Contents

Introduction and Objectives 3

Project Workflow 3

Synthesis 3

Placement and Routing 8

Static timing analysis 16

Buffer tree synthesis 21

Verification 24

Conclusion and Problems 26

Introduction and Objectives

The assigned project is based on a Verilog code of a processor chip with 9 sub modules including an ALU capable of 4 operations (buffer, addition, AND, OR) between two 8bit inputs. The code is then synthesized into a Gate level Netlist using Synthesis tool (Genus). Then the files obtained from Genus (.sdc and .v) along with other necessary files are further used to design (floorplan and route) the chip with the help of another tool (Encounter). The final design should be free of all violations.

Project Workflow

Synthesis

Before synthesis different directories were created to place the files on specific directories. The provided Verilog file (test_pro.v) was added to the directory by creating a .v file and then copying the code. The following commands were given in the command line of the server.

```
mkdir project
cd project
mkdir timing_libs
cp /Project_Setup_PnR/EDI_files/libs/* timing_libs/
cp /Project_Setup_PnR/EDI_files/lef/gsclib045.lef lef/
cp /Project_Setup_PnR/EDI_files/capTable/capTable others/
```

The provided SDC constraints for synthesis are

1. Clock frequency= 413 MHz (So, Time period = 2.42 ns)
2. Uncertainty for setup “0.35” & hold is “0.3”.
3. Maximum transition “2” [overall design]
4. Clock transition “min” “0.3” & “max” “0.3” for both “fall”l and “rise”
5. Driving cell “BUFX2”
6. Operating conditions “slow”
7. Min input & output delay is “0” and the maximum is “0.5”
8. Output load “0.5”
9. Max Fanout is “10”

Considering the above constraints, the following commands were given to the “genus” tool command window:

```
set_db init_lib_search_path ./timing_libs
set_db library slow.lib
set_db lef_library ./lef/gsclib045.lef
set_db hdl_search_path input_files/
read_hdl test_pro.v
```

```
elaborate
set_top_module proccessor
write_hdl>elaborated.v
create_clock -name clk -period 2.42 [get_ports clk]
report_clocks
syn_generic
syn_map
write_hdl>mapped.v
set verilogout_no_tri true
set verilogout_unconnected_prefix "UNCONNECTED"
set verilogout_show_unconnected pins true
set verilogout_single_bit false
set edifout_netlist_only true
set_clock_uncertainty -setup 0.35 [get_clocks clk]
report_clocks
set_clock_uncertainty -hold 0.3 [get_clocks clk]
set_max_transition 2 [current_design]
set_max_transition -min -fall 0.3 [get_clocks clk]
set_max_transition -min -fall 0.3 [get_clocks clk]
set_max_transition -min -rise 0.3 [get_clocks clk]
set_max_transition -max -rise 0.3 [get_clocks clk]
set_clock_transition -min -rise 0.3 [get_clocks clk]
set_clock_transition -min -fall 0.3 [get_clocks clk]
set_clock_transition -max -fall 0.3 [get_clocks clk]
set_clock_transition -max -rise 0.3 [get_clocks clk]
set_clock_groups -name original -group [list [get_clocks clk]]
set DRIVING_CELL BUFX2
set DRIVE_PIN {Y}
set_driving_cell -lib_cell BUFX2 -pin Y [all_inputs]
set_max_fanout 10 [current_design]
set_operating_conditions slow
set MAX_INPUT_DELAY 0.5
set MIN_INPUT_DELAY 0
set MIN_OUTPUT_DELAY 0
set MAX_OUTPUT_DELAY 0.5
set_load 0.5 [all_outputs]
set_operating_conditions slow
```

```
set_input_delay -max $MAX_INPUT_DELAY [all_inputs]
set_output_delay -max $MAX_OUTPUT_DELAY [all_inputs]
set_output_delay -max $MAX_OUTPUT_DELAY [all_outputs]
set_input_delay -max $MAX_INPUT_DELAY [all_inputs]
remove_assign -buffer_or_inverter BUFX12 -design processor
synthesize -to_mapped
remove_assigns_without_opt -buffer_or_inverter BUFX12 -verbose
set_remove_assign_options -buffer_or_inverter BUFX12 -verbose
write -mapped > processor_synth.v
write_sdc > processor.sdc
exit
```

Therefore, a Gate Level Netlist (processor_syth.v) and a Standard Design Constraint file (processor.sdc) were generated which will then be used for Placement and Routing.

The final SDC generated is shown on the following pages.

```
# ####################################################################

#  Created by Genus Synthesis Solution 15.10-s019_1 on Tue Nov 17 12:31:51 +0600 2020

# ####################################################################

set sdc_version 1.7

set_units -capacitance 1000.0fF
set_units -time 1000.0ps

# Set the current design
current_design proccessor

create_clock -name "clk" -add -period 2.42 -waveform {0.0 1.21} [get_ports clk]
set_clock_transition 0.3 [get_clocks clk]
set_load -pin_load 0.5 [get_ports {result[7]}]
set_load -pin_load 0.5 [get_ports {result[6]}]
set_load -pin_load 0.5 [get_ports {result[5]}]
set_load -pin_load 0.5 [get_ports {result[4]}]
set_load -pin_load 0.5 [get_ports {result[3]}]
set_load -pin_load 0.5 [get_ports {result[2]}]
set_load -pin_load 0.5 [get_ports {result[1]}]
set_load -pin_load 0.5 [get_ports {result[0]}]
set_load -pin_load 0.5 [get_ports {OUT1[7]}]
set_load -pin_load 0.5 [get_ports {OUT1[6]}]
set_load -pin_load 0.5 [get_ports {OUT1[5]}]
set_load -pin_load 0.5 [get_ports {OUT1[4]}]
set_load -pin_load 0.5 [get_ports {OUT1[3]}]
set_load -pin_load 0.5 [get_ports {OUT1[2]}]
set_load -pin_load 0.5 [get_ports {OUT1[1]}]
set_load -pin_load 0.5 [get_ports {OUT1[0]}]
set_load -pin_load 0.5 [get_ports {OUT2[7]}]
set_load -pin_load 0.5 [get_ports {OUT2[6]}]
set_load -pin_load 0.5 [get_ports {OUT2[5]}]
set_load -pin_load 0.5 [get_ports {OUT2[4]}]
set_load -pin_load 0.5 [get_ports {OUT2[3]}]
set_load -pin_load 0.5 [get_ports {OUT2[2]}]
set_load -pin_load 0.5 [get_ports {OUT2[1]}]
set_load -pin_load 0.5 [get_ports {OUT2[0]}]
set_load -pin_load 0.5 [get_ports {Data2[7]}]
set_load -pin_load 0.5 [get_ports {Data2[6]}]
set_load -pin_load 0.5 [get_ports {Data2[5]}]
set_load -pin_load 0.5 [get_ports {Data2[4]}]
set_load -pin_load 0.5 [get_ports {Data2[3]}]
set_load -pin_load 0.5 [get_ports {Data2[2]}]
set_load -pin_load 0.5 [get_ports {Data2[1]}]
set_load -pin_load 0.5 [get_ports {Data2[0]}]
set_load -pin_load 0.5 [get_ports {mux1out[7]}]
set_load -pin_load 0.5 [get_ports {mux1out[6]}]
set_load -pin_load 0.5 [get_ports {mux1out[5]}]
set_load -pin_load 0.5 [get_ports {mux1out[4]}]
set_load -pin_load 0.5 [get_ports {mux1out[3]}]
set_load -pin_load 0.5 [get_ports {mux1out[2]}]
set_load -pin_load 0.5 [get_ports {mux1out[1]}]
set_load -pin_load 0.5 [get_ports {mux1out[0]}]
```

```
set_load -pin_load 0.5 [get_ports {immediate[7]}]
set_load -pin_load 0.5 [get_ports {immediate[6]}]
set_load -pin_load 0.5 [get_ports {immediate[5]}]
set_load -pin_load 0.5 [get_ports {immediate[4]}]
set_load -pin_load 0.5 [get_ports {immediate[3]}]
set_load -pin_load 0.5 [get_ports {immediate[2]}]
set_load -pin_load 0.5 [get_ports {immediate[1]}]
set_load -pin_load 0.5 [get_ports {immediate[0]}]
set_load -pin_load 0.5 [get_ports {mux2out[7]}]
set_load -pin_load 0.5 [get_ports {mux2out[6]}]
set_load -pin_load 0.5 [get_ports {mux2out[5]}]
set_load -pin_load 0.5 [get_ports {mux2out[4]}]
set_load -pin_load 0.5 [get_ports {mux2out[3]}]
set_load -pin_load 0.5 [get_ports {mux2out[2]}]
set_load -pin_load 0.5 [get_ports {mux2out[1]}]
set_load -pin_load 0.5 [get_ports {mux2out[0]}]
set_load -pin_load 0.5 [get_ports {Select[2]}]
set_load -pin_load 0.5 [get_ports {Select[1]}]
set_load -pin_load 0.5 [get_ports {Select[0]}]
set_clock_groups -name original        -group [get_clocks clk]
set_clock_gating_check -setup 0.0
set_input_delay -clock [get_clocks clk] -network_latency_included -add_delay -rise -min 0.0 [get_ports clk]
set_input_delay -clock [get_clocks clk] -clock_fall -network_latency_included -add_delay -fall -min 0.0 [get_ports clk]
set_output_delay -add_delay -max 0.5 [get_ports {result[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {result[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT1[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {OUT2[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {Data2[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[7]}]


set_output_delay -add_delay -max 0.5 [get_ports {mux1out[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux1out[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {immediate[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[7]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[6]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[5]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[4]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[3]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {mux2out[0]}]
set_output_delay -add_delay -max 0.5 [get_ports {Select[2]}]
set_output_delay -add_delay -max 0.5 [get_ports {Select[1]}]
set_output_delay -add_delay -max 0.5 [get_ports {Select[0]}]
set_input_delay -add_delay -max 0.5 [get_ports clk]
set_input_delay -add_delay -max 0.5 [get_ports reset]
set_max_fanout 10.000 [current_design]
set_max_transition 2.0 [current_design]
set_operating_conditions slow
set_dont_use [get_lib_cells gpdk045bc/HOLDX1]
set_clock_uncertainty -setup 0.35 [get_clocks clk]
set_clock_uncertainty -hold 0.3 [get_clocks clk]
```

Placement and Routing

Step 1 (Design and constrains import):

Before PnR different directories were created to place the files on specific directories. Also the Gate Level Netlist (processor_syth.v) and a Standard Design Constraint file (processor.sdc) were placed in a specific directory (input).

