IC Synthesis and Optimization

Lab 8: Physical Synthesis: Finishing

Objective

To study inserting redundant vias and filler cells.

Theoretical part

Yield-optimized via rate is reported in PR Summary, using command **report_design -all**.

Voids in vias is a serious issue in manufacturing. To handle this issue, different solutions are available in ICC II.

- Add backup vias: known as redundant vias.

Filler Cell Insertion is for better yield (layer density of the chip needs to be uniform, this will provide better yield).

Some placement sites can remain empty on some rows.

The final step to write GDS and parasitic files.

Practical part

- Move to directory lab8/work

```
icc2_shell -gui -64
source ../../common/common.tcl
open_lib ${ARC_TOP}
open_block ${DESIGN_NAME}_6_complete
```

Yield-optimized via rate reported in PR Summary, using command

```
report_design -all
```

1 Redundant vias.

To do this, use *Task->Task Assistant->Routing->add redundant vias*. The dialog box is shown in Fig. 2.

```
add_redundant_vias
```

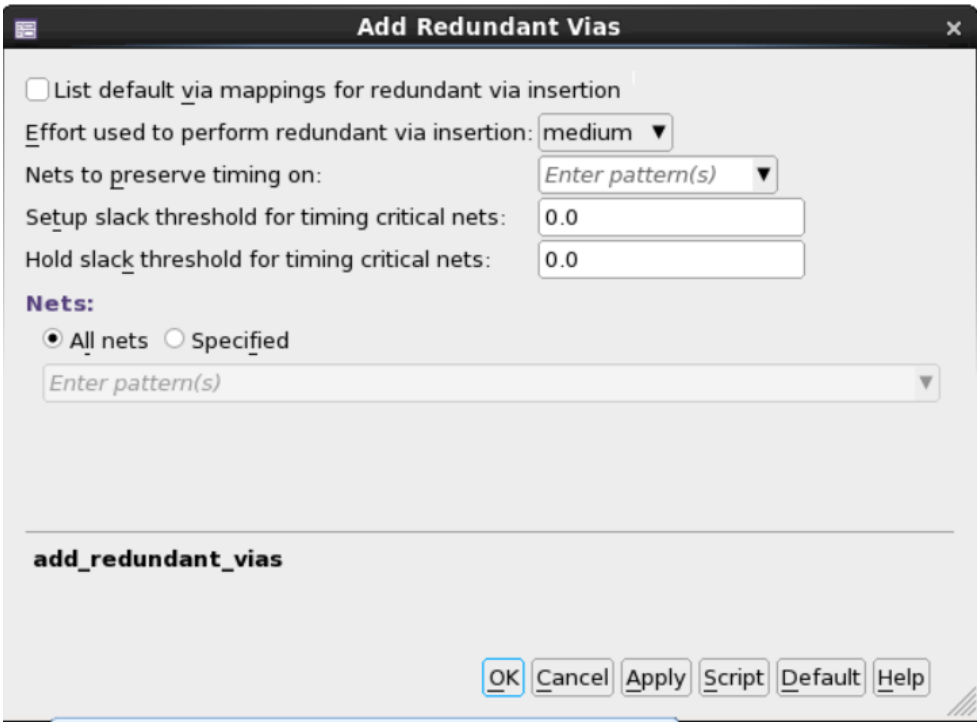


Fig.2. Dialog box of add redundant vias

2 Insert filler

Filler cells are added using *create_stdcell_fillers* command.

```
set pnr_std_fillers "SAEDRVT14_FILL*"
set std_fillers ""
foreach filler $pnr_std_fillers {lappend std_fillers "*/${filler}"}
create_stdcell_fillers \
        -lib_cells $std_fillers
```

In this lab metal fill is also done by the following steps:

```
set_app_options -name signoff.create_metal_fill.runset -value "${icv_mfill_runset}"
signoff_create_metal_fill -select_layers {M2 M6}
```

The runset file is a combination of rules for metal fill insertion which is provided by the technology vendor. Please also note that for metal fill IC Validator tool is used and loaded ICV's version must be later than ICC2's version. Dialog box for metal fill insertion is shown in Fig.3.

