Innovus data exchange

defOut

Writes the specified information to a DEF file. By default, the defOut command writes floorplan, and all placed and unplaced standard cell information to the DEF file.

The defOut command does not support multiple groups per region. The defOut command supports rectilinear die area.

~~-placement~~

-floorplan

Writes the floorplan data to the DEF file. This information includes rows, chip size, CLASS PAD instances, CLASS BLOCK instances, and all placed standard cells.

-scanChain

Writes scan chain information to the DEF file.

-netlist

Writes the netlist (that is, the routing connectivity information) to the DEF file. This option is typically used after the design has been placed.

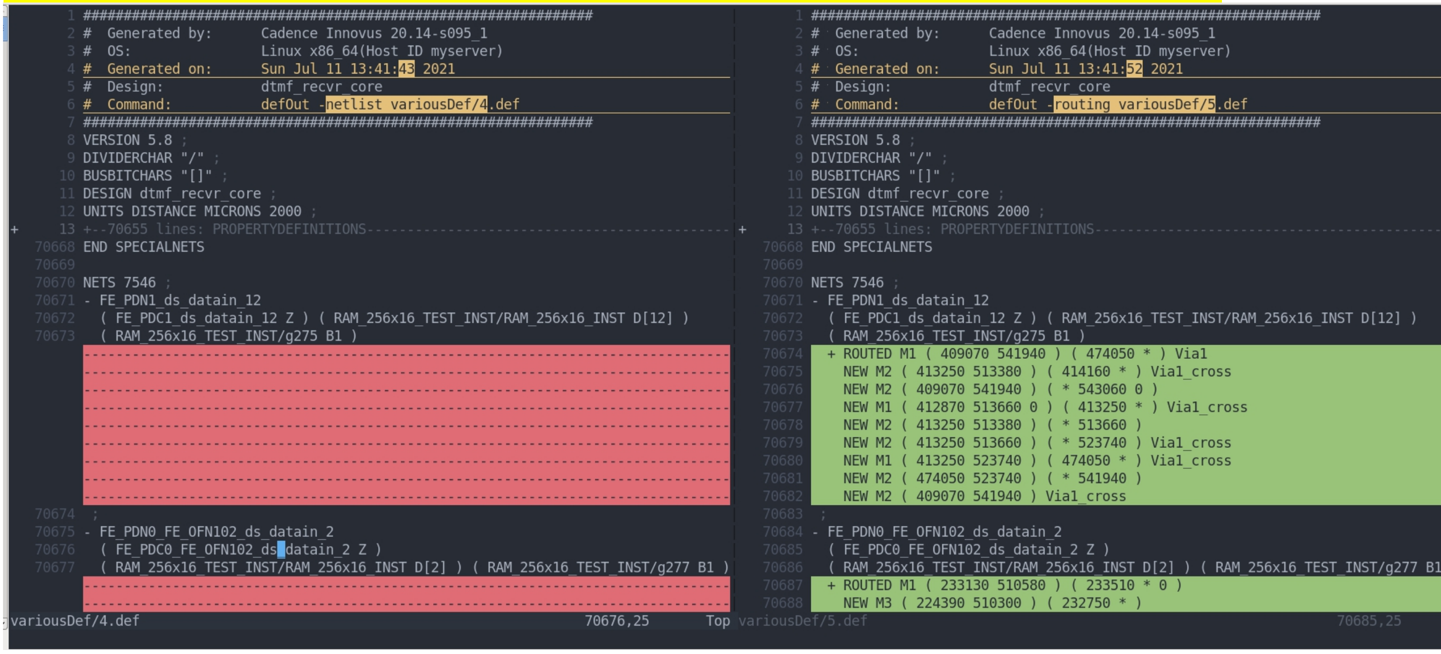
Note: Use -netlist with -unplaced if the design has not been placed.

-routing

Writes the routing information to the **NETS** section of the DEF file. If the -routing option is not specified, b oth power and signal vias are not written to the DEF file.

Note: The -routing parameter implies the -netlist parameter; therefore, if you specify this parameter, you do not need to specify -netlist.