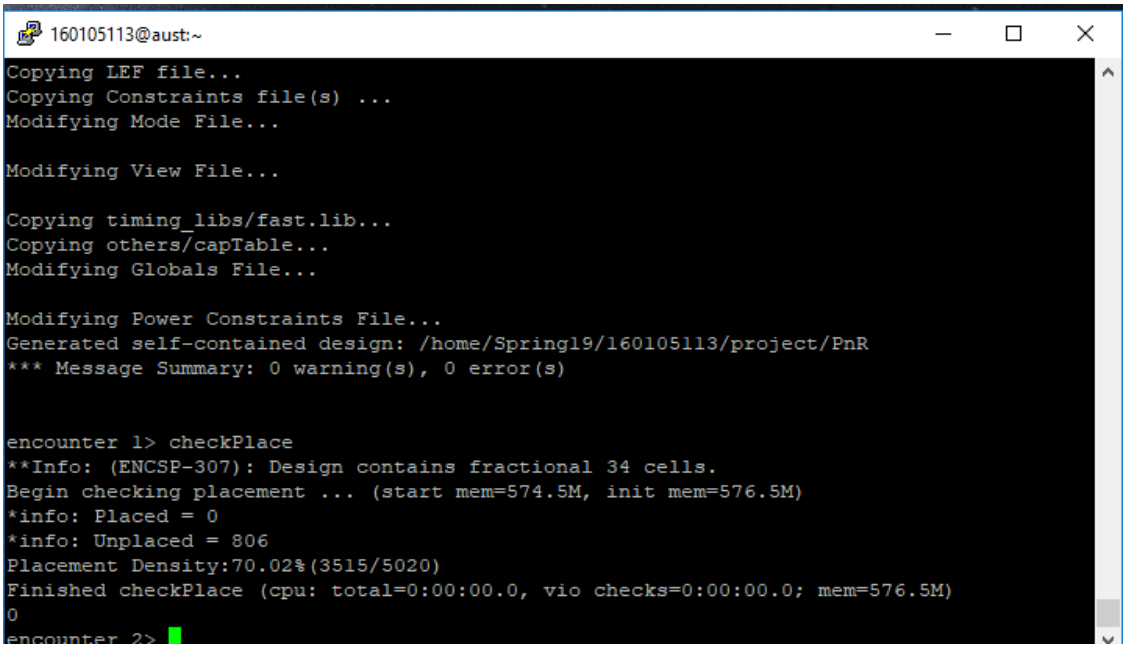
```
cd project

mkdir PnR

cd PnR

mkdir input DESIGN ver reports scripts timing_libs lef others

cp /project/processor_syth.v ./input/

cp /project/processor.sdc ./input/

cp /Project_Setup_PnR/EDI_files/libs/* ./timing_libs/

cp /Project_Setup_PnR/EDI_files/lef/gsclib045.lef ./lef/

cp /Project_Setup_PnR/EDI_files/capTable/capTable ./others/
```

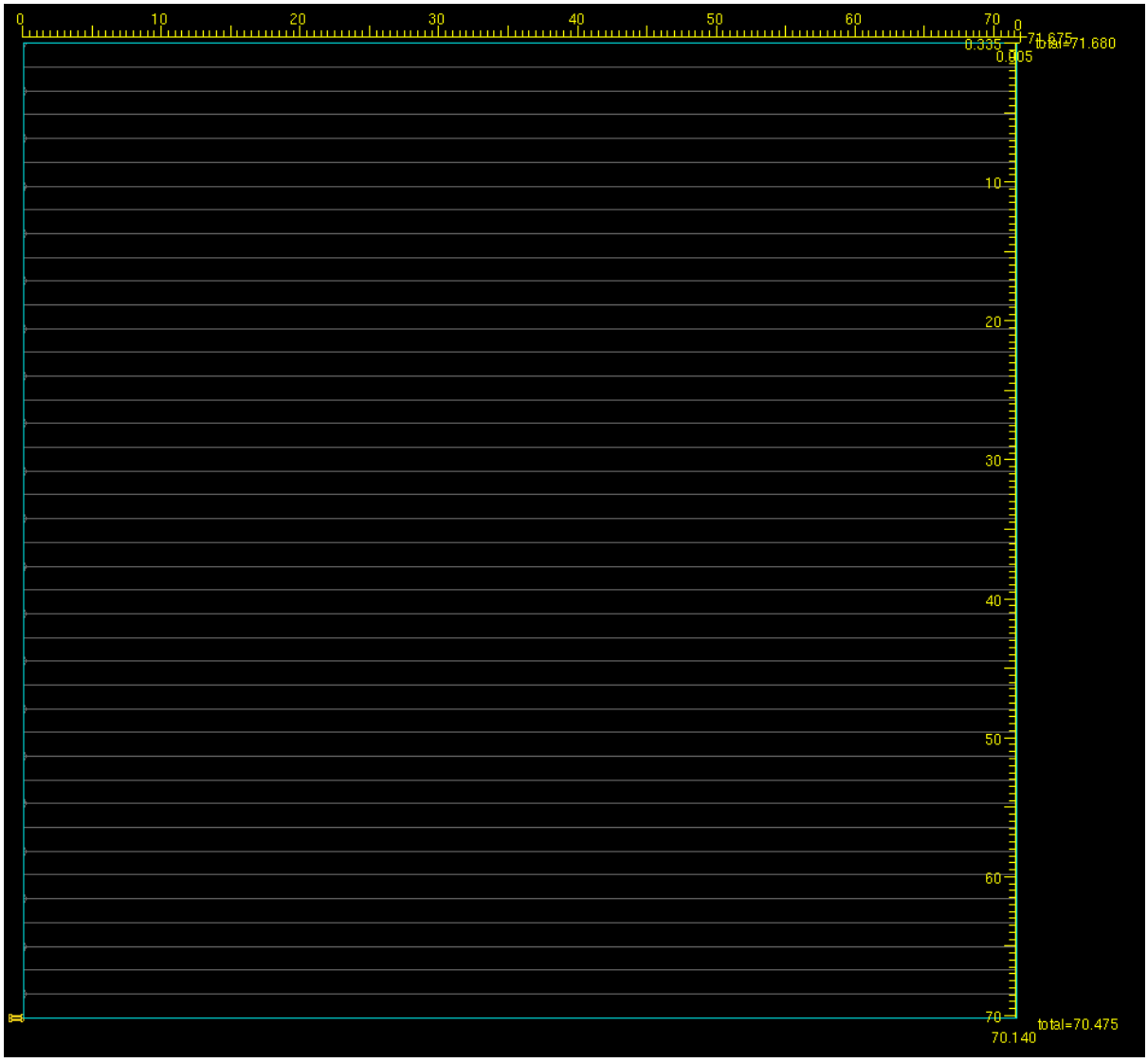
After that the “Encounter” tool was opened and the “processor_syth.v” was imported to the tool and the lef files and the necessary files are imported. As this is a block level design, the power net is named VDD and ground net is named VSS.

In the MMC Browser the MMC objects were set. The log is as follows

```
create_rc_corner -name rc_typical -cap_table {PnR/others/capTable} -preRoute_res {1.0} -preRoute_cap {1.0} -preRoute_clkres {0.0} -preRoute_clkcap {0.0} -postRoute_res {1.0} -postRoute_cap {1.0} -postRoute_xcap {1.0} -postRoute_clkres {0.0} -postRoute_clkcap {0.0}

create_library_set -name max_timing -timing {PnR/timing_libs/slow.lib}

create_library_set -name min_timing -timing {PnR/timing_libs/fast.lib}

create_constraint_mode -name functional_sdc -sdc_files {PnR/input/processor.sdc}

create_delay_corner -name max_delay -library_set {max_timing} -rc_corner {rc_typical}

create_delay_corner -name min_delay -library_set {min_timing} -rc_corner {rc_typical}

create_analysis_view -name func_slow -constraint_mode {functional_sdc} -delay_corner {max_delay}

create_analysis_view -name func_fast -constraint_mode {functional_sdc} -delay_corner {min_delay}

set_analysis_view -setup {func_slow} -hold {func_fast}
```

It was specified that the placement density (after import of the design) should be greater than or equal to 70%. Here the placement density of 70.02%.


160105113@aust:~
Copying LEF file...
Copying Constraints file(s) ...
Modifying Mode File...
Modifying View File...
Copying timing_libs/fast.lib...
Copying others/capTable...
Modifying Globals File...
Modifying Power Constraints File...
Generated self-contained design: /home/Spring19/160105113/project/PnR
*** Message Summary: 0 warning(s), 0 error(s)
encounter 1> checkPlace
**Info: (ENCSP-307): Design contains fractional 34 cells.
Begin checking placement ... (start mem=574.5M, init mem=576.5M)
*info: Placed = 0
*info: Unplaced = 806
Placement Density:70.02%(3515/5020)
Finished checkPlace (cpu: total=0:00:00.0, vio checks=0:00:00.0; mem=576.5M)
0
encounter 2>

