LAB Report of ASIC Design Flow:RTL to GDS

Implementation of a Moore FSM Sequential Circuit

Group Members: Manish Kumar, Rahul Rathore, Deepak Kumar Dubey

Guide: Dr. Rohit Chaurasiya

Institute: Indian Institute of Technology Jammu

Department: Electrical Engineering

Lab: IC-ResQ Lab

Abstract

This report documents the complete RTL to GDS flow of a Moore FSM (Finite State Machine) based sequential circuit. The FSM was implemented using Verilog, verified using simulation tools, synthesized, and physically designed using Cadence tools. The final layout was exported as GDSII after passing all verification checks. The project highlights all essential stages: from RTL coding and synthesis to floorplanning, placement, routing, and DRC/LVS.

1 Tools Used

- Vivado (for RTL simulation and schematic)
- Cadence Genus (for logic synthesis)
- Cadence Innovus (for placement, CTS, routing)
- NCLaunch (for simulation)

2 Moore FSM Sequential Circuit

We implemented a Moore Finite State Machine (FSM) where the outputs depend entirely on the current state, independent of input values at that specific clock cycle. This means the inputs influence only the transitions to the next state. Output changes occur only on clock edges, after a state transition, making the outputs stable and glitch-free—ideal for synchronous applications.

The Vivado schematic clearly illustrates this architecture: it includes a state register built using flip-flops to hold the current state, a combinational next-state logic block that determines the upcoming state based on the current state and inputs, and an output logic block that derives outputs solely from the current state. This structure reflects the fundamental principles of a Moore FSM and was successfully verified in RTL simulation.

3 RTL Designing and Simulation Testing

Step 1: Open Vivado, write the FSM code in Verilog (.v), correct any syntax errors, and generate the schematic.

Step 2: Write the testbench in Verilog (.v) format and simulate it using the Behavioral Simulation tool in Vivado.

Here we are trying to get check the RTL design in which we get the netlist according to the tool Optimization technique where we get the schematic and use the logical tool and with different simulation test vectors we can check the functionality of the tool.

4 Genus Tool Usage:

Genus is a tool from Cadence used for logic synthesis in ASIC design.

Genus takes our RTL code (like Verilog or VHDL) and converts it into a gate-level netlist using standard cells. It checks timing, area, and power, and lets you write constraints using .sdc files or automate tasks with .tcl scripts. Basically, it's the first step in turning our code into real hardware for physical design tools like Innovus.

We use Genus Tool to get the synthesis netlists in .v format according to the library files which have been provided by the foundry in which we generally use slow.lib or fast.lib.

TCL file is a script used to automate commands in tools like Genus or Vivado. Instead of clicking buttons, you write steps like reading files, setting clocks, and running synthesis. It saves time and makes the design flow repeatable and easy to debug.

After the successful running of tcl it generates the following report which shows the functionality and brief of the circuit constraints design as well as the reports of the provided information which are important for the designing.

In ASIC design, .lib files are timing library files provided by the foundry (e.g., TSMC, GlobalFoundries, etc.). They describe the timing, power, and functional behavior of standard cells at different conditions. These files are used during synthesis, timing analysis, and place-and-route.

slow.lib (also called worst-case corner): This file models the slowest performance of the standard cells — for example, when voltage is low and temperature is high. It ensures the chip works reliably in the worst-case conditions (slow speed, highest delays). Used for setup timing checks.