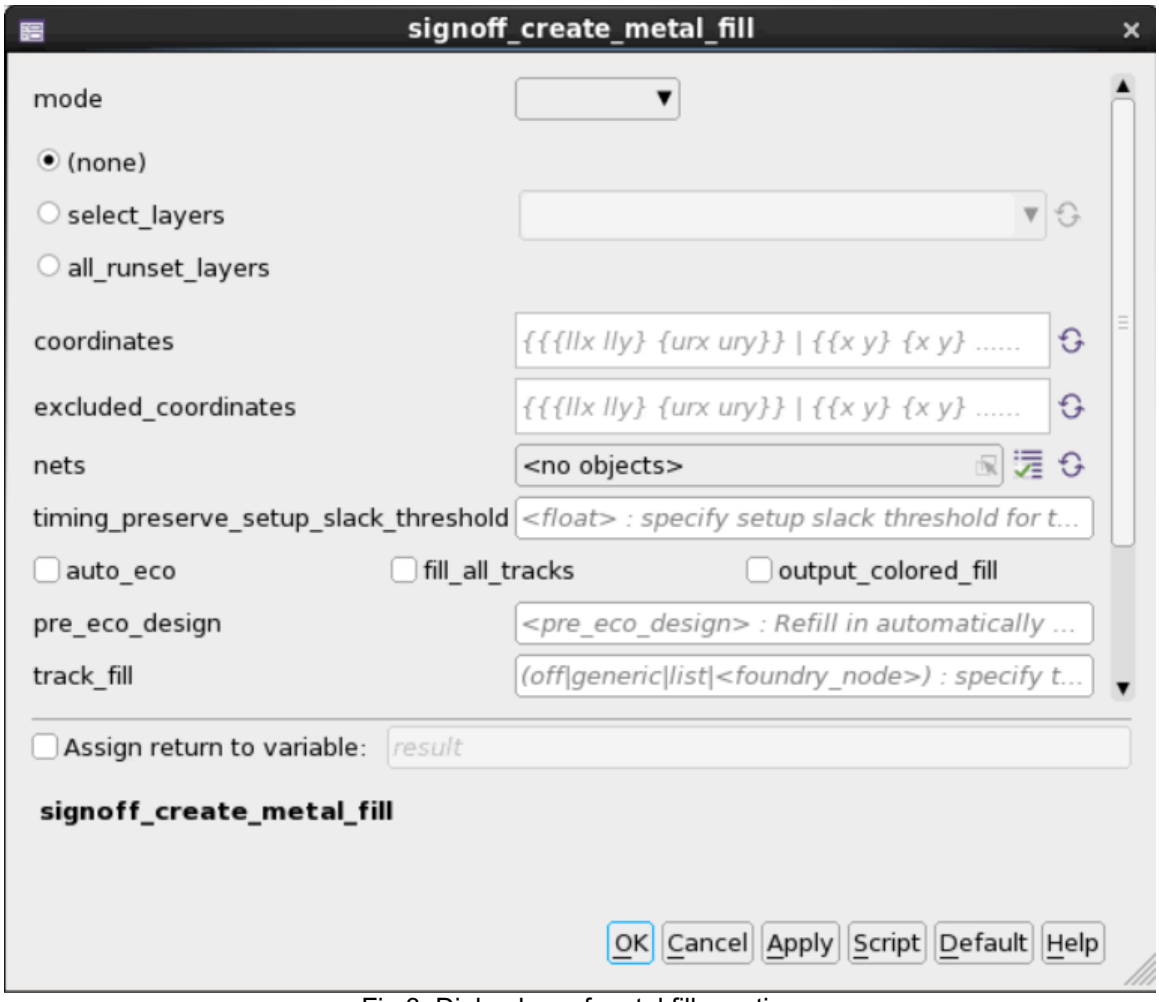


Fig.3. Dialog box of metal fill creation

The final step is connecting standard cells to power ports by *connect_pg_net* command.

```
connect_pg_net -net VDD [get_pins -physical_context */VDD]
connect_pg_net -net VSS [get_pins -physical_context */VSS]
```

3 Verify PG routing

After doing this, there is need to check if all the VDD/VSS pins of standard cells as well as macros have been connected to the PG straps. The commands *check_pg_connectivity* and *check_lvs* are used to check that.

