies the orientation of the wires to be selected.

Default: If you do not specify this parameter, selects wires with any orientation.

H: Specifies that horizontal wires are to be selected.

V: Specifies that vertical wires are to be selected.

45: Specifies that 45-degree wires are to be selected.

135: Specifies that 135-degree wires are to be selected.

-floating_via

Specifies that only floating vias are to be selected. A via, or a stack of vias, is floating, if it is not connected to another object, wire, pin, or via, on either its top or bottom metal layer or on both these layers.

-from_pin pin_name

Specifies the name of the starting pin of the wire to be selected.

The starting pin you specify should be connected to the same net as the ending pin specified with -to_pin. If they connect to different nets, the tool reports an error. In addition, the from and to pins you specify should be connected to the same wire.

Note: Selection of wires between specified pins is supported only for signal and clock nets.

-to_pin pin_name

Specifies the name of the ending pin of the wire to be selected.

The ending pin you specify should be connected to the **same net** as the starting pin specified with -from_pin. If they connect to different nets, the tool reports an error. In addition, the from and to pins you specify should be connected to the same wire. If there there are more than one wire or via between the specified starting and ending pins, the command is not able to select the wire path and returns an error.

The -from_pin and -to_pin parameters are compatible with only the -direction, -layer, -object_type, and -status options of the editSelect command. If you use -from_pin and -to_pin with any other editSelect parameters, the tool returns an error.

Note: Selection of wires between specified pins is supported only for signal and clock nets.

-layer list_of_layers

Specifies the layers from which to select wires and vias. If you are specifying multiple layers, enclose them in braces.

Default: If you do not specify this parameter, selects wires and vias on all layers.

-net net*

Specifies the net names from which to select wires and vias.

You can use wildcard values in this field. For example, if you specify V*, selects wires/vias from nets VDD and VSS.

-object_type {Wire Via}

Specifies whether wires or vias or both should be selected. If the parameter is not specified, only wires are selected.

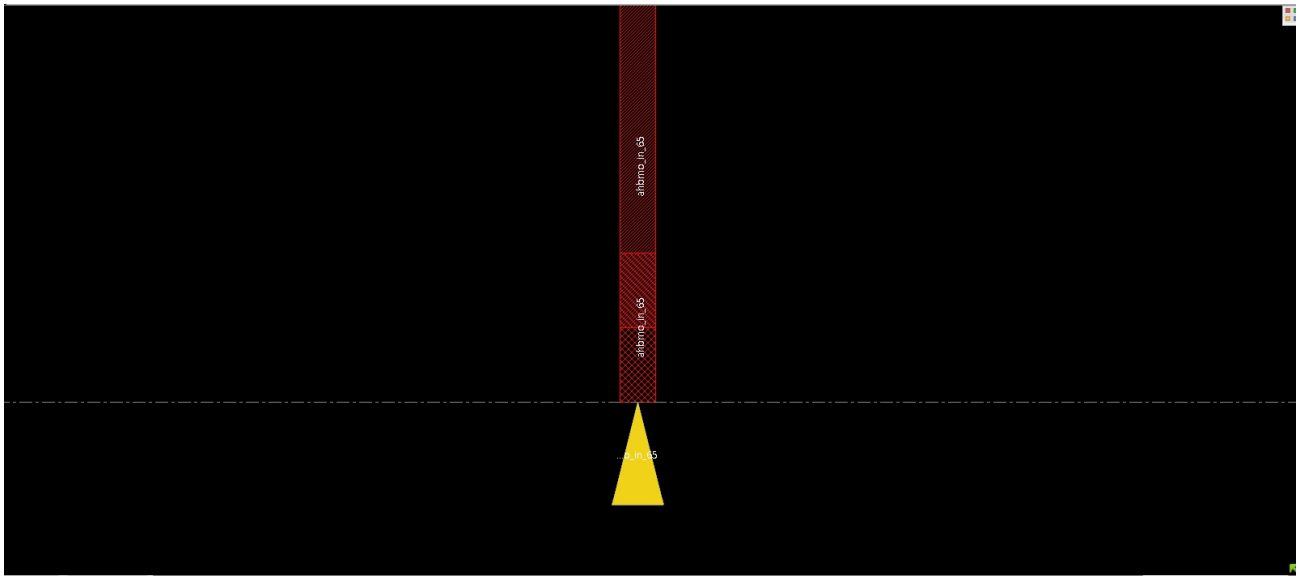
Default: Wire

```
innovus #> editSelect -net ahbmo_3_hwdata[12] -object_type Via
innovus #> dbGet selected.objType -u
    viaInst
```

