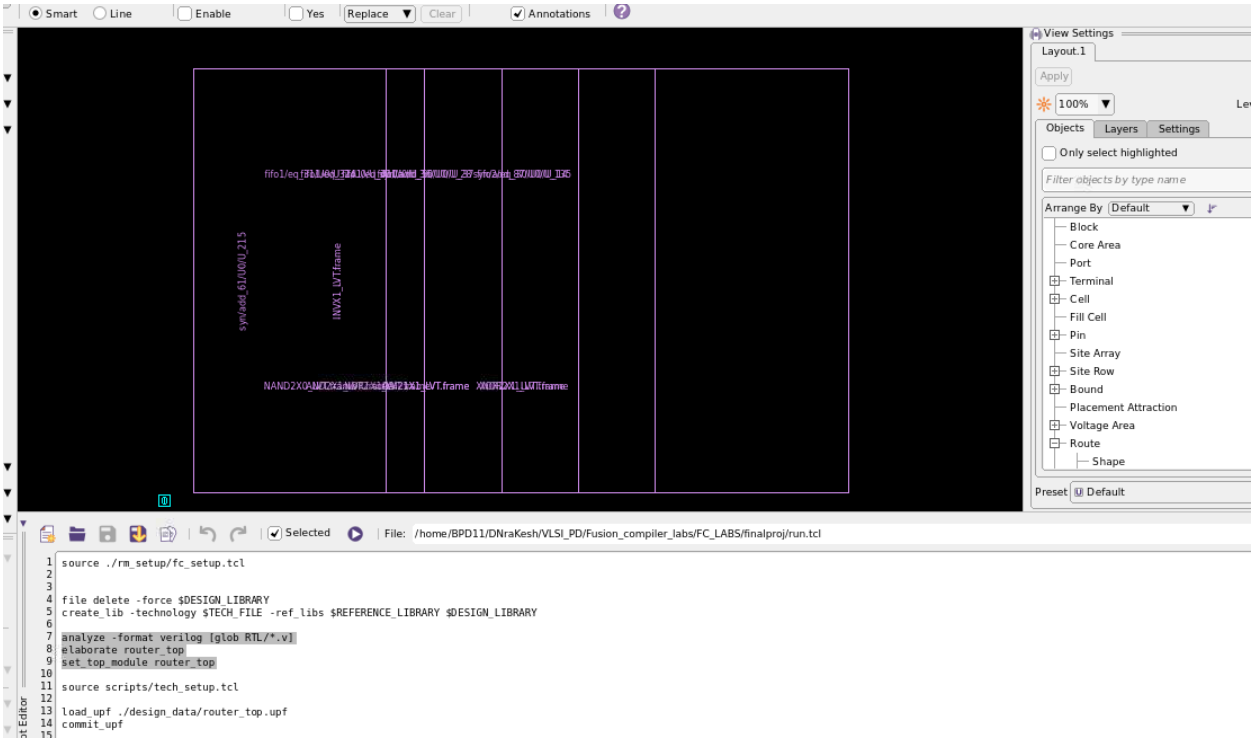
Router3x1 project report

Task1:: Create the Design Library, Read the RTL

Analyze the Verilog RTL files: analyze -format verilog [glob risc_rtl1/*.v] • Note: The glob command used above returns a list of all files that match the pattern. ⎪ Create a linked “top” block: elaborate risc_core set_top_module risc_core

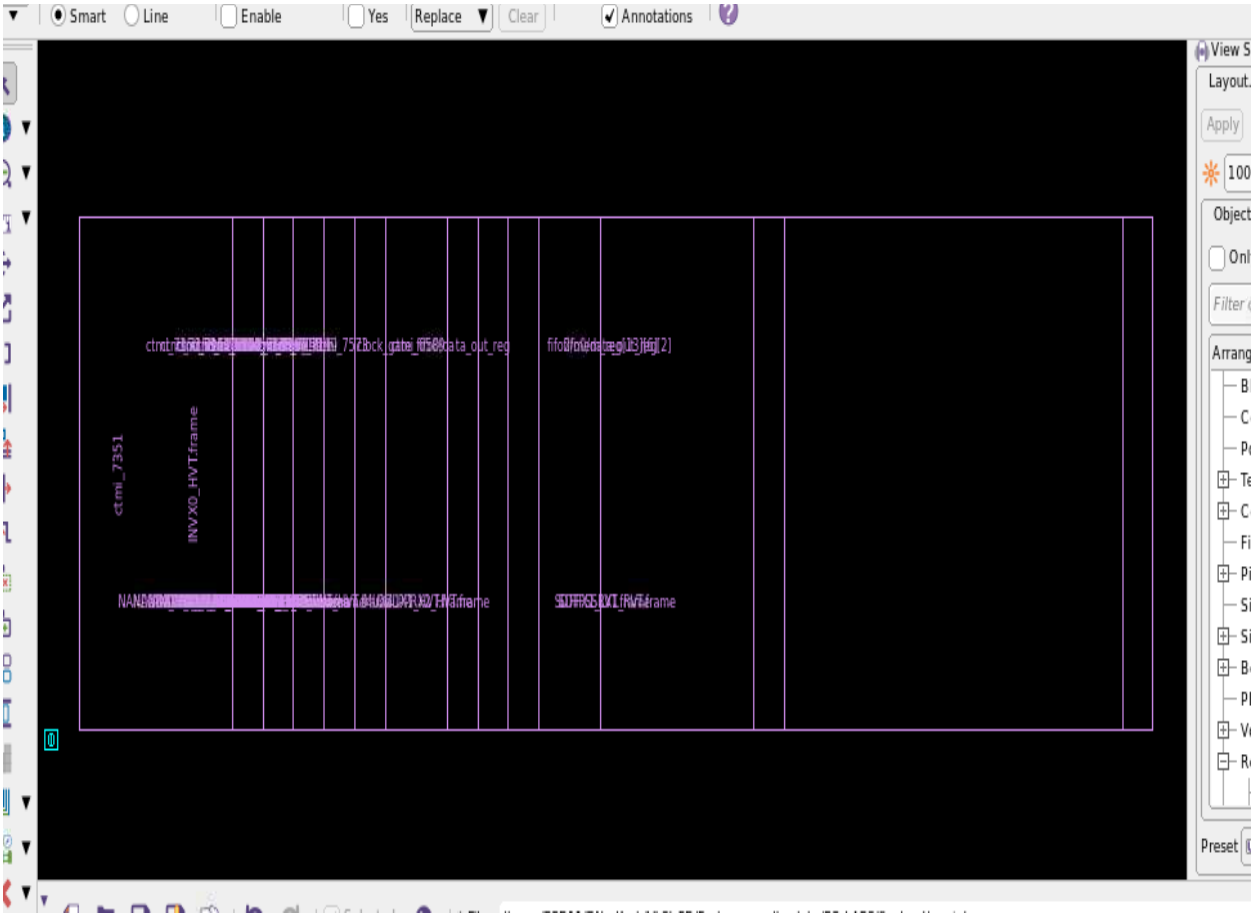


set_auto_floorplan_constraints -core_utilization 0.7 -side_ratio {1 2} -core_offset 10