-netlist only add routing connectivity information, -routing add physical routing like sWire



```
################################################################
#  Generated by:      Cadence Innovus 20.14-s095_1
#  OS:                Linux x86_64(Host ID myserver)
#  Generated on:      Sun Jul 11 13:41:43 2021
#  Design:            dtmf_recvr_core
#  Command:           defOut -netlist variousDef/4.def
################################################################
VERSION 5.8 ;
DIVIDERCHAR "/" ;
BUSBITCHARS "[]" ;
DESIGN dtmf_recvr_core ;
UNITS DISTANCE MICRONS 2000 ;
+--70655 lines: PROPERTYDEFINITIONS------------------------------------
END SPECIALNETS

NETS 7546 ;
- FE_PDN1_ds_datain_12
  ( FE_PDC1_ds_datain_12 Z ) ( RAM_256x16_TEST_INST/RAM_256x16_INST D[12] )
  ( RAM_256x16_TEST_INST/g275 B1 )
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
 ;
- FE_PDN0_FE_OFN102_ds_datain_2
  ( FE_PDC0_FE_OFN102_ds_datain_2 Z )
  ( RAM_256x16_TEST_INST/RAM_256x16_INST D[2] ) ( RAM_256x16_TEST_INST/g277 B1 )
-----------------------------------------------------------------------
-----------------------------------------------------------------------
variousDef/4.def                                        70676,25       Top
```

```
################################################################
#  Generated by:      Cadence Innovus 20.14-s095_1
#  OS:                Linux x86_64(Host ID myserver)
#  Generated on:      Sun Jul 11 13:41:52 2021
#  Design:            dtmf_recvr_core
#  Command:           defOut -routing variousDef/5.def
################################################################
VERSION 5.8 ;
DIVIDERCHAR "/" ;
BUSBITCHARS "[]" ;
DESIGN dtmf_recvr_core ;
UNITS DISTANCE MICRONS 2000 ;
+--70655 lines: PROPERTYDEFINITIONS------------------------------------
END SPECIALNETS

NETS 7546 ;
- FE_PDN1_ds_datain_12
  ( FE_PDC1_ds_datain_12 Z ) ( RAM_256x16_TEST_INST/RAM_256x16_INST D[12] )
  ( RAM_256x16_TEST_INST/g275 B1 )
  + ROUTED M1 ( 409070 541940 ) ( 474050 * ) Via1
    NEW M2 ( 413250 513380 ) ( 414160 * ) Via1_cross
    NEW M2 ( 409070 541940 ) ( * 543060 0 )
    NEW M1 ( 412870 513660 0 ) ( 413250 * ) Via1_cross
    NEW M2 ( 413250 513380 ) ( * 513660 )
    NEW M2 ( 413250 513660 ) ( * 523740 ) Via1_cross
    NEW M1 ( 413250 523740 ) ( 474050 * ) Via1_cross
    NEW M2 ( 474050 523740 ) ( * 541940 )
    NEW M2 ( 409070 541940 ) Via1_cross
 ;
- FE_PDN0_FE_OFN102_ds_datain_2
  ( FE_PDC0_FE_OFN102_ds_datain_2 Z )
  ( RAM_256x16_TEST_INST/RAM_256x16_INST D[2] ) ( RAM_256x16_TEST_INST/g277 B1
  + ROUTED M1 ( 233130 510580 ) ( 233510 * 0 )
    NEW M3 ( 224390 510300 ) ( 232750 * )
variousDef/5.def                                        70685,25
```

defOutLefVia

defOutLefVia {0 | 1}

Type: Integer

Default: 0

Controls whether defOut writes out **LEF via** information in DEF. By default, defOut does not write out vias defined in LEF. Set defOutLefVia to 1 to write out LEF vias in the DEF file.

defOutBySection

saveNetlist

-excludeCellInst cellName

Excludes the instances of the specified cells from the netlist. You can exclude logical or physical instances. The cellName can be a single cell name or a list of cell names separated by a space and surrounded by braces ({}) or double quotation marks ("").

For example: -excludeCellInst {cell1 cell2} or "cell1 cell2".