-physical_pin_only

Specifies that physical pins are to be selected. By default, editSelect does not select physical pins.

You can use the -physical_pin_only parameter along with -area, -layer, or -net to select the physical pins in the specified area, layer, or net, respectively.



```
innovus #> deselectAll
innovus #> editSelect -net ahbmo_in_65 -physical_pin_only
innovus #> dbGet selected.objType
    pinShape
innovus #> deselectAll
innovus #> editSelect -net ahbmo_in_65 -object_type Wire
innovus #> dbGet selected.objType   -u
    wire
innovus #> deselectAll
innovus #> editSelect -net ahbmo_in_65 -object_type Via
innovus #> dbGet selected.objType -u
    viaInst
```

priority Via (-object_type Via) > physical pin (-physical_pin_only) > Wire (-object_type Wire)

```
innovus #> editSelect -net ahbmo_in_65 -object_type Wire -physical_pin_only
innovus #> dbGet selected.objType
    pinShape
innovus #> editSelect -net ahbmo_in_65 -object_type Via -physical_pin_only
```

innovus #> dbGet selected.objType -u
viaInst

-shape {RING STRIPE FOLLOWPIN IOWIRE COREWIRE BLOCKWIRE PADRING BLOCKRING FILLWIRE FILLWIREOPC DRCFILL None}

Specifies the shapes of the wires and vias to be selected. Use all upper-case letters for the shapes. If you specify multiple shapes, enclose the list of shapes in braces. For shape definitions, see Shapes, in the "DEF Syntax" chapter of the LEF/DEF Language Reference.

Default: If you do not specify this parameter, selects wires and vias of all shapes.

-status {COVER FIXED NOSHIELD ROUTED SHIELD UNKNOWN}

Specifies the status of wires and vias to be selected. Use all upper-case letters for the status. If you specify more than one status, enclose the status list in braces.

Note: If you select -type Special -status NOSHIELD or -type Signal -status SHIELD the software generates an error message.

-type {Regular Special Patch}

Specifies whether to select regular wires and vias, special wires and vias, or patch wires.

Note: If you specify Regular, you must omit the -shapes parameter or specify -shapes None.

-use {CLOCK POWER SIGNAL}

Specifies whether to select only CLOCK wires/vias, POWER wires, or SIGNAL wires.

-via_cell cell_name_list

Specifies the via cells of the vias to be selected. You can use wildcards while specifying via cell names. If you are specifying multiple cell names, enclose them in braces.

What are regular nets and spec...

viaInst

viaInst (equivalent to via in **DEF NETS** wiring)

.botRects

List of rectangles (typically only one) on bottom routing layer in terms of design coordinates (equivalent attribute on the via master is in coordinates local to the via master)

.cutRects

List of rectangles on cut layer in terms of design coordinates (equivalent attribute on the via master is in coordinates local to the via master)

.topRects

List of rectangles (typically only one) on top routing layer in terms of design coordinates (equivalent attribute on the via master is in coordinates local to the via master)