floore plane is set using above command

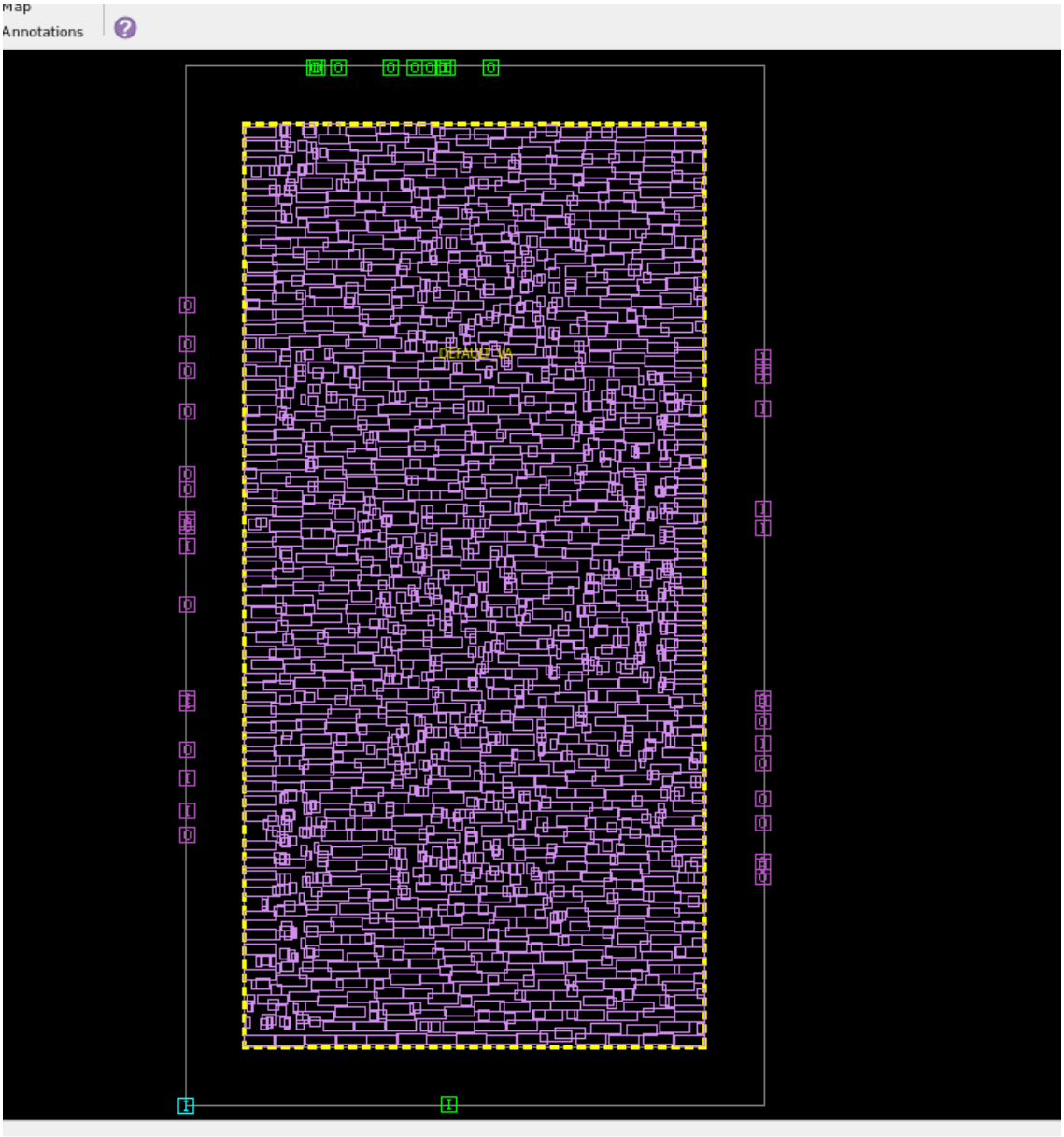
Task2: Run compile_fusion initial_map

Run stage 1 of compile, initial_map:



Task 3: Run compile_fusion logic_opto

Run stage 2 of compile, logic_opto:_run_compile_stage_save logic_opto ,You should see that the floorplan has been created automatically, pins are placed along all 4 edges, and a first coarse placement has taken place (zoom in to examine). Timing-driven logic optimization occurs before and after placement. Here is a screenshot of the design after stage



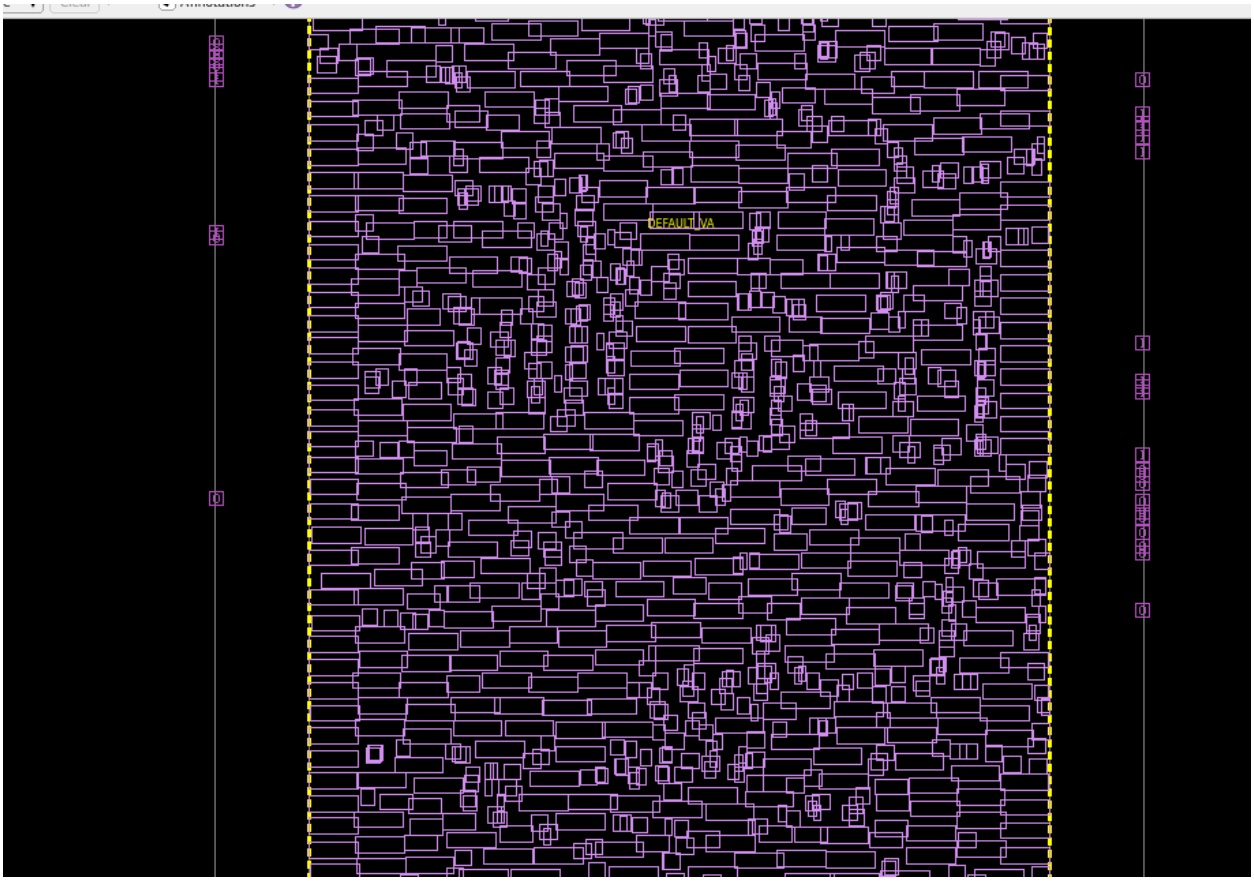
esh/VLSI_PD/Fusion_compiler_labs/FC_LABS/finalproj/run.tcl

Task 4: Run compile_fusion initial_place

Run stage 3 of compile, initial_place:_run_compile_stage_save initial_place

The placement performed here is buffering-aware timing-driven placement(confirm by searching for the second “Placement Options” table in the log file (compile _fusion.initial_place.log).