-excludeLeafCell

Writes I/O instances, macro or block instances, and standard cell instances to the netlist, but does not include leaf cell definitions in the netlist.

-flat

Writes a flattened Verilog netlist.

Default: If you omit this parameter, and if the original netlist loaded was hierarchical, the netlist is written in hierarchical Verilog format

-phys

Writes out physical cell instances, and inserts power and ground nets in the netlist. This is used for LVS and for designs with multiple supply voltages. Hierarchical power and ground nets are also supported.

Note: -phys creates a netlist that **should not** be read back into Innovus.

saveNetlist -phys == saveNetlist -includePhysicalInst -includePowerGround

How to write out a netlist th...

-includePowerGround

Includes power and ground connections in the netlist file.

-includePhysicalInst

Activates physical netlisting by including physical instances, such as fillers, in the netlist. With this parameter, GNC-based/physical connections, such as vss, are used for tie-off pins.

-includePhysicalCell cellName

Includes the physical instances of the specified cells in the netlist. With this parameter, 1'b1/1'b0-style, logical connections are used for tie-off pins.

The cellName can be a single cell name or a list of cell names separated by a space and surrounded by braces ({}) or double quotes (""), for example, -includePhysicalCell {cell1 cell2} or "cell1 cell2".

Default: If you omit this parameter, no physical cell module definitions will be in the netlist file.

Note: -includePhysicalCell creates a netlist that should not be read back into Innovus.

-excludeTopCellPGPort portName

Excludes specified power and ground port(s) from the top cell's module port list.

The portName can be a single port name or a list of port names separated by a space and surrounded by braces ({}) or double quotation marks ("").

For example: -excludeTopCellPGPort {VSS VDD} or "VSS VDD".

saveDesign

[-help]

[-addTiming]

[-noFill]

[-no_pvs_fill]

[-no_wait out_file]

{{fileName } | {{{-cellview {libname cellname viewname} | -view viewname} [-saveRestoreFile file_name]} }}

[-timingGraph [-noConstraint]]

[-user_path]

[-tcon]

[-rc]

[-libs]

[-lib2ldb]

[-def]

[-verilog]

[-tgz]

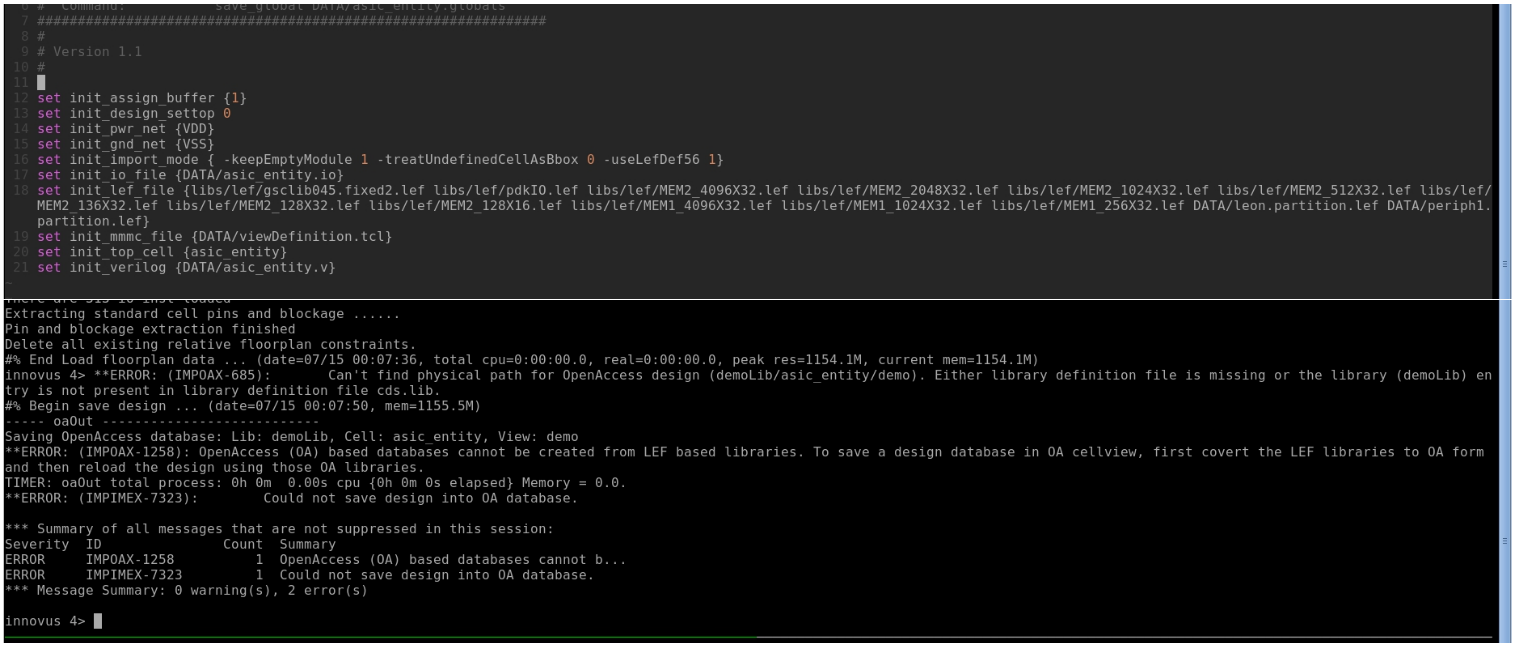
Saves the complete design database in the native Innovus format if fileName is specified, or as an OpenAccess database (OA DB) if -cellview or -view are specified.