.net

Pointer to net that the via belongs to

.via

Pointer to via cell

.rule

Pointer to non-default rule corresponding to the via, vias with the default routing rule will return NULL (0x0).

```
innovus #> dbGet head.vias.name
    VIA12_1C VIA12_1C_H VIA12_1C_V VIA12_2C_E VIA12_2C_W VIA12_2C_N VIA12_2C_S VIA12_2C_HN
    VIA12_2C_HS VIA12_2C_ES VIA12_2C_WS VIA12_4C VIA23_1C VIA23_1C_V VIA23_1C_H VIA23_1ST_N
    VIA23_1ST_S VIA23_2C_E VIA23_2C_W VIA23_2C_N VIA23_2C_S VIA23_4C VIA34_1C VIA34_1C_H VIA34_1C_V
    VIA34_1ST_E VIA34_1ST_W VIA34_2C_E VIA34_2C_W VIA34_2C_N VIA34_2C_S VIA34_4C VIA45_1C VIA45_1C_H
    VIA45_1C_V VIA45_1ST_N VIA45_1ST_S VIA45_2C_E VIA45_2C_W VIA45_2C_N VIA45_2C_S VIA45_2ST_N
    VIA45_2ST_S VIA45_4C VIA5_0_VH VIA5_0_X2H_VH VIA5_0_X2E_VH VIA5_0_X2W_VH VIA5_0_X2V_VH
    VIA5_0_X2N_VH VIA5_0_X2S_VH VIA5_0_X4H_VH VIA5_0_X4E_VH VIA5_0_X4W_VH VIA5_0_X4V_VH
    VIA5_0_X4N_VH VIA5_0_X4S_VH VIA6_0_HV VIA6_0_X2H_HV VIA6_0_X2E_HV VIA6_0_X2W_HV VIA6_0_X2V_HV
    VIA6_0_X2N_HV VIA6_0_X2S_HV VIA7_0_VH VIA7_0_X2H_VH VIA7_0_X2E_VH VIA7_0_X2W_VH VIA7_0_X2V_VH
    VIA7_0_X2N_VH VIA7_0_X2S_VH VIA8_0_VH VIA8_0_X2H_VH VIA8_0_X2E_VH VIA8_0_X2W_VH VIA8_0_X2V_VH
    VIA8_0_X2N_VH VIA8_0_X2S_VH M9_M8_1 M9_M8_2 M8_M7_1 M7_M6_1 M6_M5_1 M5_M4_1 M4_M3_1
    M3_M2_1 M2_M1_1 M3_M2_2 M2_M1_2 M3_M2_3 M4_M3_2 M5_M4_2 M6_M5_2 M7_M6_2 M8_M7_2
    M2_M1_3 M2_M1_4 M2_M1_5 M3_M2_4 M4_M3_3 M5_M4_3 M6_M5_3 M7_M6_3 M8_M7_3 M8_M7_4
    M7_M6_4 M6_M5_4 M5_M4_4 M4_M3_4 M3_M2_5 M2_M1_6 M8_M7_5 M7_M6_5 M6_M5_5 M5_M4_5
    M4_M3_5 M3_M2_6 M2_M1_7 M3_M2_7 M2_M1_8 M3_M2_8 M9_M8_3 M4_M3_6 M5_M4_6 M6_M5_6
    M7_M6_6 M8_M7_6 M9_M8_4 M2_M1_9
innovus #> dbGet head.vias.objType -u
    via
```

via.

*:Via cell (equivalent to **LEF VIA or DEF VIA**)*

isDefault: Indicates that the via is a default via (LEF **VIA DEFAULT**)