Note that the placement is still coarse, the standard cells are not legalized.

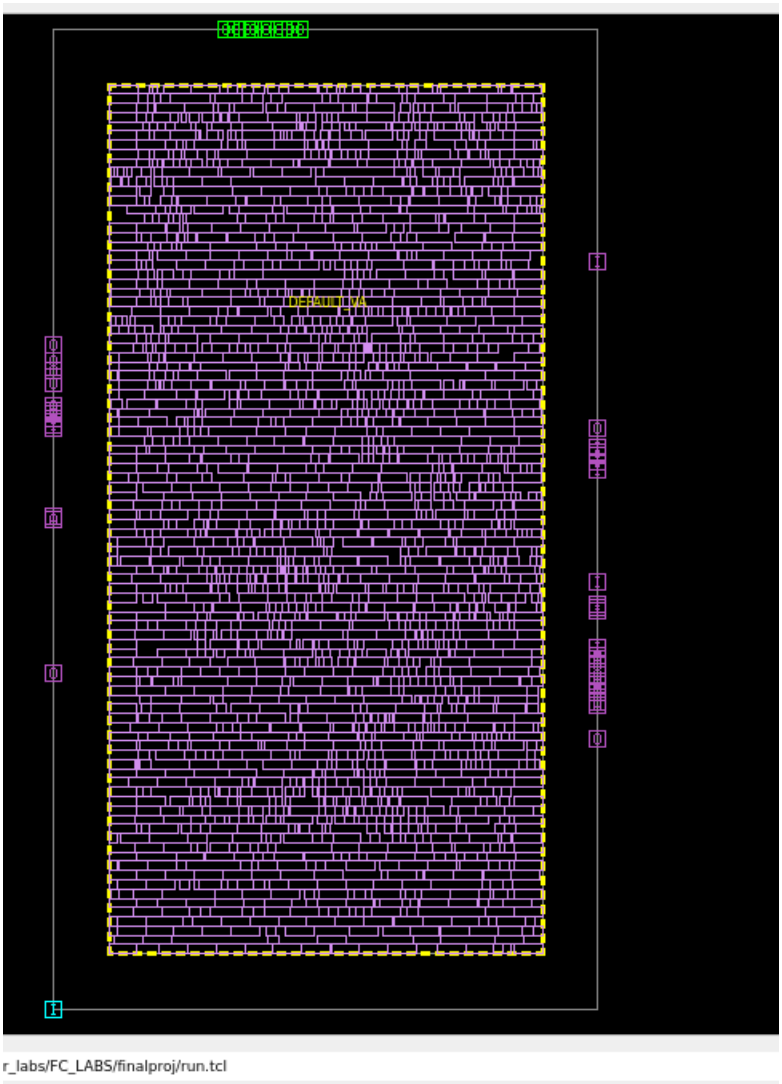


Task5::Run compile_fusion initial_opto

Run stage 5 of compile, initial_opto:_run_compile_stage_save initial_opto

This stage performs many optimization steps, including CCD, scan insertion, etc.

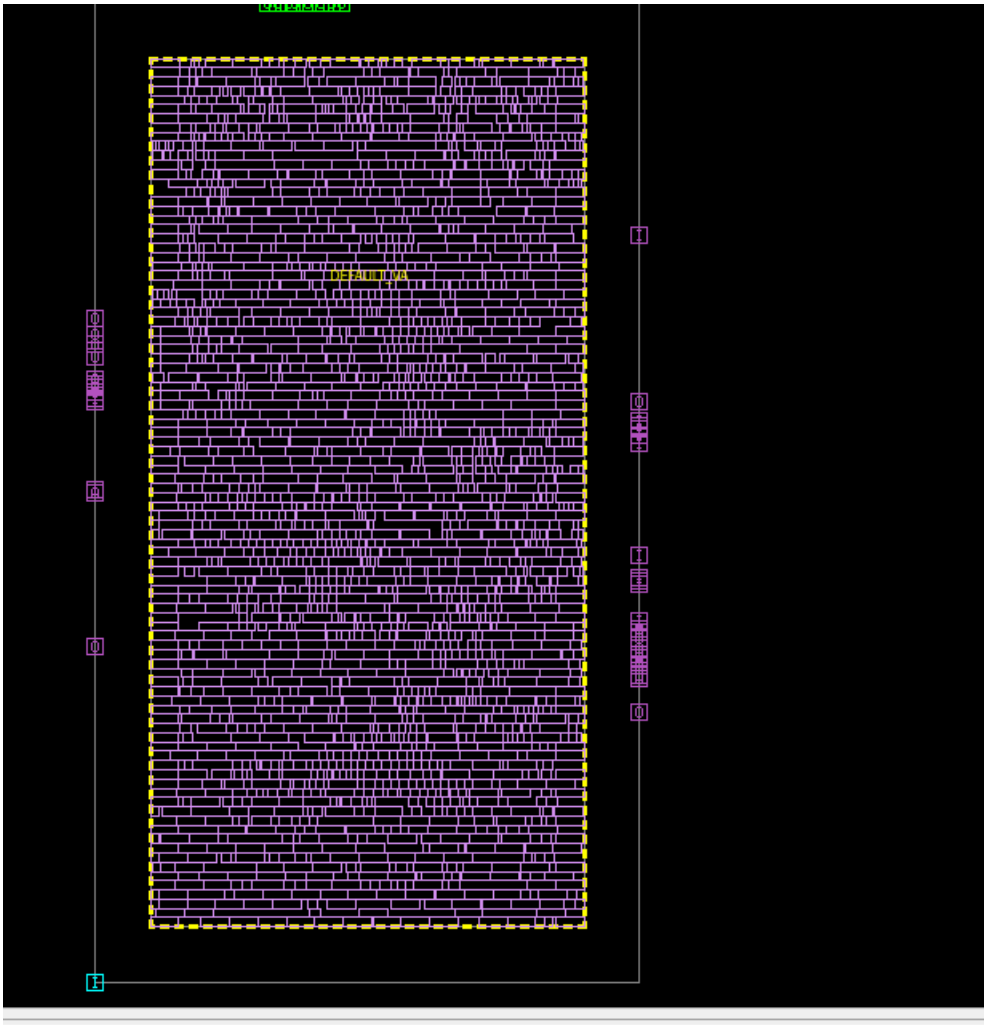
Zoom into the layout and you will see that placement seems to be fully legalized -standard cells are placed inside the placement rows and are not overlapping. Run the following command to check whether all cells are indeed placed legally:check_legality