The native Innovus DB format in output directory must be used when you have LEF files for technology and physical library data in your original initialization script (e.g. init_lef_file <files>), while the OA DB output must be used if you use OA libraries (e.g. init_oa_ref_lib <libs> and init_oa_search_libs <libs>) during initialization.

You can save a design to the same location from which you restored the design. The new data will overwrite the previous data.

You can use the saveDesign command at any time after init_design

By default, **external file references** used inside the **fileName.dat** directory (e.g. to library files, or SDC files that are **not modified**) are stored using a portable format that allows you to copy **fileName.dat** anywhere in the **same network** where the **full path names** are still valid, and then restore it with restoreDesign. This portable format is not supported for OA DBs that require the usage of a cds.lib file to find physical file locations.



```
#  Command:            save_global DATA/asic_entity.globals
################################################################
#
# Version 1.1
#

set init_assign_buffer {1}
set init_design_settop 0
set init_pwr_net {VDD}
set init_gnd_net {VSS}
set init_import_mode { -keepEmptyModule 1 -treatUndefinedCellAsBbox 0 -useLefDef56 1}
set init_io_file {DATA/asic_entity.io}
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}
set init_mmmc_file {DATA/viewDefinition.tcl}
set init_top_cell {asic_entity}
set init_verilog {DATA/asic_entity.v}
```

```
Extracting standard cell pins and blockage ......
Pin and blockage extraction finished
Delete all existing relative floorplan constraints.
#% End Load floorplan data ... (date=07/15 00:07:36, total cpu=0:00:00.0, real=0:00:00.0, peak res=1154.1M, current mem=1154.1M)
innovus 4> **ERROR: (IMPOAX-685):       Can't find physical path for OpenAccess design (demoLib/asic_entity/demo). Either library definition file is missing or the library (demoLib) en
try is not present in library definition file cds.lib.
#% Begin save design ... (date=07/15 00:07:50, mem=1155.5M)
----- oaOut --------------------------
Saving OpenAccess database: Lib: demoLib, Cell: asic_entity, View: demo
**ERROR: (IMPOAX-1258): OpenAccess (OA) based databases cannot be created from LEF based libraries. To save a design database in OA cellview, first covert the LEF libraries to OA form
and then reload the design using those OA libraries.
TIMER: oaOut total process: 0h 0m  0.00s cpu {0h 0m 0s elapsed} Memory = 0.0.
**ERROR: (IMPIMEX-7323):        Could not save design into OA database.

*** Summary of all messages that are not suppressed in this session:
Severity  ID               Count  Summary
ERROR     IMPOAX-1258          1  OpenAccess (OA) based databases cannot b...
ERROR     IMPIMEX-7323         1  Could not save design into OA database.
*** Message Summary: 0 warning(s), 2 error(s)

innovus 4> 
```