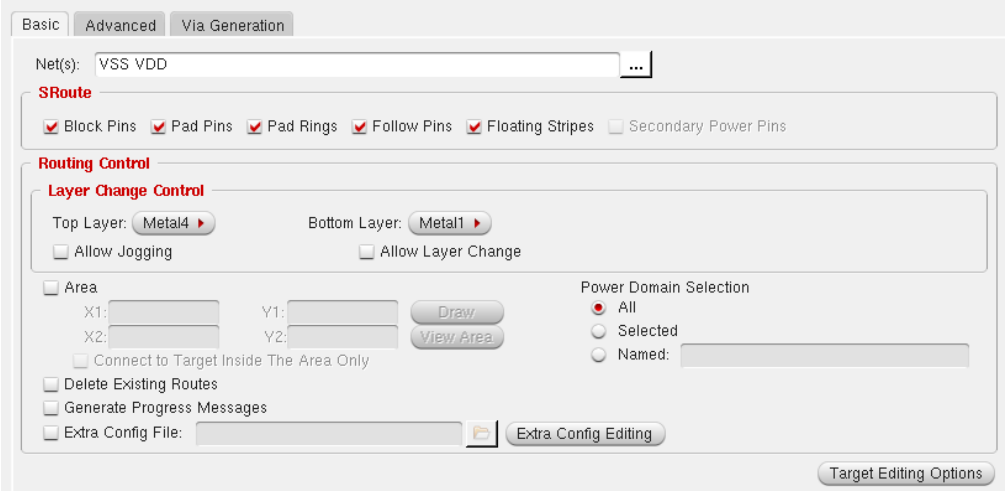

0
10
20
30
40
50
60
70
0.335
0.905
total=71.680
total=70.475
70.140

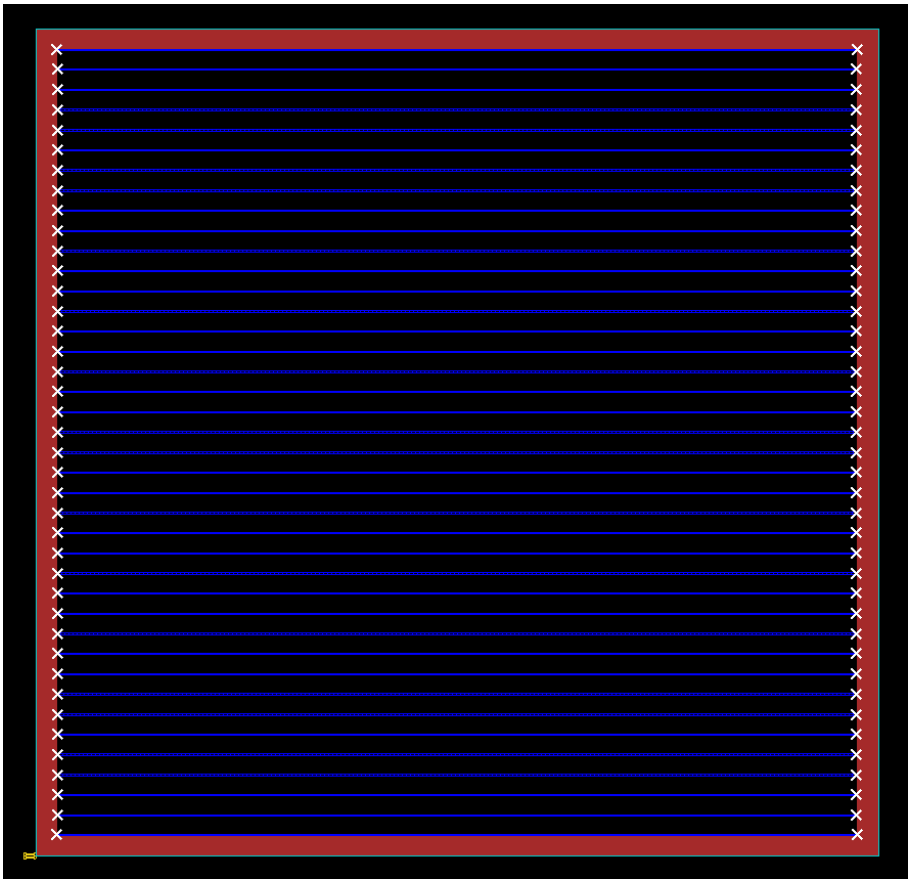
Step 2 (Floorplanning and placement blockage):

The Design Constraint specified the following:

1. Die boundary is not required.
2. Use placement blockage around the block.
3. Top Metal should be M4.

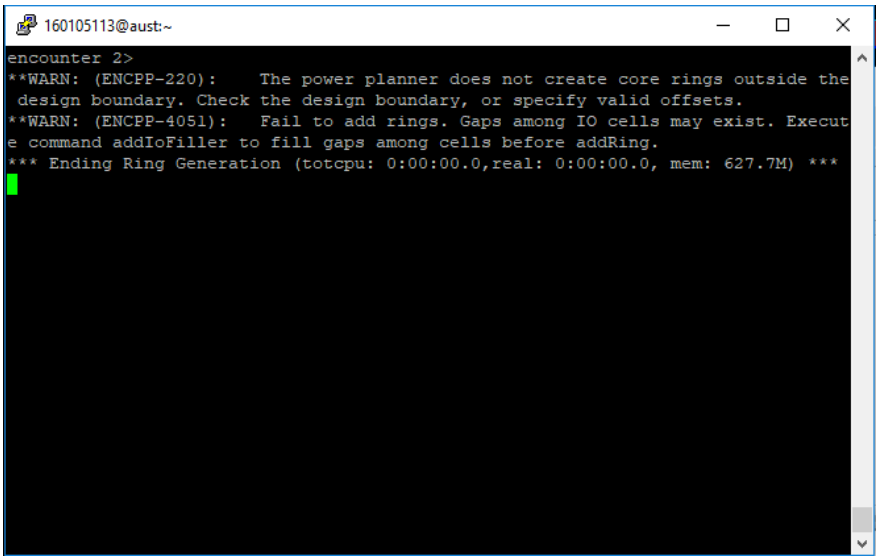
So, no die boundary was placed around the chip. Instead placement blockage is placed around it. After that the power planning is done using Special Route. The Top Layer is selected as Metal 4.



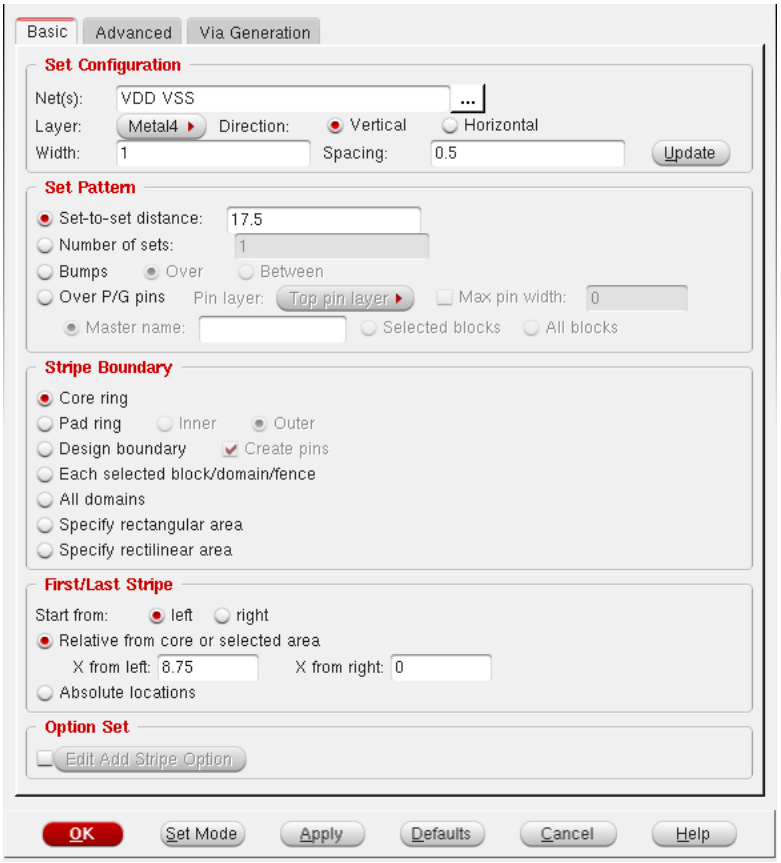


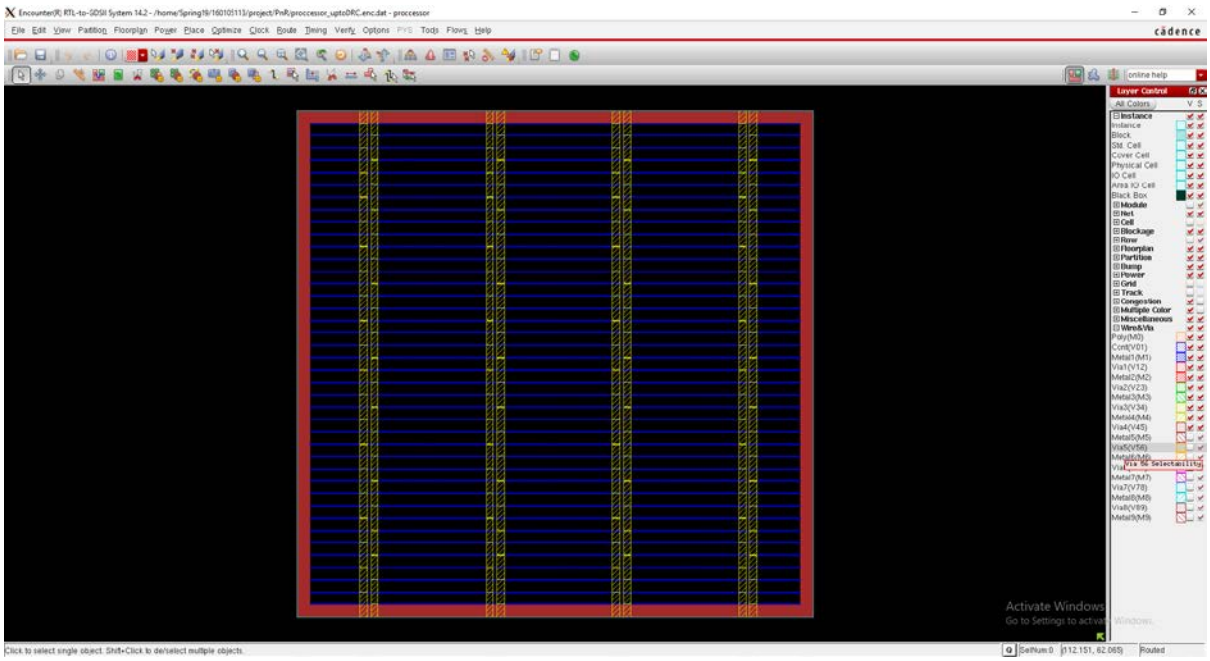
Step 3 (Power Planning):

For this design, power ring is not added as there is no core to die boundary present.



Therefore, power strips are added. Now the width is approximately 70, so the power strips are at set to set distances (17.5) and the first stripe begins at (17.5/2) or 8.75 from the left. Metal 4 is selected according to given specification.