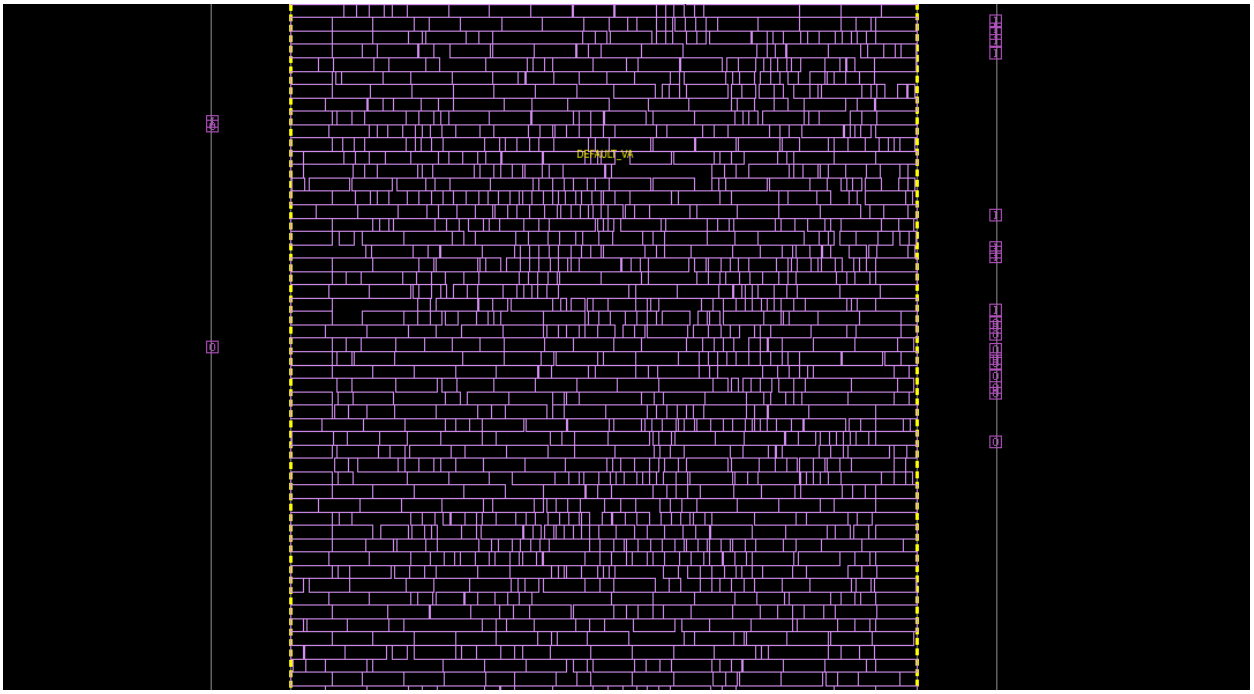
Task 6: Run compile_fusion final_place Run stage 6 of compile, final_place:_run_compile_stage_save final_place

Check placement legality again:check_legality

This time cell placement is fully legal. This is not the last stage though, more optimizations and incremental placement will occur during the 7th compile stage, after which final legalization takes place



- Run compile_fusion final_opt



Power report

```
  Cell Internal Power    = 6.75e+08 pW ( 86.5%)
  Net Switching Power    = 1.05e+08 pW ( 13.5%)
Total Dynamic Power      = 7.81e+08 pW (100.0%)

Cell Leakage Power       = 4.76e+08 pW


  Attributes
  ----------
      u  -  User defined power group

Power Group         Internal Power        Switching Power          Leakage Power            Total Power    (   %  )    Attrs
--------------------------------------------------------------------------------------------------------------------------------
io_pad                   0.00e+00              0.00e+00              0.00e+00              0.00e+00    (  0.0%)
memory                   0.00e+00              0.00e+00              0.00e+00              0.00e+00    (  0.0%)
black_box                0.00e+00              0.00e+00              0.00e+00              0.00e+00    (  0.0%)
clock_network            5.80e+08              8.95e+07              8.74e+06              6.78e+08    ( 54.0%)
register                 8.59e+07              3.09e+06              3.36e+08              4.25e+08    ( 33.8%)
sequential               0.00e+00              0.00e+00              0.00e+00              0.00e+00    (  0.0%)
combinational            9.40e+06              1.27e+07              1.31e+08              1.53e+08    ( 12.2%)
--------------------------------------------------------------------------------------------------------------------------------
Total                    6.75e+08 pW           1.05e+08 pW           4.76e+08 pW           1.26e+09 pW
1
Information: 3190 out of 3200 POW-046 messages were not printed due to limit 10  (MSG-3913)
Information: 87 out of 97 POW-069 messages were not printed due to limit 10  (MSG-3913)
```

Multi-Corner Multi-Mode Setup ⎫

In a different terminal window, open the file: Design_data/mcmm_risc_core.tcl ⎫

This script first creates the modes, corners, and scenarios needed for multi-corner multi-mode (MCMM) optimization of this design.

• Note: The script takes advantage of Tcl arrays (set array Name (var Name)) value) to create efficiently-coded for each loop.

Mcmm is loaded using below command

*****source scripts/mcmm/mcmm_router.tcl

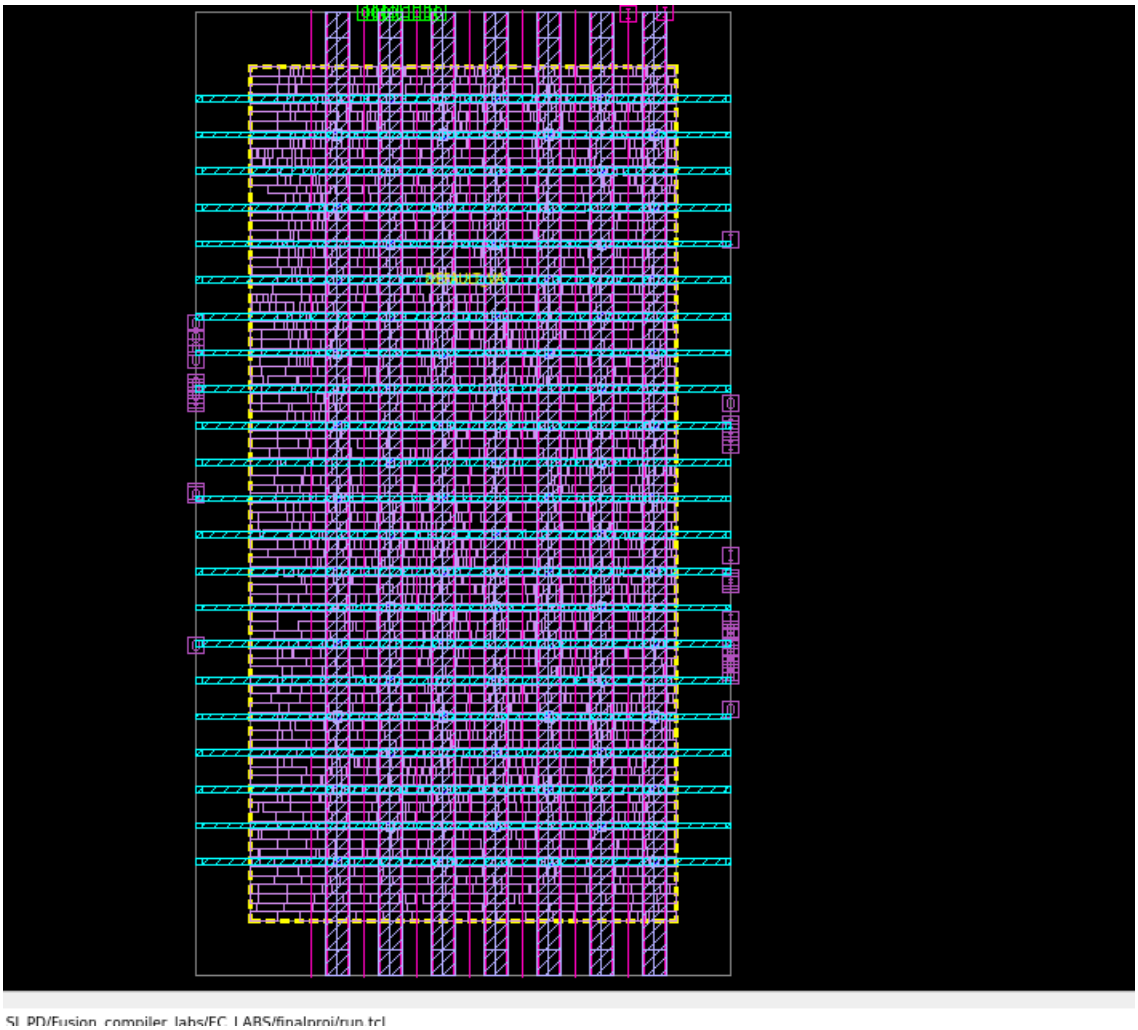
```
puts "RM-info : Running script [info script]\n"

##########################################################################################
# Tool: Fusion Compiler
# Script: mcmm_risc_core.tcl
# Applies MCMM constraints
##########################################################################################

remove_scenarios -all
remove_modes -all
remove_corners -all


set m_constr(func)         "risc_core_m_func.tcl"
set m_constr(test)         "router_m_test.tcl"

set c_constr(ss_125c)      "risc_core_c_ss_125c.tcl"
set c_constr(ss_m40c)      "risc_core_c_ss_m40c.tcl"
set c_constr(ff_m40c)      "risc_core_c_ff_m40c.tcl"

set s_constr(func.ss_125c) "risc_core_s_func.ss_125c.tcl"
set s_constr(func.ss_m40c) "risc_core_s_func.ss_m40c.tcl"
set s_constr(func.ff_m40c) "risc_core_s_func.ff_m40c.tcl"
set s_constr(test.ss_125c) "risc_core_s_test.ss_125c.tcl"
set s_constr(test.ff_m40c) "risc_core_s_test.ss_m40c.tcl"

#######################################
## Mode, corner and scenario creation
#######################################

foreach m [array names m_constr] {
        create_mode $m
}
foreach c [array names c_constr] {
        create_corner $c
}

foreach s [array names s_constr] {
        lassign [split $s "."] m c
        create_scenario -name $s -mode $m -corner $c
}

#######################################
## Populate modes, corners and scenarios
#######################################

foreach m [array names m_constr] {
        current_mode $m
```