isNonDefault: Indicates that the via is declared in a **LEF NONDEFAULTRULE**

viaRuleGenerate: The viaRuleGenerate for this via. It is only set for generated vias created from viaRuleGenerate parameters. See the LEF/DEF manual VIA statement with VIARULE for more details. It returns **0x0** for fixed vias (e.g. a LEF/DEF **VIA with only RECT values**).

```
innovus #> editSelect -net ahbmo_in_65 -object_type Via
innovus #> dbGet selected.objType -u
    viaInst
innovus #> dbGet selected.via.objType -u
    via
innovus #> dbGet selected.botRects
    {{68.835 27.42 68.965 27.49}} {{68.865 27.01 68.935 27.14}} {{69.465 27.01 69.535 27.14}}
innovus #> dbGet selected.via.botRects
    {{-0.065 -0.035 0.065 0.035}} {{-0.035 -0.065 0.035 0.065}} {{-0.035 -0.065 0.035 0.065}}
innovus #> expr (69.535 - 69.465) - (0.035 - (-0.035))
    -6.82787160144e-15
```

redirect

The dbGet and get_property commands return their values as **TCL_RESULTS**. To redirect the output to any file, you can use any of the following methods:

1) You can redirect **output** of dbGet and get_property using the following commands:

redirect {puts "[dbGet top.insts.name]"} > inst_name.rpt

redirect {puts "[get_property [get_lib_cells F1] area]"} > property.rpt

2) You can get the value of output stored in a **variable** using the following commands:

redirect a {puts "[dbGet top.insts.name]"} -**variable**

redirect b {puts "[get_property [get_lib_cells F1] area]"} -**variable**

innovus #> dbGet head.layers.name

Metal1 Metal2 Metal3 Metal4 Metal5 Metal6 Via12 Via23 Via34 Via45 Via56 OVERLAP

innovus #> dbGet head.layers.width

0.23 0.28 0.28 0.28 0.28 0.44 0.0 0.0 0.0 0.0 0.0 0.0

innovus #> dbGet head.vias.name

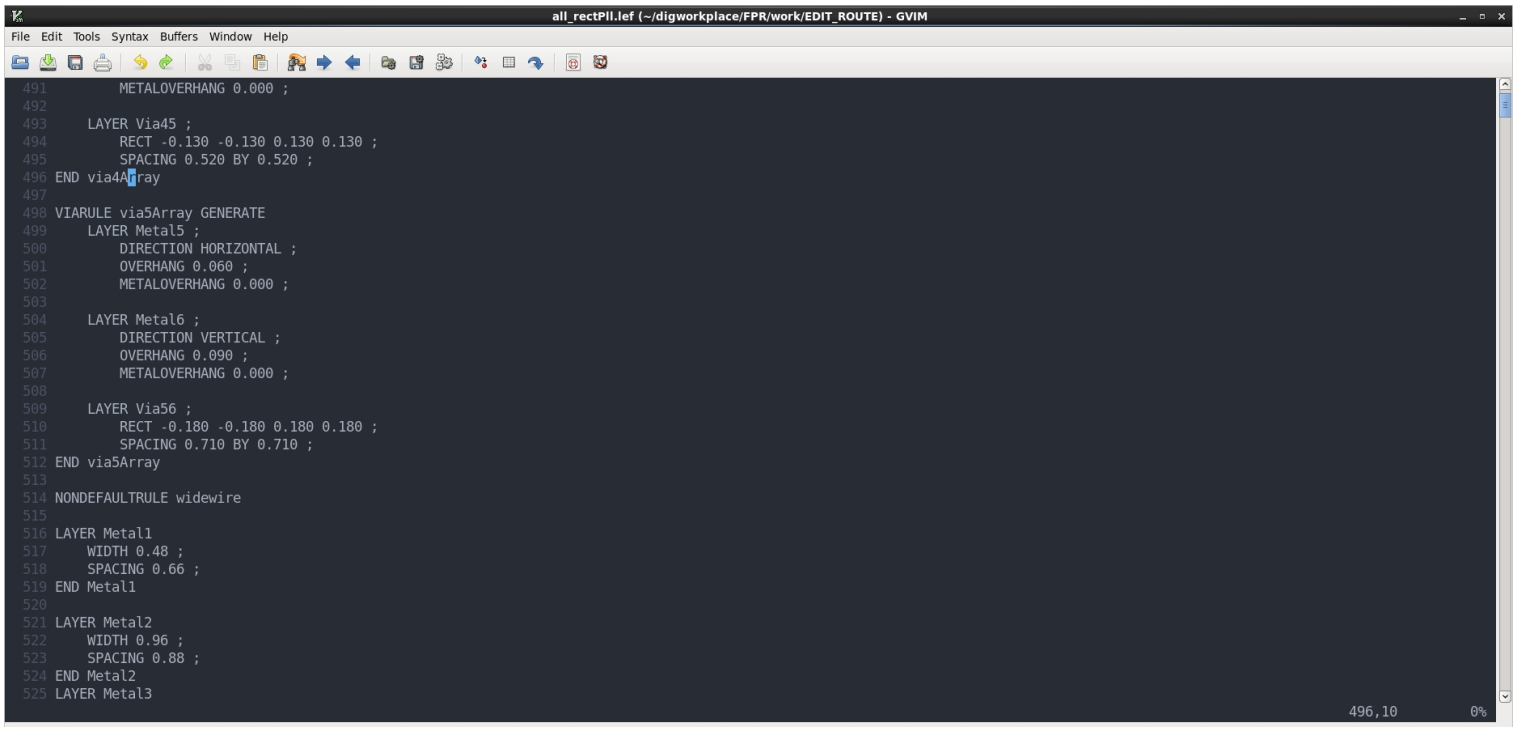
via1 via1new via2 via2new via3 via3new via4 via4new via5 via5new ndvia12 ndvia23 ndvia34 ndvia45 ndvia56