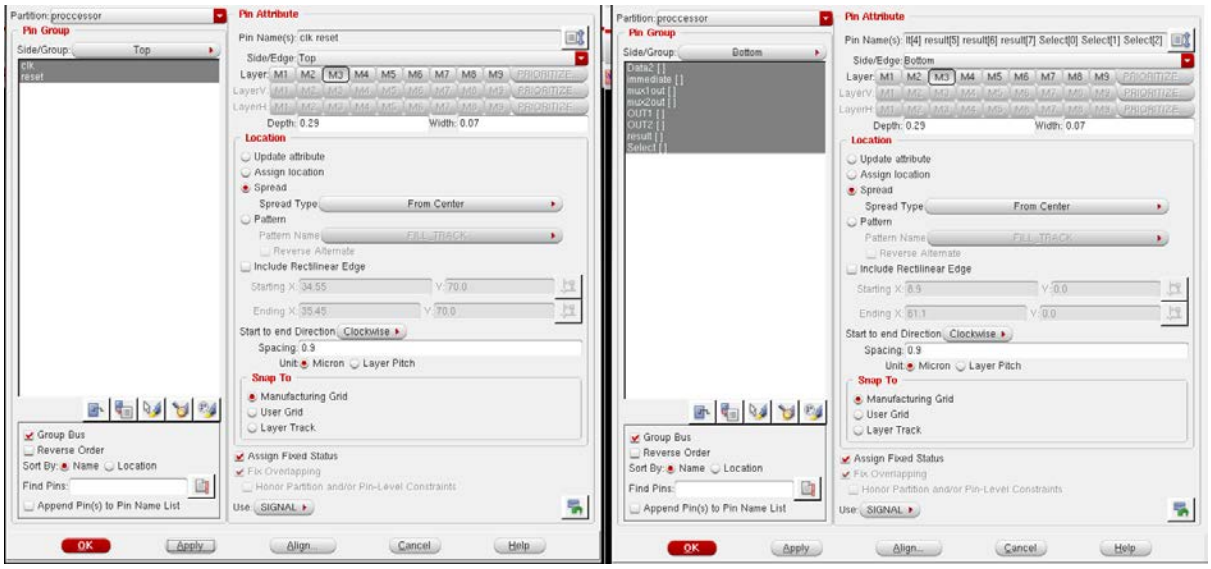


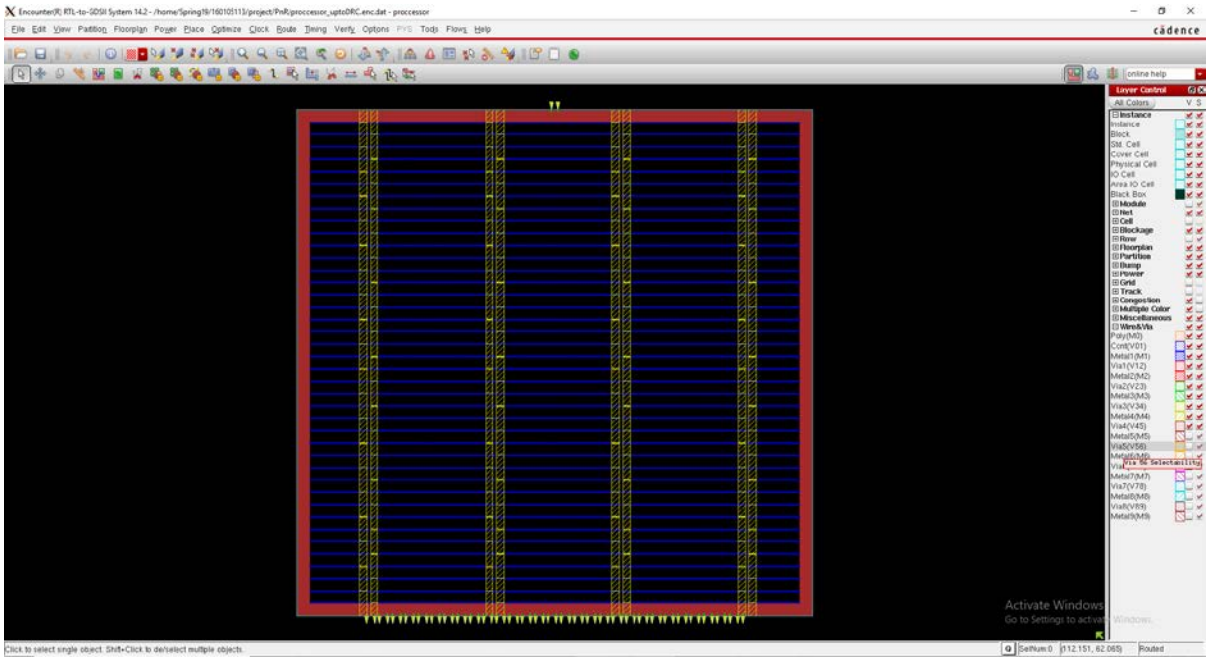
Step 4 (I/O pin Placement):

The input and output pins can be known from the provided Verilog code. The specification mentioned that the input pins should be placed at the “top” side and output pins at the “bottom” side. Therefore, they are needed to be specified and positioned separately. Metal 3 is used for the pins. Here the I/O pins are

input: reset, clk

output: Select, immediate, OUT2, Data2, mux1out, result, OUT1, mux2out

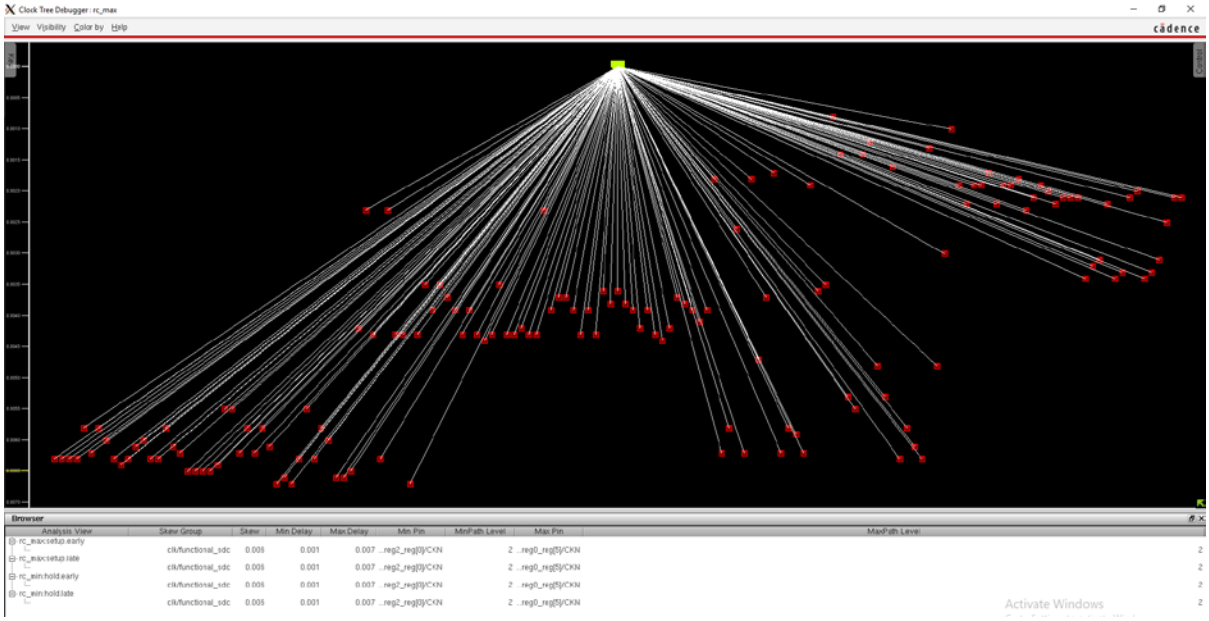




Step 5 (Standard cell placement and CTS):

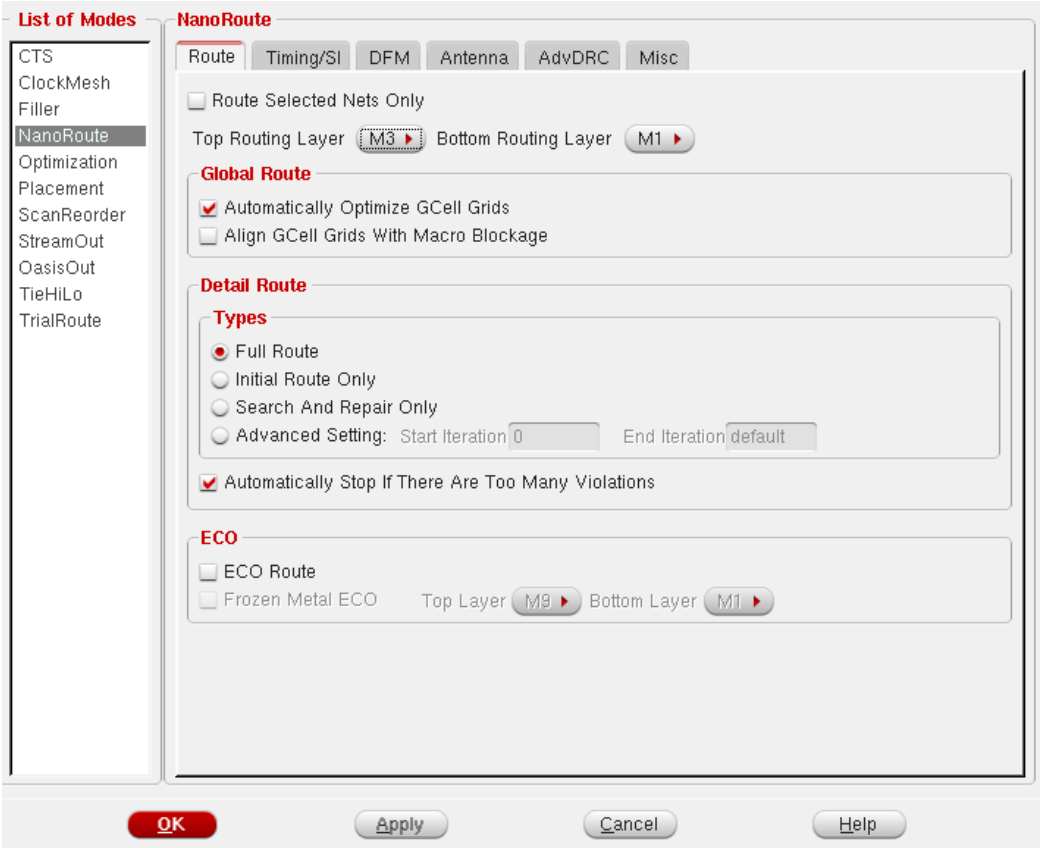
At this step, the Standard cells are placed. The tool does optimized placement of the cells. After that the Clock Tree Synthesis is done. This builds the Clock Tree.

A clock tree debugger window is generated like follows



Step 6 (Routing):

In this step the Routing is done. Since the specification states that the Maximum Routing Layer should be up to M3, the Top Routing layer is set to M3. Therefore, the tool routes up to Metal 3.



Step 7 (Global net connection of Power and Ground Nets):

In this step, the power and ground net connections are made in CIW mode. The following commands are given in CIW.

```
globalNetConnect VDD -pin VDD -inst * -verbose
globalNetConnect VSS -pin VSS -inst * -verbose
```