Power and Ground (PG) Prototyping

PG prototyping can help you create a basic PG mesh very quickly. All you need to specify are the PG net names, the layers, and the percentage of the layers you want to use for PG routing. This is most commonly used as a placeholder for the final power mesh, to check congestion issues that a PG mesh may cause.) From the Tasks palette, select PG Planning PG Prototyping

PG is source using below command

source scripts/pns.tcl



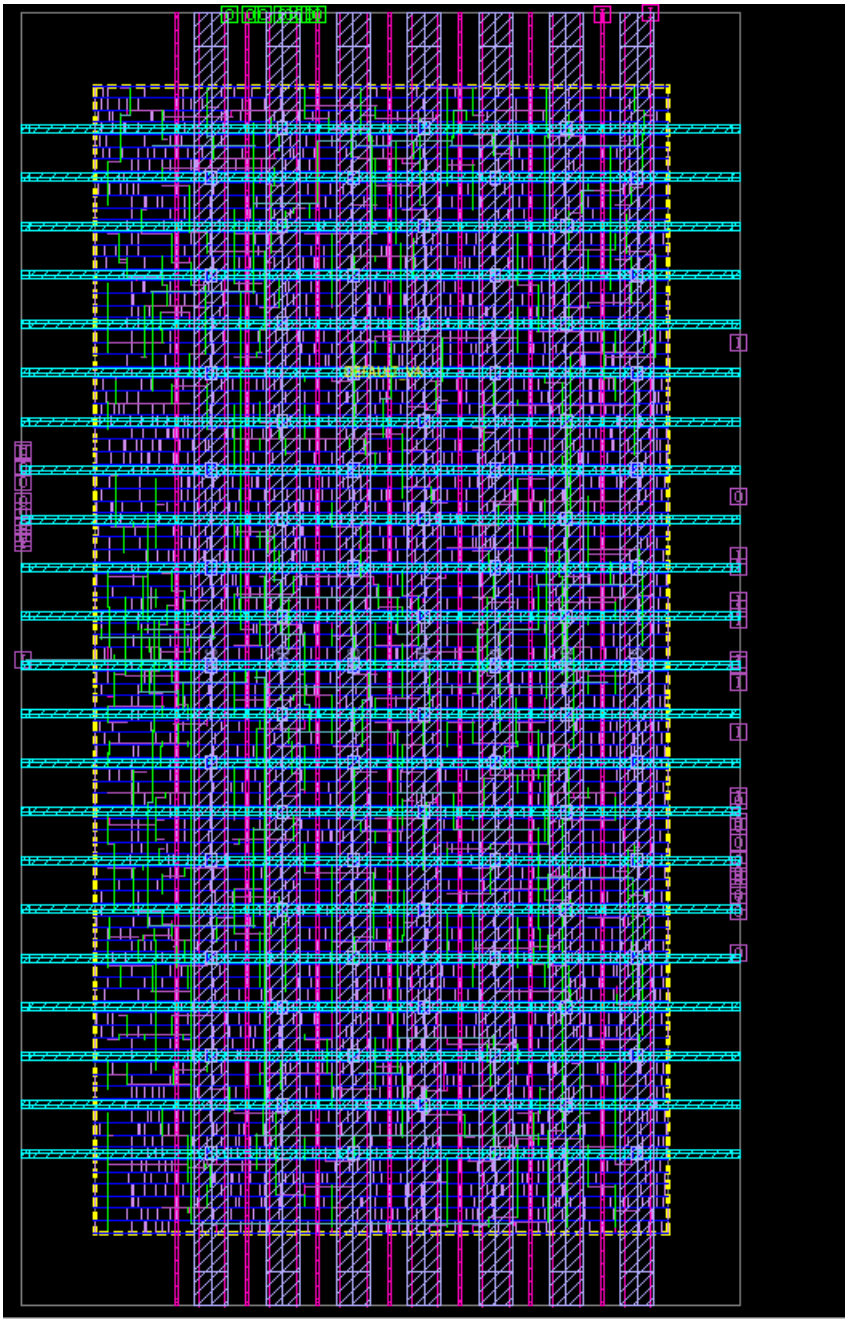
SL PD/Fusion_compiler_labs/FC_LABS/finalproj/run.tcl