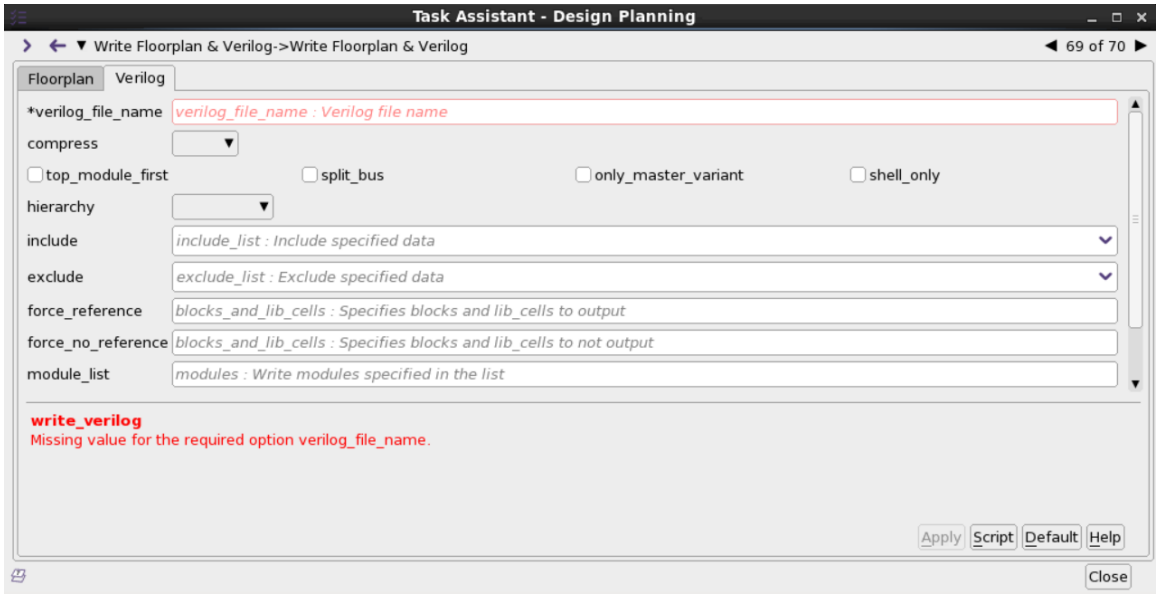
```
check_pg_connectivity -nets "VDD VSS"

check_lvs
```

4 Write Verilog, SDC, Parasitics, GDS

At the end some views must written out from design. The examples are Verilog file, SDC, parasitics file and GDS. The appropriate commands are used:

```
write_verilog \
      -include {pg_netlist unconnected_ports} \
      ../../results/${DESIGN_NAME}.pg.v

write_verilog -exclude {physical_only_cells} \
../../results/${DESIGN_NAME}.v

write_parasitics -format SPEF -output \
../../results/${DESIGN_NAME}.out.spef

write_sdc -output ../../results/${DESIGN_NAME}.out.sdc

save_block -as ${DESIGN_NAME}_7_finished

write_gds -design ${DESIGN_NAME}_7_finished \
        -layer_map $Gds_map_file \
        -keep_data_type \
        -fill include \
        -output_pin all \
        -merge_files $Std_cell_gds \
        -long_names \
        ../../results/${DESIGN_NAME}.gds
```

For example, in Fig.4 dialog box is shown for writing Verilog file from design.



Reports

1. Screenshots of each stage

2. Verification reports.