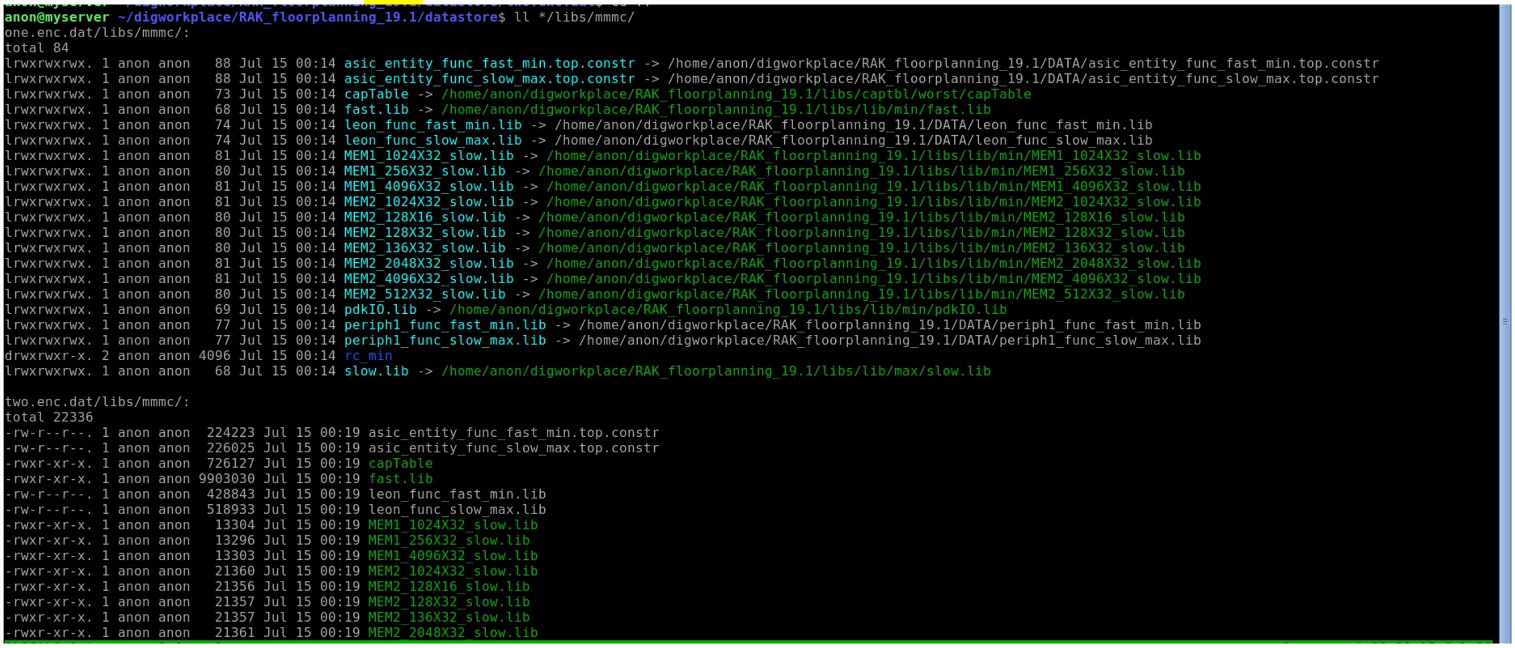
-libs

Normally external file references, such as references to the **LEF**, **.lib**, **SDC** files are saved using a full-path name, symbolic links to the original files inside a libs sub-directory inside the DB directory.