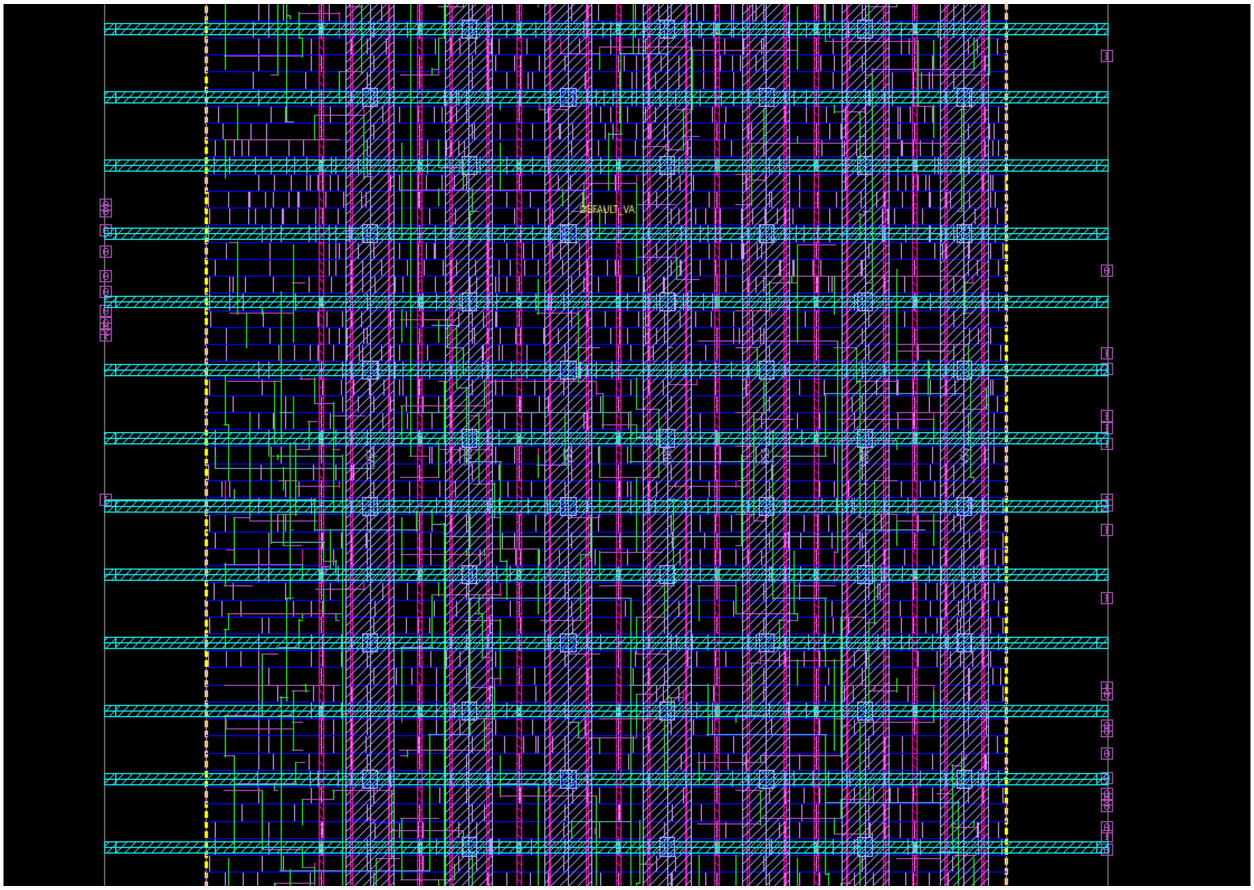
CTS

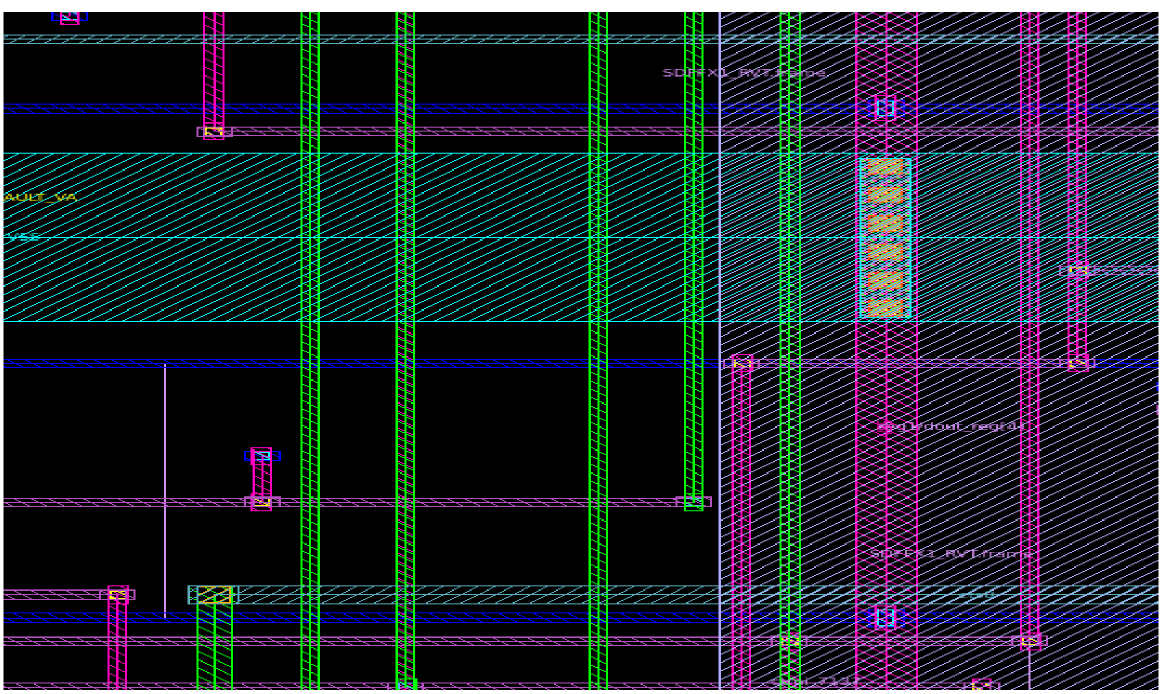
Define CTS Non-Default Routing Rules

we will specify CTS non-default routing rules, as well as clock cell spacing rules. ⟩ Open the file scripts/ndr.tcl in an editor