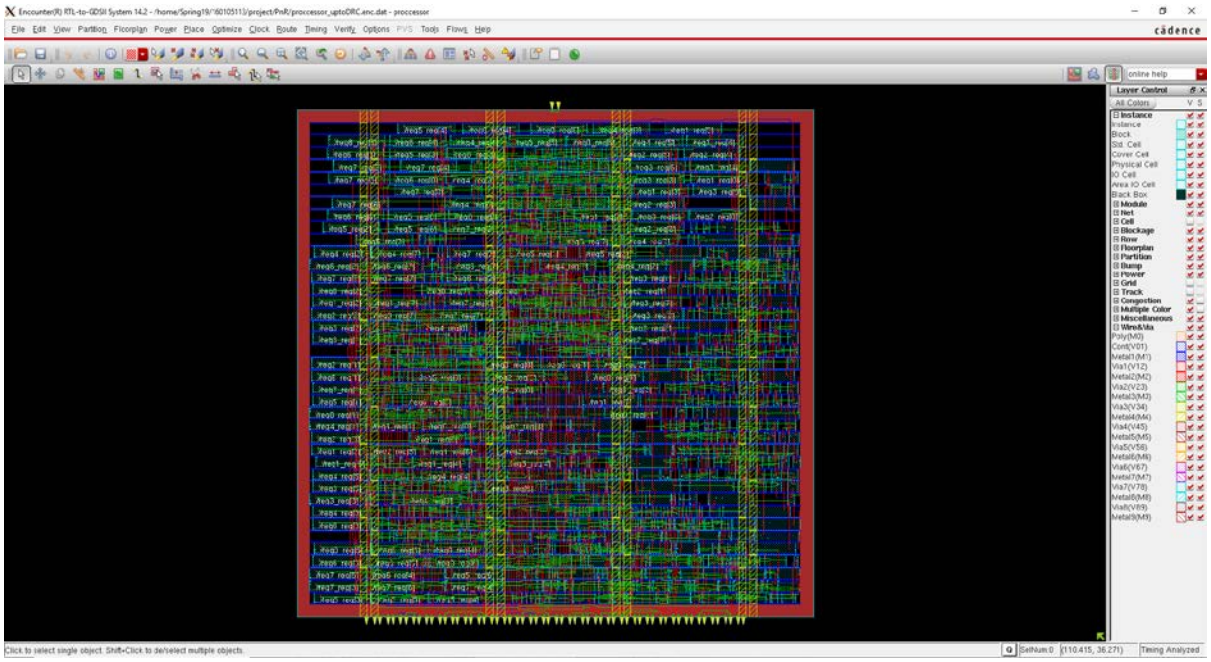


Fig: Layout after Standard cell Placement, CTS and Routing

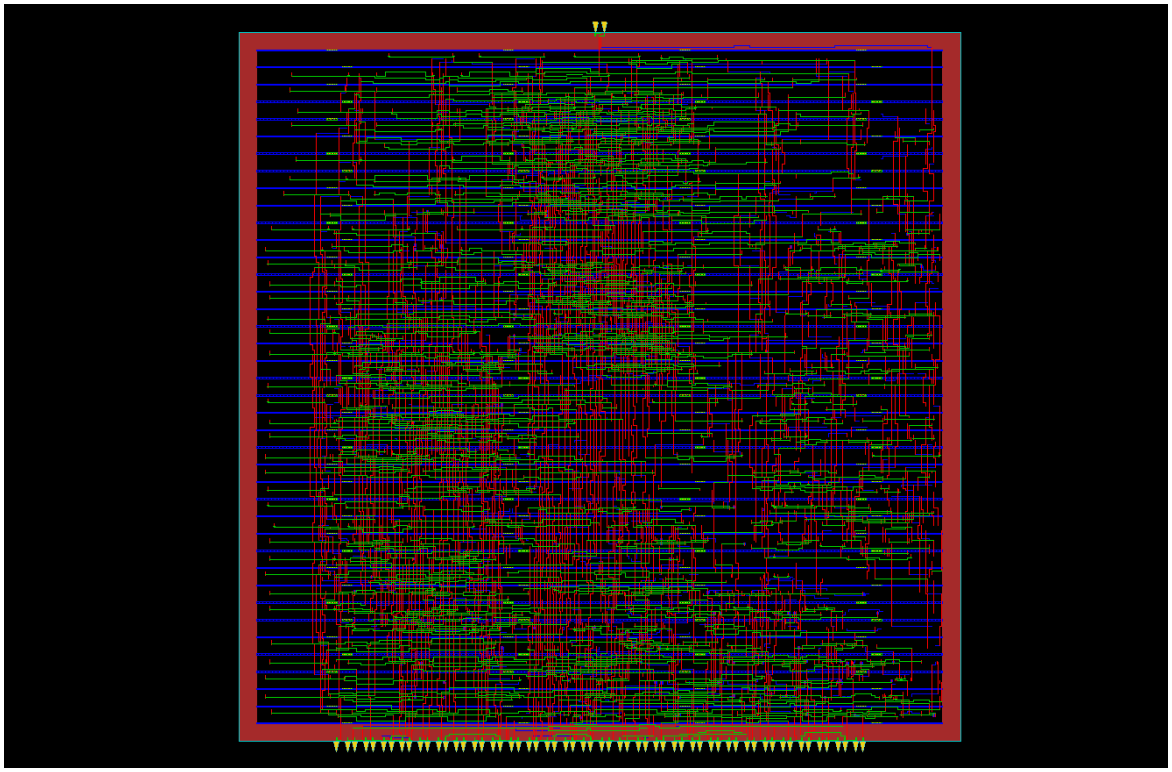


Fig: Maximum Routing up to M3

Static timing analysis

Step 1 (Determining the violations):

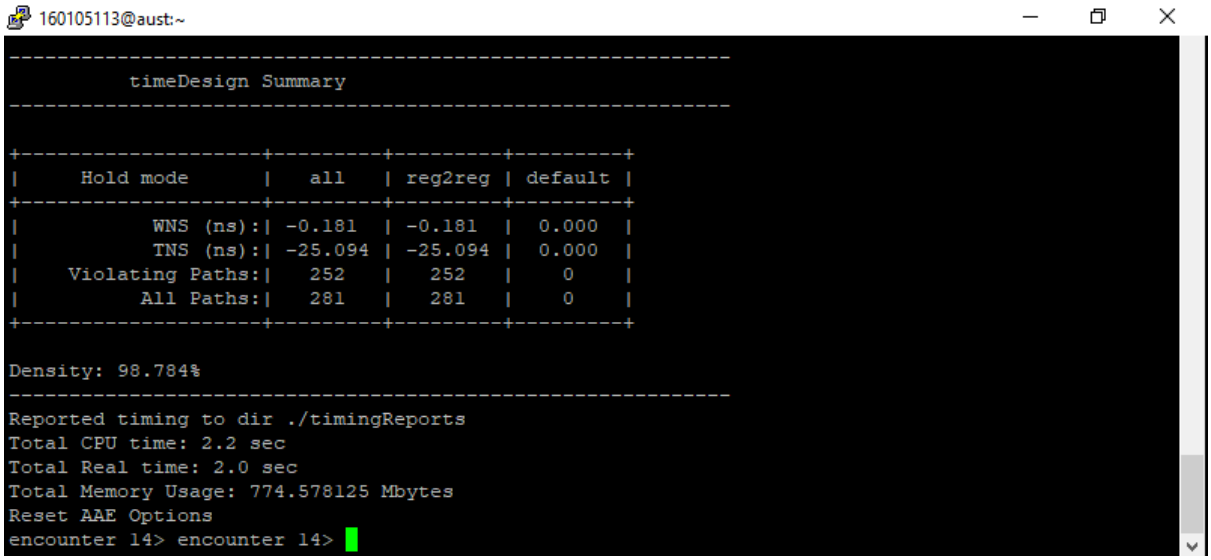
The specification mentioned to clean all the design rule violations and timing violations (setup & hold). Right after Routing is done the violations are checked in the CIW mode which gives the following result.

```
160105113@aust:~
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  0.243  |  0.471  |  0.243  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   281   |   281   |    3    |
+--------------------+---------+---------+---------+

+----------------+-----------------------------------+------------------+
|                |              Real                 |       Total      |
|    DRVs        +------------------+----------------+------------------|
|                |  Nr nets(terms)  | Worst Vio      |  Nr nets(terms)  |
+----------------+------------------+----------------+------------------+
|   max_cap      |      0 (0)       |   0.000        |      0 (0)       |
|   max_tran     |      1 (2)       |   -0.285       |      1 (2)       |
|   max_fanout   |     18 (18)      |    -12         |     19 (19)      |
|   max_length   |      0 (0)       |     0          |      0 (0)       |
+----------------+------------------+----------------+------------------+

Density: 98.784%
Total number of glitch violations: 0
------------------------------------------------------------
Reported timing to dir ./timingReports
Total CPU time: 1.77 sec
Total Real time: 2.0 sec
Total Memory Usage: 802.4375 Mbytes
Reset AAE Options
encounter 6> encounter 6>
```

The transition violation is as follows

```
160105113@aust:~
encounter 11> set_max_transition 2 [current_design]
encounter 12> report_constraint -drv_violation_type max_transition
#################################################################
#  Generated by:      Cadence Encounter 14.20-p004_1
#  OS:                Linux x86_64(Host ID aust)
#  Generated on:      Thu Nov 19 10:26:02 2020
#  Design:            proccessor
#  Command:           report_constraint -drv_violation_type max_transition
#################################################################
# format : frame 1 : split 1

Check type : max_transition
---------------------------
  Pin : mux1/g334/B      View : func_slow

  max_transition :     0.814
- Transition Time:     1.099
---------------------------------------------------
  slack          :    -0.285 ( VIOLATED )
encounter 13>
```

Then the hold violations for the design were



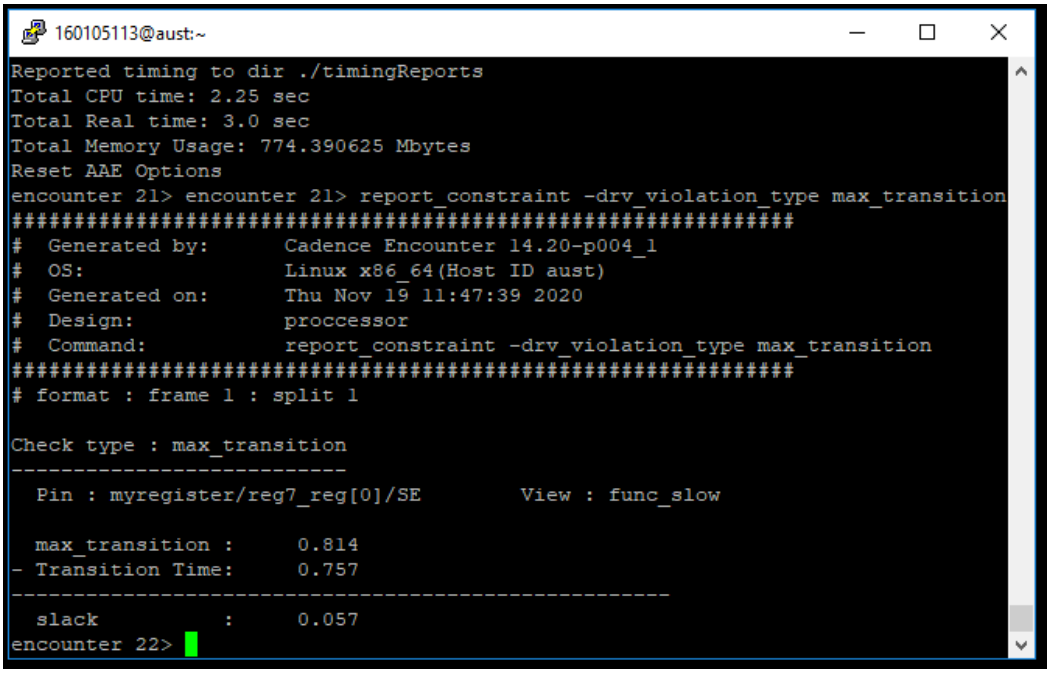
Therefore, the initial reports show that there are

a. No setup violations
b. Transition violation
c. 252 Hold violations

Step 2 (Cleaning the violations):

a. Cleaning the transition violation:

After cleaning the transition violation, the max_transition report is like as follows

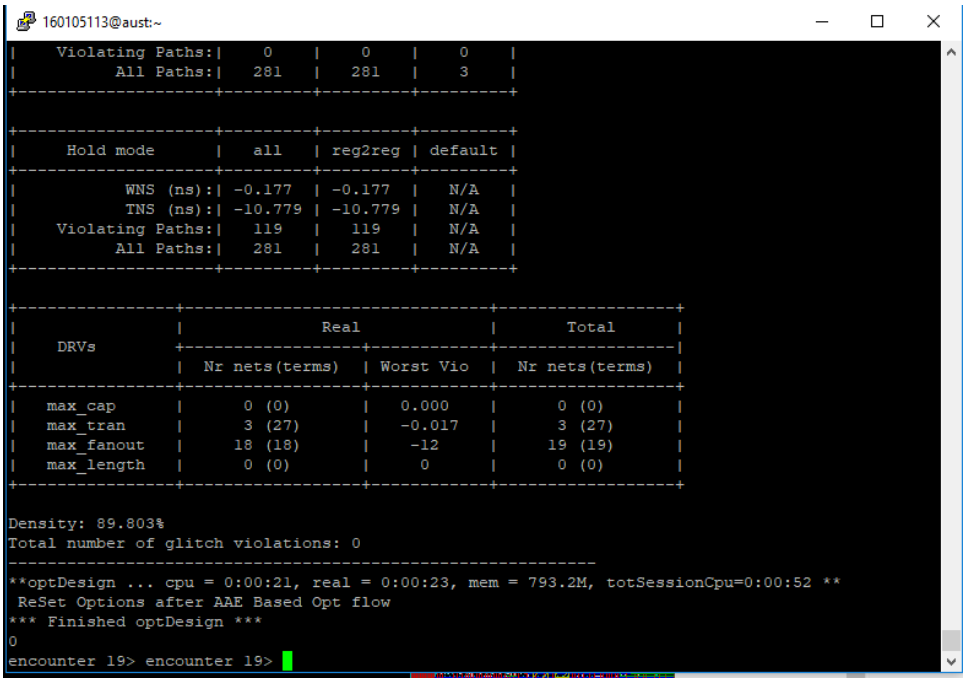


b. Cleaning the hold violations:

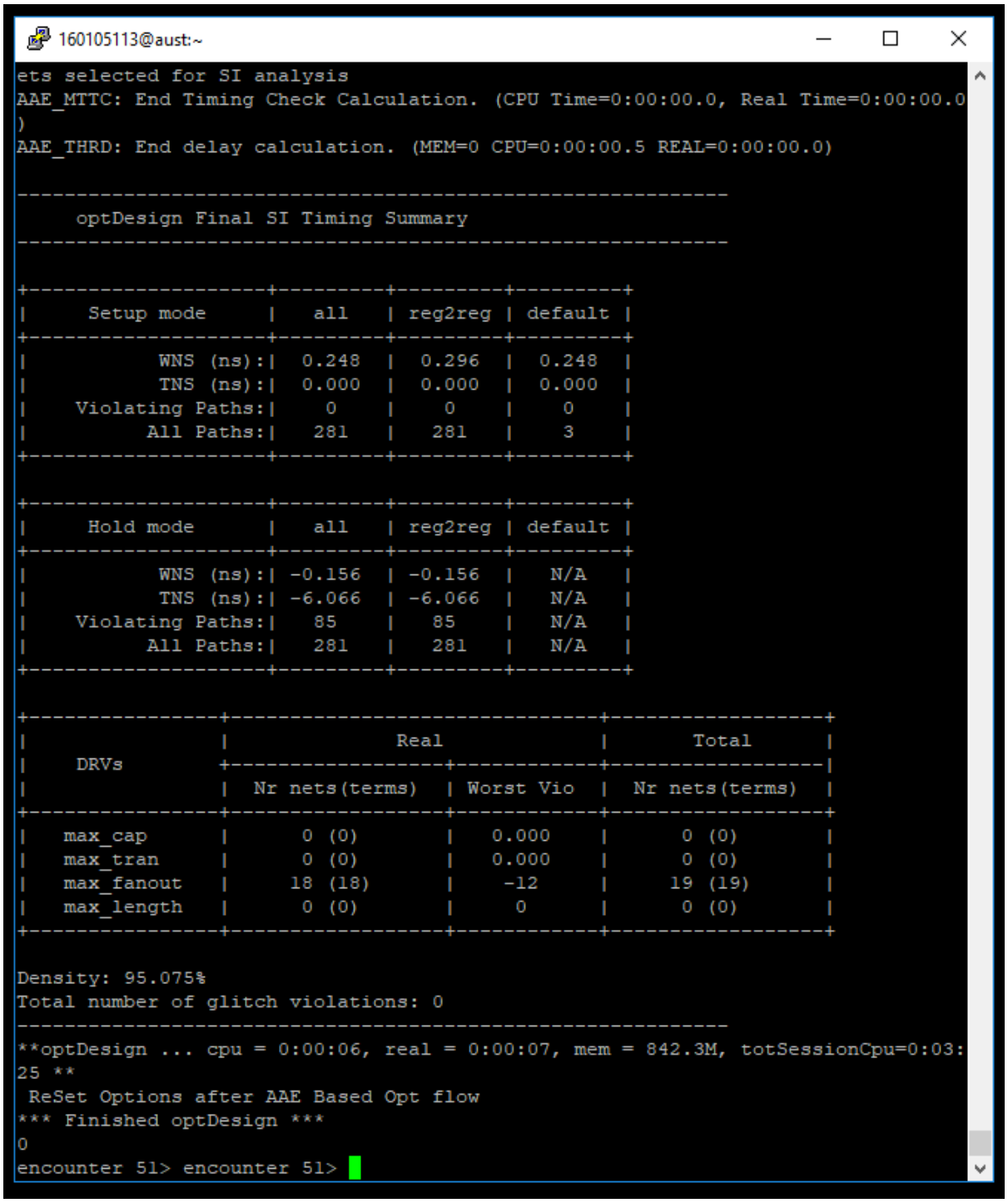
First the hold violations were removed using the tool by the following command

```
optDesign -hold -postRoute
```

This reduced the violations to a certain amount.



Then ECO placement and routing was done and again optimization gave the following results with reduced violations.

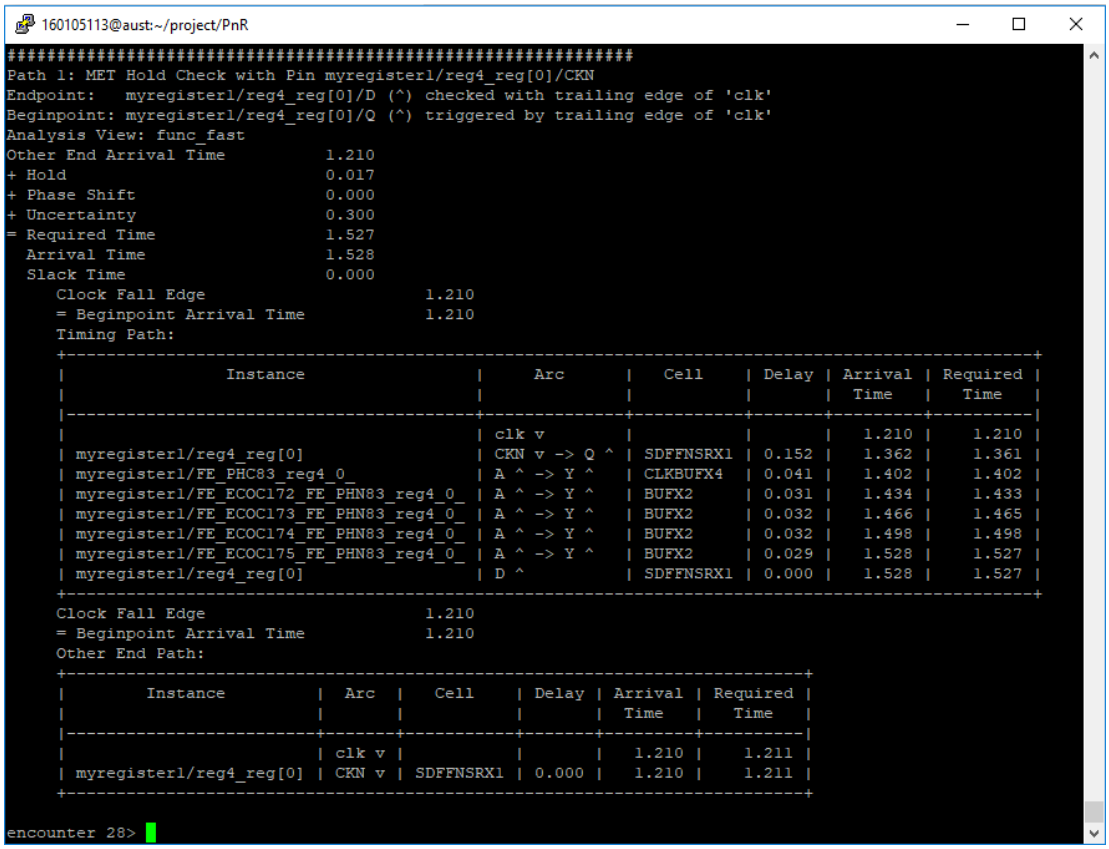


The remaining hold violations have to be moved manually. At first the hold violators are determined using the following code

```
setAnalysisMode -checkType hold
```

```
report_timing -check_type hold -max_paths 150 -nworst 10 -path_type full_clock
```

Then one of the violations is selected and then buffer cells are added until that specific VIOLATION becomes MET. It required two iteration so two BUFX2 cells were added.

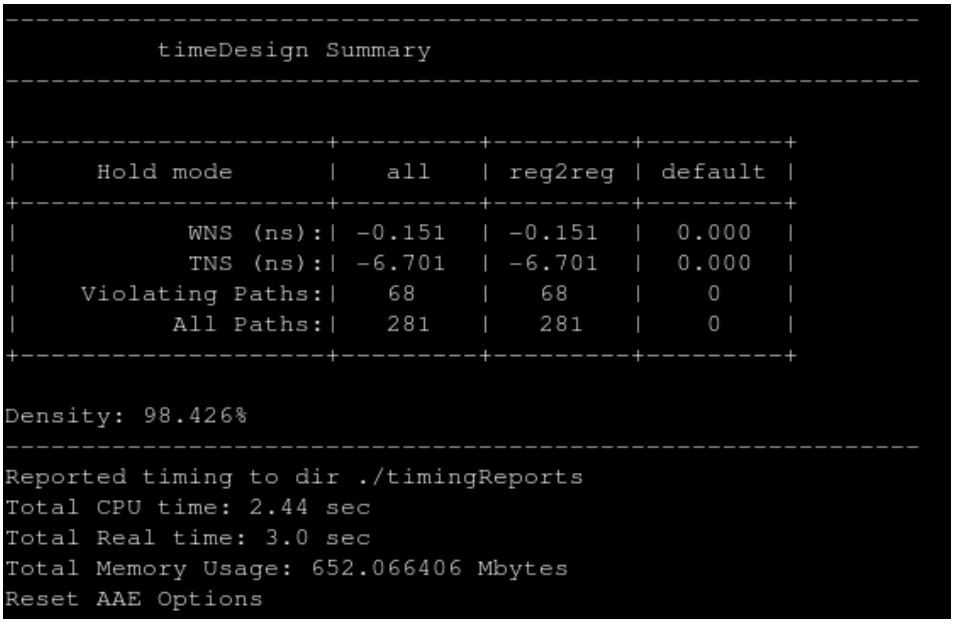


The following command was used twice to place the buffer cells

```
ecoAddRepeater -term myregister1/reg4_reg[0]/D -cell BUFX2 -relativeDistToSink 0.1
```

This places BUFX2 1-unit distance before the Endpoint.