This option forces copies of the files, rather than symbolic links, in the libs sub-directory so there are no external file references. This makes the DB directory much larger, but it is suitable for sending to a different network where the library and external data files are not available.

innovus #> saveDesign datastore/one.enc

innovus #> saveDesign **-libs** datastore/two.enc



```
anon@myserver ~/digworkplace/RAK_floorplanning_19.1/datastore$ ll */libs/mmmc/
one.enc.dat/libs/mmmc/:
total 84
lrwxrwxrwx. 1 anon anon   88 Jul 15 00:14 asic_entity_func_fast_min.top.constr -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/asic_entity_func_fast_min.top.constr
lrwxrwxrwx. 1 anon anon   88 Jul 15 00:14 asic_entity_func_slow_max.top.constr -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/asic_entity_func_slow_max.top.constr
lrwxrwxrwx. 1 anon anon   73 Jul 15 00:14 capTable -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/captbl/worst/capTable
lrwxrwxrwx. 1 anon anon   68 Jul 15 00:14 fast.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/fast.lib
lrwxrwxrwx. 1 anon anon   74 Jul 15 00:14 leon_func_fast_min.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/leon_func_fast_min.lib
lrwxrwxrwx. 1 anon anon   74 Jul 15 00:14 leon_func_slow_max.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/leon_func_slow_max.lib
lrwxrwxrwx. 1 anon anon   81 Jul 15 00:14 MEM1_1024X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM1_1024X32_slow.lib
lrwxrwxrwx. 1 anon anon   80 Jul 15 00:14 MEM1_256X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM1_256X32_slow.lib
lrwxrwxrwx. 1 anon anon   81 Jul 15 00:14 MEM1_4096X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM1_4096X32_slow.lib
lrwxrwxrwx. 1 anon anon   81 Jul 15 00:14 MEM2_1024X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_1024X32_slow.lib
lrwxrwxrwx. 1 anon anon   80 Jul 15 00:14 MEM2_128X16_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_128X16_slow.lib
lrwxrwxrwx. 1 anon anon   80 Jul 15 00:14 MEM2_128X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_128X32_slow.lib
lrwxrwxrwx. 1 anon anon   80 Jul 15 00:14 MEM2_136X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_136X32_slow.lib
lrwxrwxrwx. 1 anon anon   81 Jul 15 00:14 MEM2_2048X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_2048X32_slow.lib
lrwxrwxrwx. 1 anon anon   81 Jul 15 00:14 MEM2_4096X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_4096X32_slow.lib
lrwxrwxrwx. 1 anon anon   80 Jul 15 00:14 MEM2_512X32_slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/MEM2_512X32_slow.lib
lrwxrwxrwx. 1 anon anon   69 Jul 15 00:14 pdkIO.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/min/pdkIO.lib
lrwxrwxrwx. 1 anon anon   77 Jul 15 00:14 periph1_func_fast_min.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/periph1_func_fast_min.lib
lrwxrwxrwx. 1 anon anon   77 Jul 15 00:14 periph1_func_slow_max.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/DATA/periph1_func_slow_max.lib
drwxrwxr-x. 2 anon anon 4096 Jul 15 00:14 rc_min
lrwxrwxrwx. 1 anon anon   68 Jul 15 00:14 slow.lib -> /home/anon/digworkplace/RAK_floorplanning_19.1/libs/lib/max/slow.lib

two.enc.dat/libs/mmmc/:
total 22336
-rw-r--r--. 1 anon anon  224223 Jul 15 00:19 asic_entity_func_fast_min.top.constr
-rw-r--r--. 1 anon anon  226025 Jul 15 00:19 asic_entity_func_slow_max.top.constr
-rwxr-xr-x. 1 anon anon  726127 Jul 15 00:19 capTable
-rwxr-xr-x. 1 anon anon 9903030 Jul 15 00:19 fast.lib
-rw-r--r--. 1 anon anon  428843 Jul 15 00:19 leon_func_fast_min.lib
-rw-r--r--. 1 anon anon  518933 Jul 15 00:19 leon_func_slow_max.lib
-rwxr-xr-x. 1 anon anon   13304 Jul 15 00:19 MEM1_1024X32_slow.lib
-rwxr-xr-x. 1 anon anon   13296 Jul 15 00:19 MEM1_256X32_slow.lib
-rwxr-xr-x. 1 anon anon   13303 Jul 15 00:19 MEM1_4096X32_slow.lib
-rwxr-xr-x. 1 anon anon   21360 Jul 15 00:19 MEM2_1024X32_slow.lib
-rwxr-xr-x. 1 anon anon   21356 Jul 15 00:19 MEM2_128X16_slow.lib
-rwxr-xr-x. 1 anon anon   21357 Jul 15 00:19 MEM2_128X32_slow.lib
-rwxr-xr-x. 1 anon anon   21357 Jul 15 00:19 MEM2_136X32_slow.lib
-rwxr-xr-x. 1 anon anon   21361 Jul 15 00:19 MEM2_2048X32_slow.lib
```