the green color in below image is clock tree



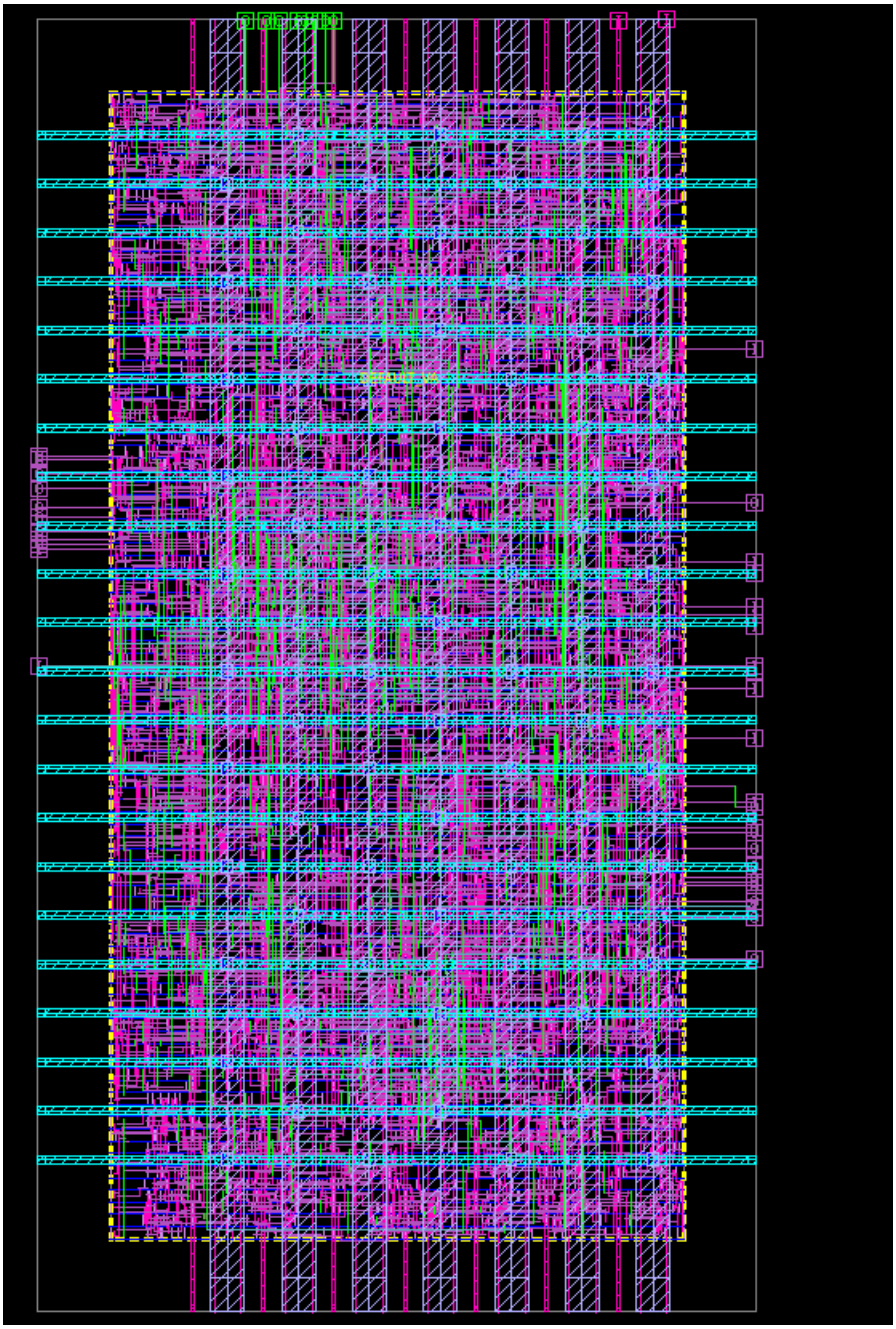

DEFAULT_VA

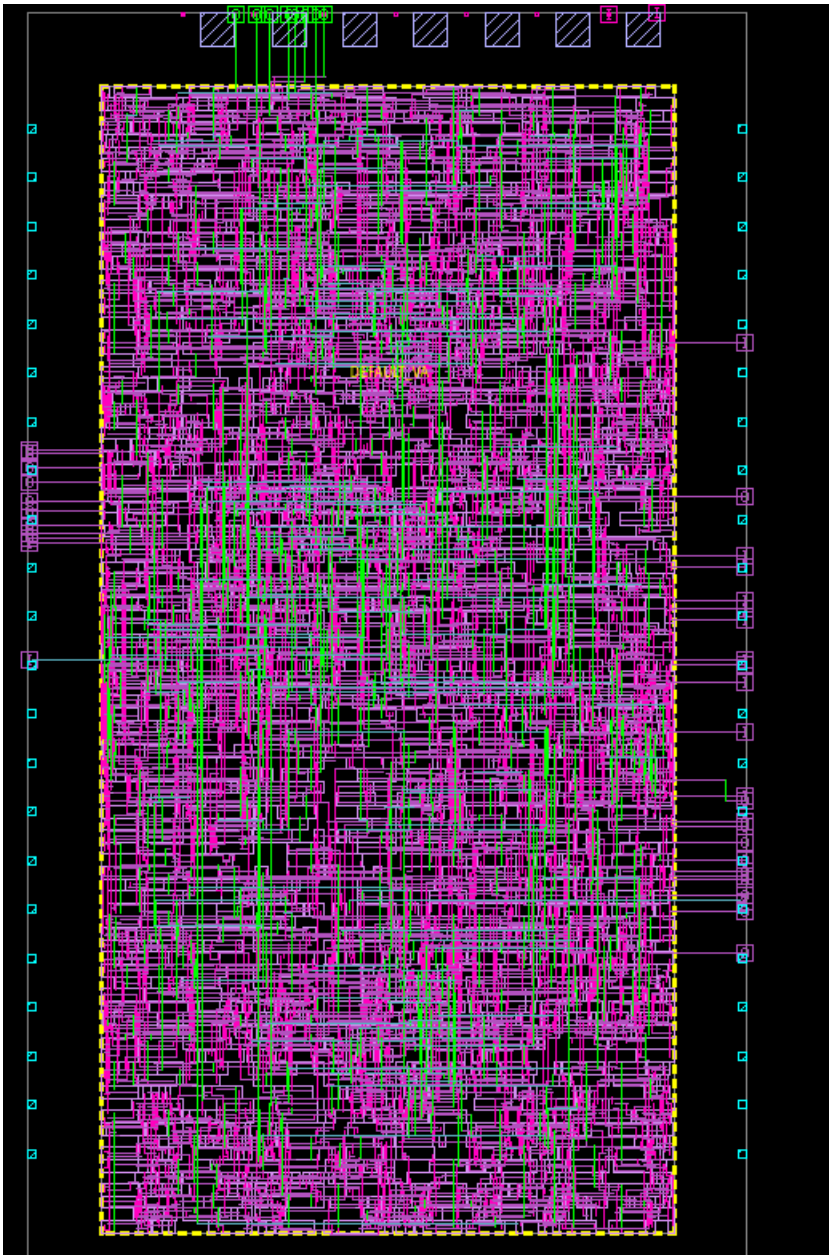

AULT_VA
VSS

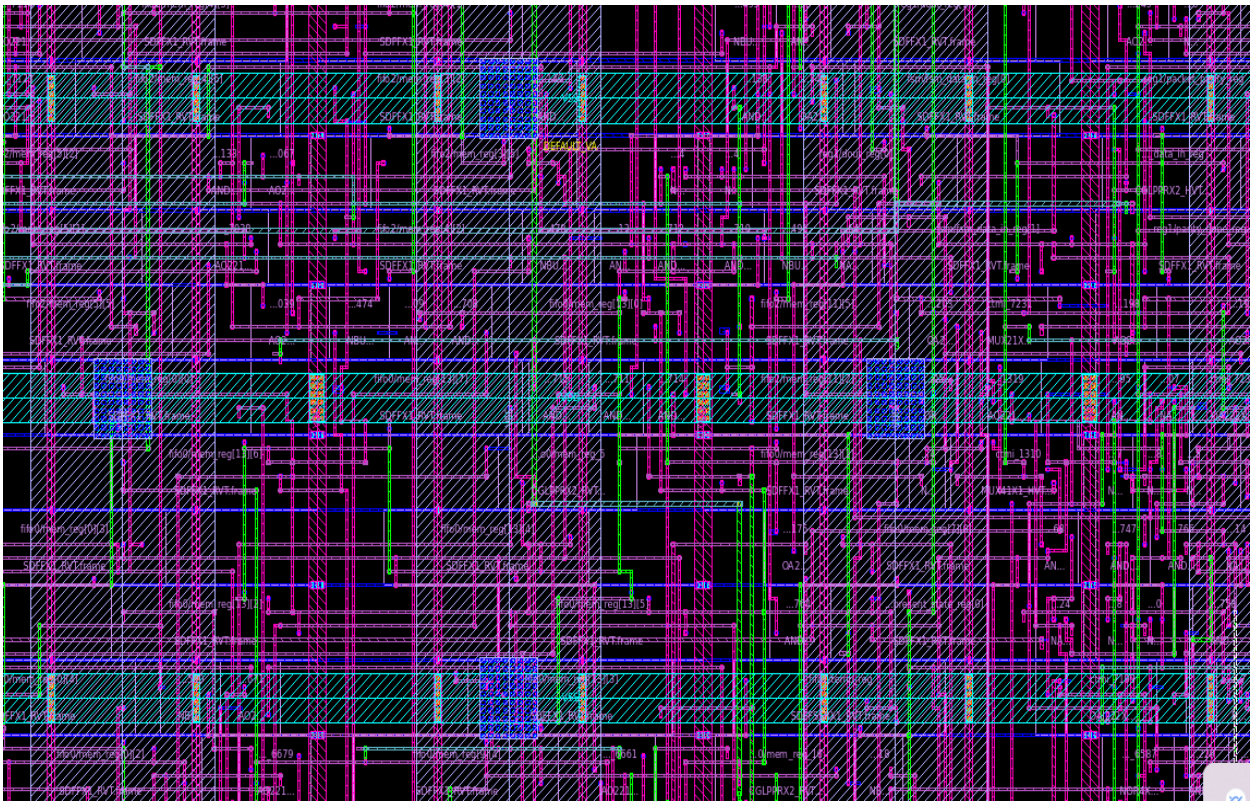
Routing and Post-Route Optimization

The routing is done by using below command

- route_auto







The RUN script used for our design

```
source ./rm_setup/fc_setup.tcl


file delete -force $DESIGN_LIBRARY
create_lib -technology $TECH_FILE -ref_libs $REFERENCE_LIBRARY $DESIGN_LIBRARY

analyze -format verilog [glob RTL/*.v]
elaborate router_top
set_top_module router_top

source scripts/tech_setup.tcl

load_upf ./design_data/router_top.upf
commit_upf

#source ./design_data/router_constraints.tcl

set_qor_strategy -stage synthesis -metric total_power
set_app_options -name opt.power.mode -value total

set_dont_touch [get_lib_cells */TIE*] false
set_lib_cell_purpose -include optimization [get_lib_cells */TIE*]

set_auto_floorplan_constraints -core_utilization 0.7 -side_ratio {1 2} -core_offset 10
#create_keepout_margin -type hard_macro -outer {3 3 3 3} [get_cells -hierarchical -filter is_hard_macro]
set_block_pin_constraints -self -allowed_layers "M3 M4 M5 M6"

source scripts/cts_setup.tcl

#source scripts/router_dft.tcl


####COMPILE STAGE#######
compile_fusion -from initial_map -to initial_map
report_unloaded_registers
report_clock_gating

#logic pto
compile_fusion -from logic_opto -to logic_opto
create_test_protocol
dft_drc -v
set_app_option -name dft.insertion_post_logic_opto -value true
insert_dft

#initail place
compile_fusion -from initial_place -to initial_place
all_high_transitive_fanout -nets -threshold 100
report_design
```

```
#drc
compile_fusion -from initial_drc -to initial_drc
all_high_transitive_fanout -nets -threshold 100
all_high_transitive_fanout -nets -threshold 40
get_scan_chain_count

#INITIAL_OPTO
compile_fusion -from initial_opto -to initial_opto
check_legality
report_optimization_history

###final place
compile_fusion -from final_place -to final_place
check_legality
report_power

####final opto
compile_fusion -from final_opto -to final_opto
report_qor -summary
#####
source scripts/mcmm/mcmm_router.tcl

write_floorplan
source scripts/pns.tcl

#################
report_clock_qor -type structure
report_clock_tree_options

derive_clock_cell_references -output cts_leq_set.tcl

set CTS_CELLS [get_lib_cells "*/NBUFF*LVT */NBUFF*RVT */INVX*_LVT */INVX*_RVT */CGL* */LSUP* */*DFF*"]
set_dont_touch $CTS_CELLS false
set_lib_cell_purpose -exclude cts [get_lib_cells]
set_lib_cell_purpose -include cts $CTS_CELLS

source scripts/cts_include_refs.tcl
report_lib_cells -objects [get_lib_cells] -columns {name:20 valid_purposes dont_touch}


source -echo scripts/ndr.tcl
report_routing_rules -verbose
report_clock_routing_rules
report_ports -verbose [get_ports *clk]
set_driving_cell -scenarios [all scenarios] \
-lib_cell NBUFFX16_RVT [get_ports ate_clk]
report_ports -verbose [get_ports *clk]
```

```
current_scenario $scen
set_clock_uncertainty 0.1 -setup [all_clocks]
set_clock_ uncertainty 0.05 -hold [all_ clocks]
}

report_clock_settings

set_app_options -name clock_opt.flow.enable_ccd \
-value false
clock_opt -to route_clock
########################################

report_qor -summary
current_mode
report_clocks
report_clocks -skew

report_clocks -groups
source scripts/cts_ex_ndr.tcl
get_scenarios -filter active&&hold
report_scenarios



##################### ROUTING   ###################################



 Synopsys Customer Education Services
# Fusion Compiler
#
# Run script for Routing and Post-routing Optimization Lab
#


# Design should have been loaded using load.tcl

report_qor -summary


#      check for any issues that might cause problems during routing
#      the app option allows for more detailed reporting
set_app_options -name route.common.verbose_level -value 1