innovus #> dbGet head.rules.name

widewire

innovus #> dbGet head.viaRuleGenerates.name

via1Array via2Array via3Array via4Array via5Array



rule.

*:Rule information (equivalent to LEF/DEF **NONDEFAULTRULE**)*

layerRules: List of pointers to the layers rules

vias: List of pointers to via, default or USEVIA or derived from mincut

viaRuleGenerate

*: Via rule information defined in the **LEF** or OpenAccess techfile*

layer.

:Layer (from technology source, LEF or OpenAccess)

direction: Layer pref. direction from LEF/OpenAccess

Legal enum: Diag135,Diag45, HVUnassigned, Horizontal, Vertical

extName: Layer name from **LEF**/OpenAccess technology file definition.

name: By default is the LEF or OA layer name.

If the layerNameNoAbbreviation global is set to 0, then .name uses an older style abbreviation that is NO longer recommended (**M1** for first routing layer, **M2** for second routing layer, etc.).

```
innovus #> set layerNameNoAbbreviation
    1
innovus #> dbGet head.layers.name
    Metal1 Metal2 Metal3 Metal4 Metal5 Metal6 Via12 Via23 Via34 Via45 Via56 OVERLAP
innovus #> set layerNameNoAbbreviation 0
    0
innovus #> dbGet head.layers.name
    M1 M2 M3 M4 M5 M6 Via12 Via23 Via34 Via45 Via56 OVERLAP
innovus #> set layerNameNoAbbreviation 1
    1
innovus #> dbGet head.layers.name
    Metal1 Metal2 Metal3 Metal4 Metal5 Metal6 Via12 Via23 Via34 Via45 Via56 OVERLAP
```

setAttribute

selectObjByProp <objType> <expression>

get_property

var_name

Specifies a pointer to a **collection** containing the object.

Note: If the collection contains more than one object, the get_property command returns the property values for each object in the specified collection.

property

[-clock clock_name]

[-view view_name]

[-quiet]

report_property

[-help]

{**collection** | list_of_collections}

[-property_list list of properties]

[{> | >>} filename]

```
innovus> report_property [get_pins mcore0/a0/g1626/B]
```

all_fanin

```
innovus> all_fanin -to mcore0/a0/g1626/B
```