-def

Saves a DEF file, in addition to the normal DB files in the DB directory. This parameter is not normally used for flat or block flows, and is not used by restoreDesign. It can be useful in a hierarchical flow, where you want to use assembleDesign to read only the **netlist** and the **DEF** files without opening up each design DB and writing out the Verilog and DEF files separately (see -verilog).

-verilog

Saves a Verilog file inside the DB directory, in addition to the normal DB files. This parameter is not normally used for flat or block flows, and is not used by restoreDesign. It can be used in a hierarchical flow, where you want to use assembleDesign to read only the Verilog and the DEF files without opening up each design DB and writing out the Verilog and DEF files separately (see -def).

e.g.

```
innovus #> saveDesign -verilog -def   datastore/three.enc
    ...
    Writing Netlist "datastore/three.enc.dat/asic_entity.v.gz" ...
    ...
    Saving Def ...
    Writing DEF file 'datastore/three.enc.dat/asic_entity.def.gz', current time is Thu Jul 15 00:44:39 2021 ...
```

additonal asic_entity.def.gz, asic_entity.v.gz is created.

-tcon

By default, if the SDC timing constraints have not been modified, a link to the original external SDC file is kept to reduce the disk size. If the SDC timing constraints were modified, then the SDC files are automatically stored in the DB directory. This option will force all SDC files to be copied locally inside the DB directory, even if they did not change.

Note: Any future sessions that use restoreDesign and then saveDesign will have references to SDC files inside this design directory rather than to the original external location. So, you must keep this DB directory around as long as any other DB references it (or you must use saveDesign in the later session to force another local copy of the SDC files).

saveFPlan

file

[-help]

[-noName]

[-objType {macro | pin | bndry | special_route | pin_constraint}]

[-xml]

Saves the floorplan information to a file. The command saves all the floorplan blocks that are placed/fixed/covered, in the floorplan file. When you save a floorplan, the Innovus software saves all objects which can be saved and any specific properties which are affected if the data is saved. By default, the floorplan information is saved as a Version 8 floorplan file. After you save a floorplan, the Innovus software creates the following files:

A general floorplanning file with the extension **.fp**

A power route data file with the extension **.fp.spr**

Note: If there is an entry in the IO Cell Libraries field in the Design Import form, a third file is created with the extension **.fp.areaio**.

if there is relative place, a file is created with the extension **.fp.relFPlan**

loadFPlan

file

[-help]

[-noEqualizePtnHInst]

[-objType {macro | pin | bndry | special_route | pin_constraint}]

[-xml]

Loads a floorplan file. You can use this command after importing the design.

Note:

When you run the loadFPlan command, the Innovus software **removes any existing floorplan information** (with -objType, only the specified section will be load and other floorplan objects remaining unchanged)and reads in the new information. The loadFPlan command honors the blackbox min/max aspect ratio information in the floorplan information file.

Blocks and instances loaded with this command are set as preplaced.

The loadFPlan command loads the user-specified seed information by default.

writeFPlanScript

[-help]

[-appendToFile]

[-compactRow]

-fileName fileName

[-selected | -sections {boundary row placeBlockages routeBlockages pinBlockages groups constraints pins ioPad areaIO bump partitions blocks globalNetConnect busSinkGroups blackboxes netGroupAndBusGuide relativefplan pinGroups pinGuides}]

Allows you to write out the specified floorplan sections and source the output file after init_design. With this command the tool would write a file that you can examine, edit, and then source to selectively (or completely) define your floorplan. You can specify the floorplan section to write out. The writeFPlanScript command writes out a file with Tcl commands to recreate the floorplan. It can optionally write out specific floorplan sections listed with option -sections. The file includes commands to delete any existing data for each section if any exists.

write_sdc