Similarly, after manually cleaning a few more violations, the Violating Paths reduced to 68.

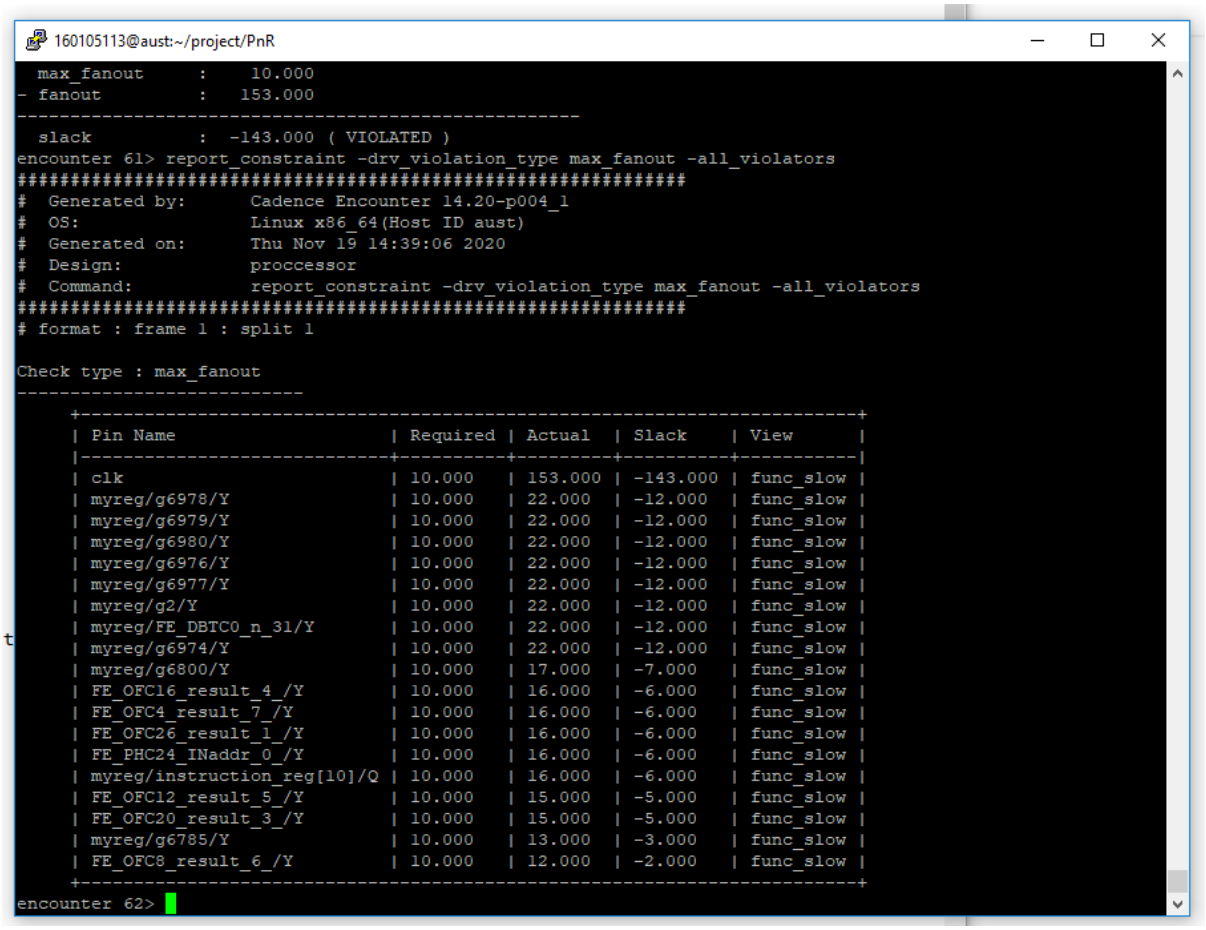


Buffer tree synthesis

In the design, the fanout violations were listed as follows

DRVs	Real		Total	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	
max_cap	0 (0)	0.000	0 (0)	
max_tran	0 (0)	0.000	0 (0)	
max_fanout	18 (18)	-12	19 (19)	
max_length	0 (0)	0	0 (0)	

And the violators were



Pin Name	Required	Actual	Slack	View
clk	10.000	153.000	-143.000	func_slow
myreg/g6978/Y	10.000	22.000	-12.000	func_slow
myreg/g6979/Y	10.000	22.000	-12.000	func_slow
myreg/g6980/Y	10.000	22.000	-12.000	func_slow
myreg/g6976/Y	10.000	22.000	-12.000	func_slow
myreg/g6977/Y	10.000	22.000	-12.000	func_slow
myreg/g2/Y	10.000	22.000	-12.000	func_slow
myreg/FE_DBTC0_n_31/Y	10.000	22.000	-12.000	func_slow
myreg/g6974/Y	10.000	22.000	-12.000	func_slow
myreg/g6800/Y	10.000	17.000	-7.000	func_slow
FE_OFC16_result_4_/Y	10.000	16.000	-6.000	func_slow
FE_OFC4_result_7_/Y	10.000	16.000	-6.000	func_slow
FE_OFC26_result_1_/Y	10.000	16.000	-6.000	func_slow
FE_PHC24_INaddr_0_/Y	10.000	16.000	-6.000	func_slow
myreg/instruction_reg[10]/Q	10.000	16.000	-6.000	func_slow
FE_OFC12_result_5_/Y	10.000	15.000	-5.000	func_slow
FE_OFC20_result_3_/Y	10.000	15.000	-5.000	func_slow
myreg/g6785/Y	10.000	13.000	-3.000	func_slow
FE_OFC8_result_6_/Y	10.000	12.000	-2.000	func_slow

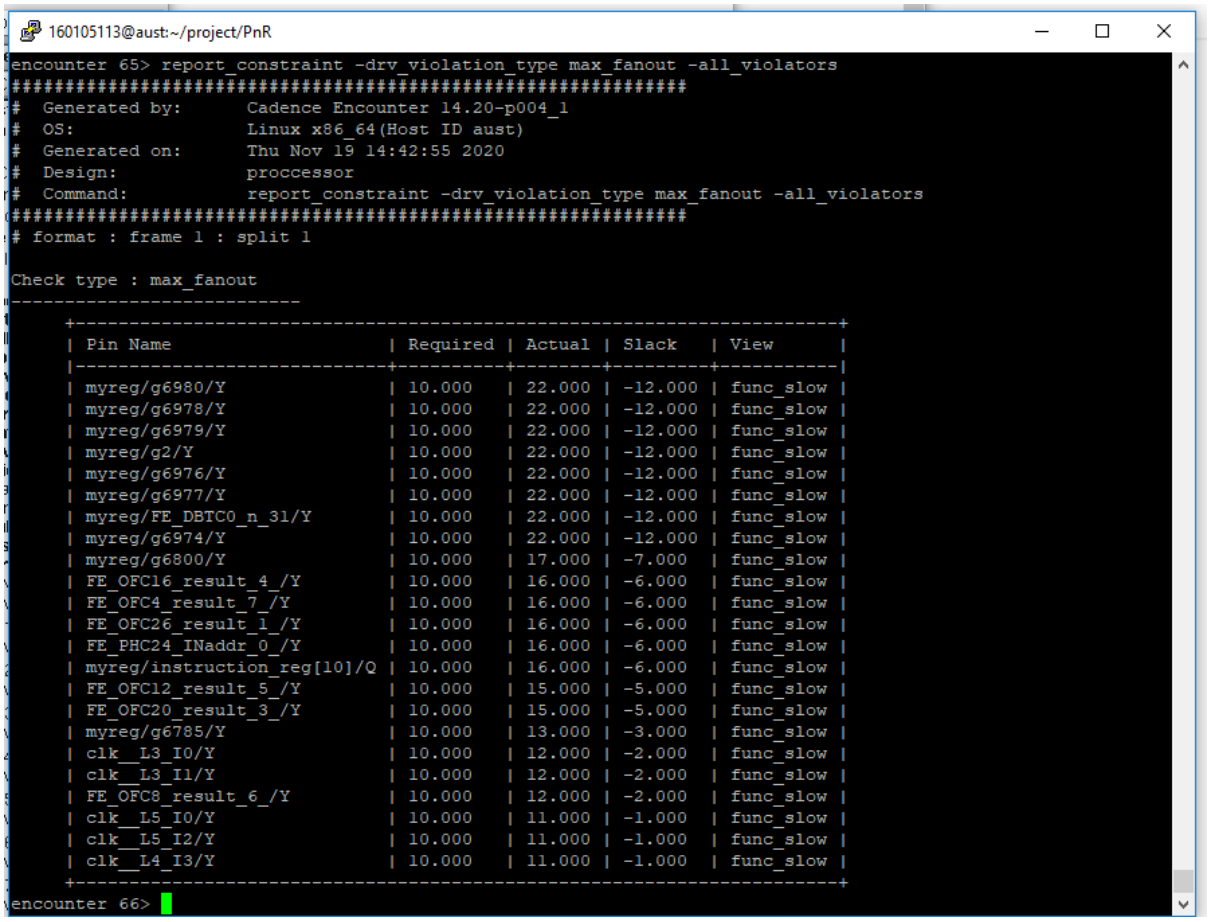
The nets for the violators were obtained with the following code

```
reportFanoutViolation
```

Then the each of the violations were removed manually. The code for the first violator (clk) is

```
bufferTreeSynthesis -nets clk -maxFanout 10
```