check_design -checks pre_route_stage
set_app_options -name route.common.verbose_level -value 0

# Examine the EMS messages using the GUI
# Please ignore the NEX-048 Error
```

```
#      Antenna
source -echo ../ref/tech/saed32nm_ant_1p9m.tcl
report_app_options route.detail.*antenna*


#      Set application options for track and detail routing
set_app_options -name route.track.timing_driven     -value true
set_app_options -name route.track.crosstalk_driven  -value true
set_app_options -name route.detail.timing_driven    -value true
set_app_options -name route.detail.force_max_number_iterations -value false

#      Since we enabled timing driven routing, we enable SI as well so the router knows about the delta delays
#      ... or we don't in order to make this lab more exciting towards the end...
# set_app_options -name time.si_enable_analysis -value true


## Secondary PG Routing : Not applicable for New GRE flow
## This would have already occurred dring clock_opt -from final_opto
####################################

#      Check the power supplies
report_power_domains

#   The following lines were already done prior to clock_opt final_opto
# set_app_options -name route.common.number_of_secondary_pg_pin_connections -value 2
# set_app_options -name route.common.separate_tie_off_from_secondary_pg     -value true
# if {[get_routing_rules -quiet VDDwide] != ""} {remove_routing_rules VDDwide }
# create_routing_rule VDDwide -widths {M1 0.1 M2 0.1 M3 0.1} -taper_distance 0.2
# set_routing_rule -rule VDDwide -min_routing_layer M2 -min_layer_mode allow_pin_connection -max_routing_layer M3 [get_nets VDD]
# route_group -nets {VDD}

# Have a look at the secondary PG routing for the level shifters:
change_selection [get_cells -hierarchical -filter is_level_shifter&&full_name=~*RISC*]


## Routing
####################################
route_auto

check_routes
# GUI DRC analysis

#      DRC errors?
# route_detail -incremental true -initial_drc_from_input true
#      Shorts? can't be fixed? Try deleting shapes, then run:
# route_eco -utilize_dangling_wires true -open_net_driven true
```

```
# Configure StarRC extraction
set_starrc_in_design -config ./scripts/starrc_config.txt

report_qor -summary
set_app_options -name time.si_enable_analysis -value true
set_app_options -name time.enable_ccs_rcv_cap -value true
# Will only work with CCS libraries:
set_app_options -name time.delay_calc_waveform_analysis_mode -value full_design
# set_app_options -name time.awp_compatibility_mode -value false   ;# false by default

report_qor -summary
report_qor -summary -pba_mode path



## Post-Route Optimization
##################################

#      power optimization, CCD, CTO are controlled via app options.
# report_app_options route_opt.*

# Note: For this lab, leave these disabled. Not required, and will lead to longer runtimes...
# set_app_options -name route_opt.flow.enable_ccd -value true
# set_app_options -name ccd.post_route_buffer_removal -value true

route_opt
report_qor -summary
## To disable soft-rule-based timing optimization during ECO routing, uncomment the following.
#  This is to limit spreading which can touch multiple routes and impact convergence.
#set_app_options -name route.detail.eco_route_use_soft_spacing_for_timing_optimization -value false
set_app_options -name route_opt.flow.enable_ccd -value false

# For more accuracy, you could switch to PBA now.
# For this lab, it's not necessary
# set_app_options -name time.pba_optimization_mode -value path

route_opt
report_qor -summary -pba_mode path

##

rename_block -to_block ORCA_TOP/route_opt
save_lib
~
```

After completion of routing the content in project folder are