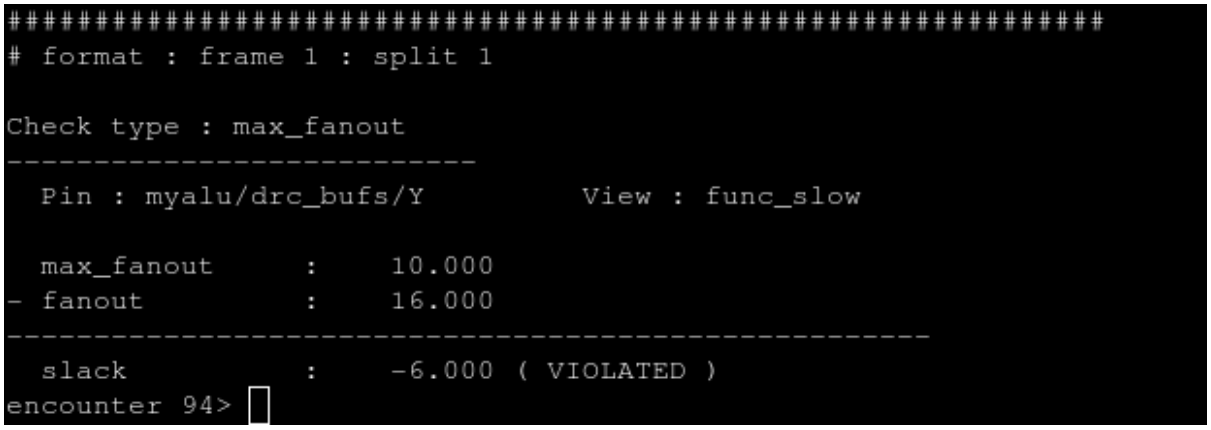
After cleaning the violation, the list looks like following



Before BTS, the maximum fanout was

```
encounter 5> report_constraint -drv_violation_type max_fanout
#####################################################################
#  Generated by:      Cadence Encounter 14.20-p004_1
#  OS:                Linux x86_64(Host ID aust)
#  Generated on:      Wed Nov 18 13:25:49 2020
#  Design:            proccessor
#  Command:           report_constraint -drv_violation_type max_fanout
#####################################################################
# format : frame 1 : split 1

Check type : max_fanout
---------------------------
  Pin : clk      View : func_slow

  max_fanout     :    10.000
- fanout         :   153.000
-----------------------------------------------------
  slack          :  -143.000 ( VIOLATED )
```

After manually fixing most of the violators, the maximum fanout became



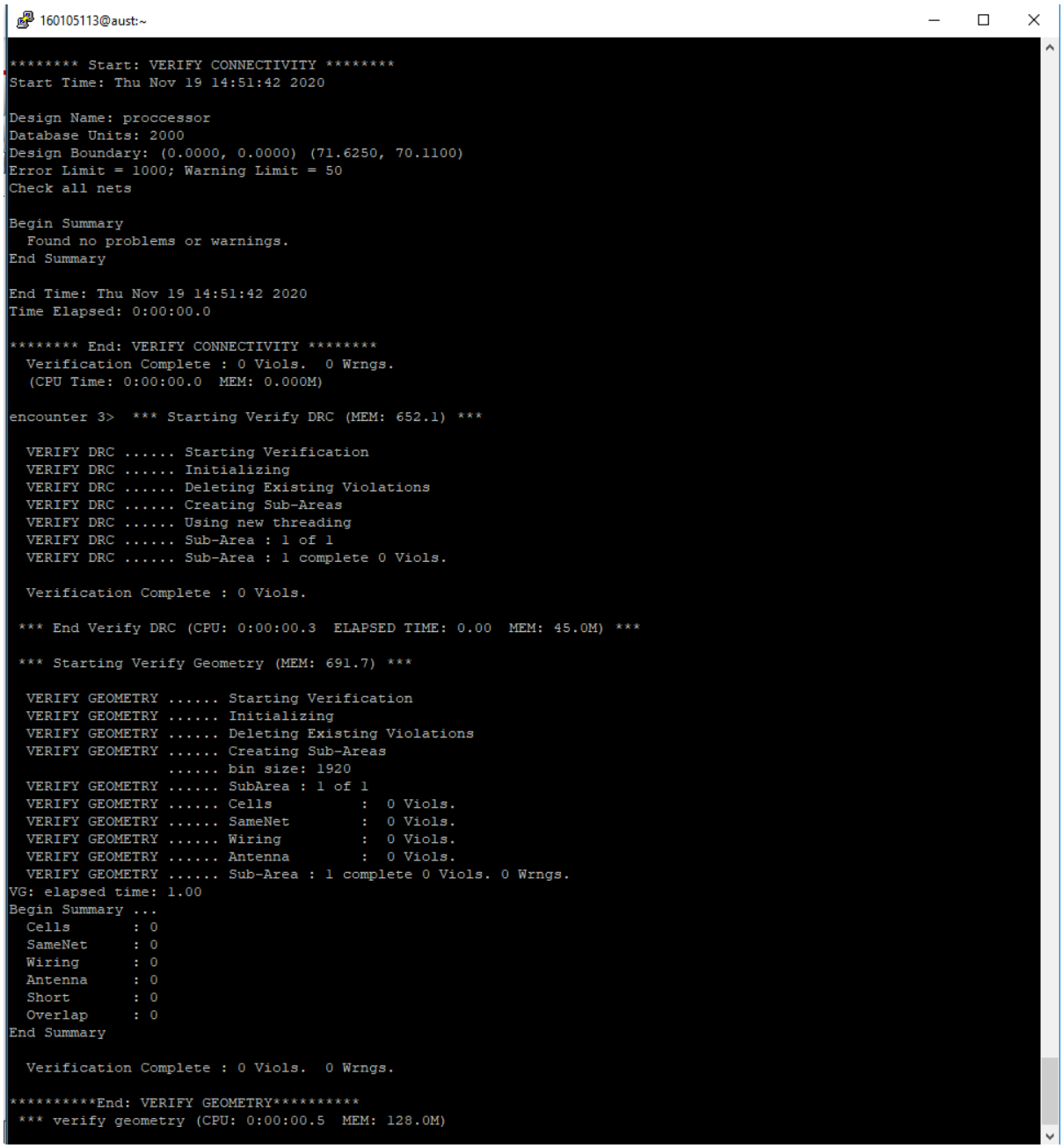
The remaining violators could not be cleaned with the following code anymore

```
bufferTreeSynthesis -nets <list_of_nets> -maxFanout 10
```

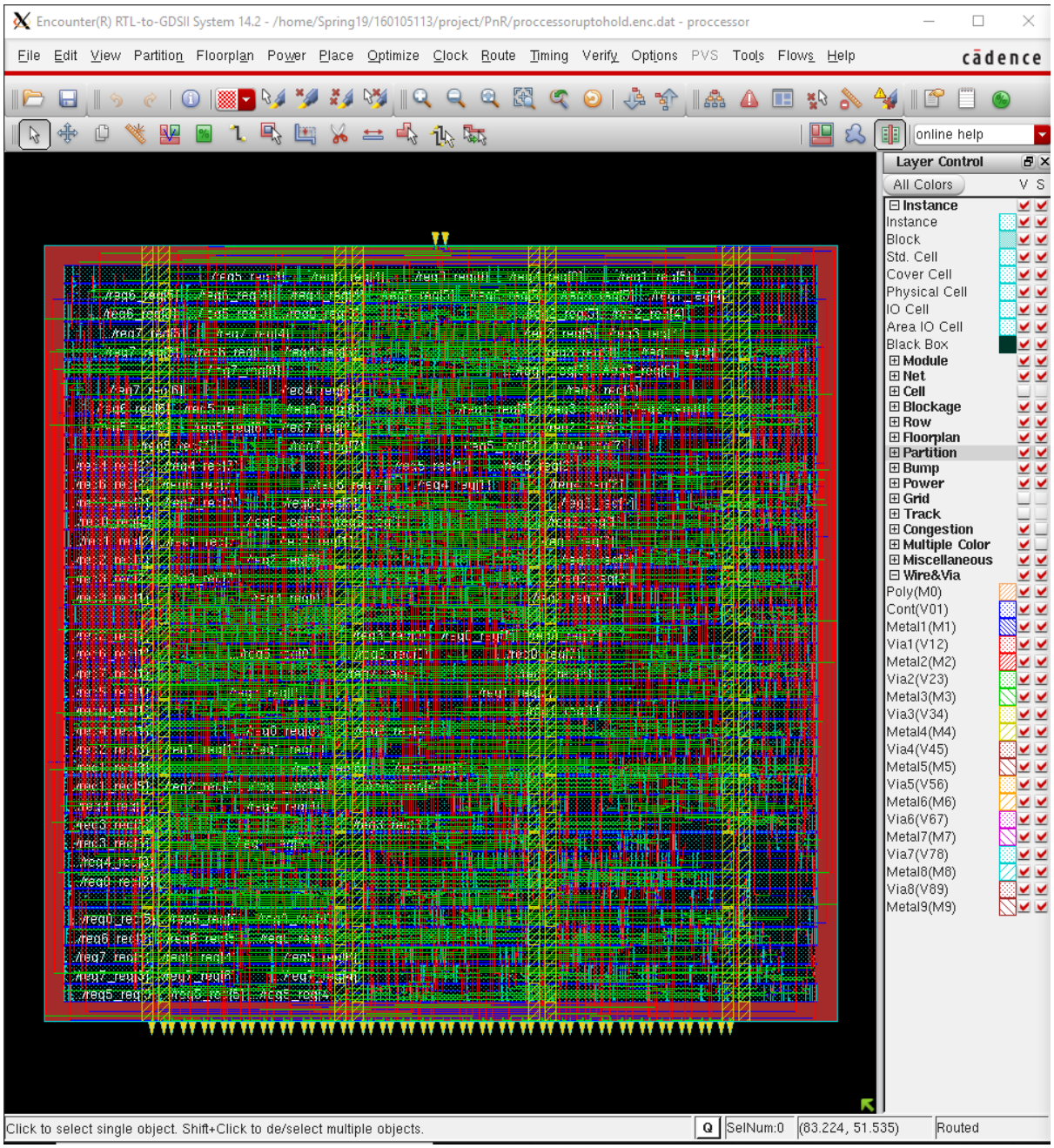
Therefore, the Buffer Tree Synthesis reduced the maximum fanout for the design.

Verification

Filler cells are added after the violations are reduced to a certain extent. Then in the GUI mode, connectivity, DRC and geometry verification is done. The design passes all of the verifications.



Therefore, the final design looks like following:



Conclusion and Problems

The whole project was constructed following all of the given design specifications. The design passed the DRC, geometry and connectivity verifications. Therefore, there are no problems with the placement or routing of the design. But there were problems during the Static Timing Analysis. The specification mentioned that the design should be free of all setup and hold violations. No setup violations were found during the first analysis of the design. But there were significant amounts of hold violations among which almost two-thirds were met using both the tool and manually adding buffer cells to the nets. But still there are some violations remaining which could be met if further iterations are run. Also, the value of maximum fanout was reduced using Buffer Tree Synthesis. Although the maximum fanout still violates the specification, it was also reduced to a significant amount from what it was during the first check.