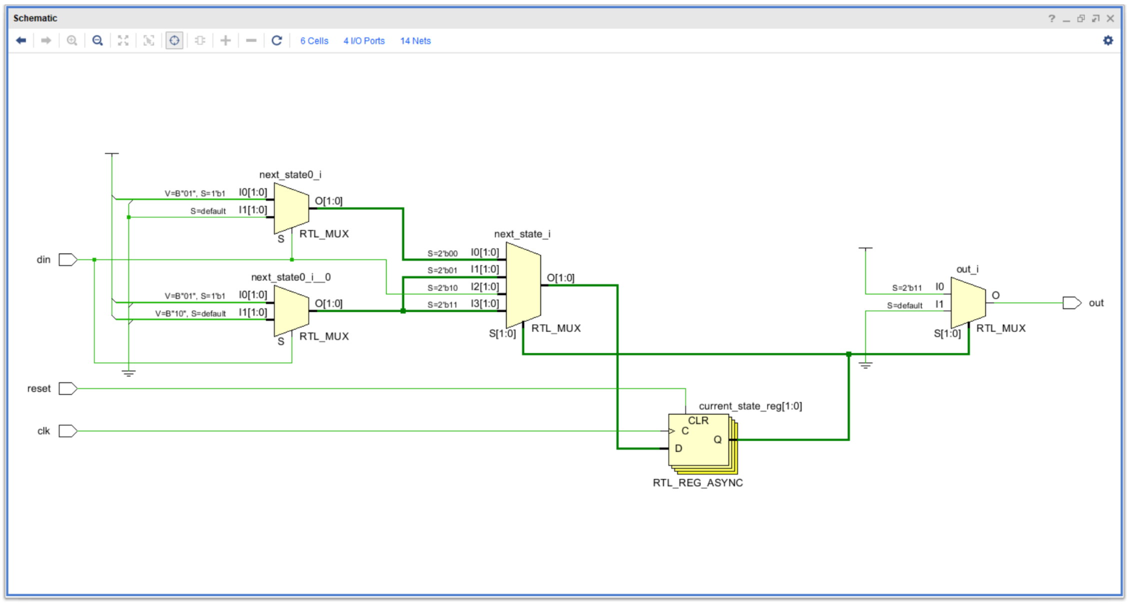


Figure 1: FSM Schematic in Vivado

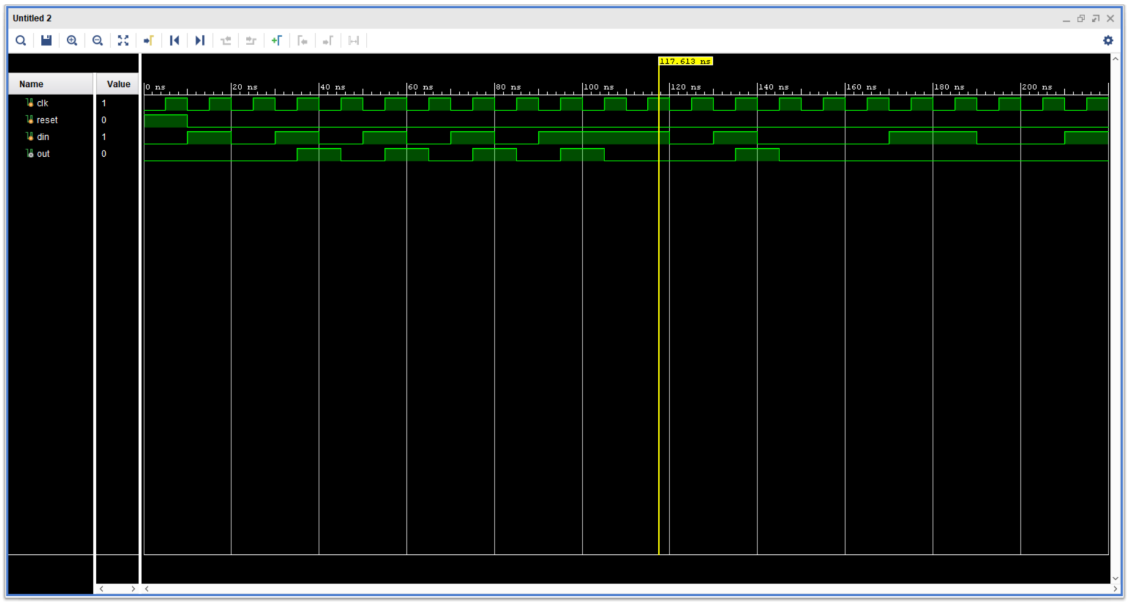
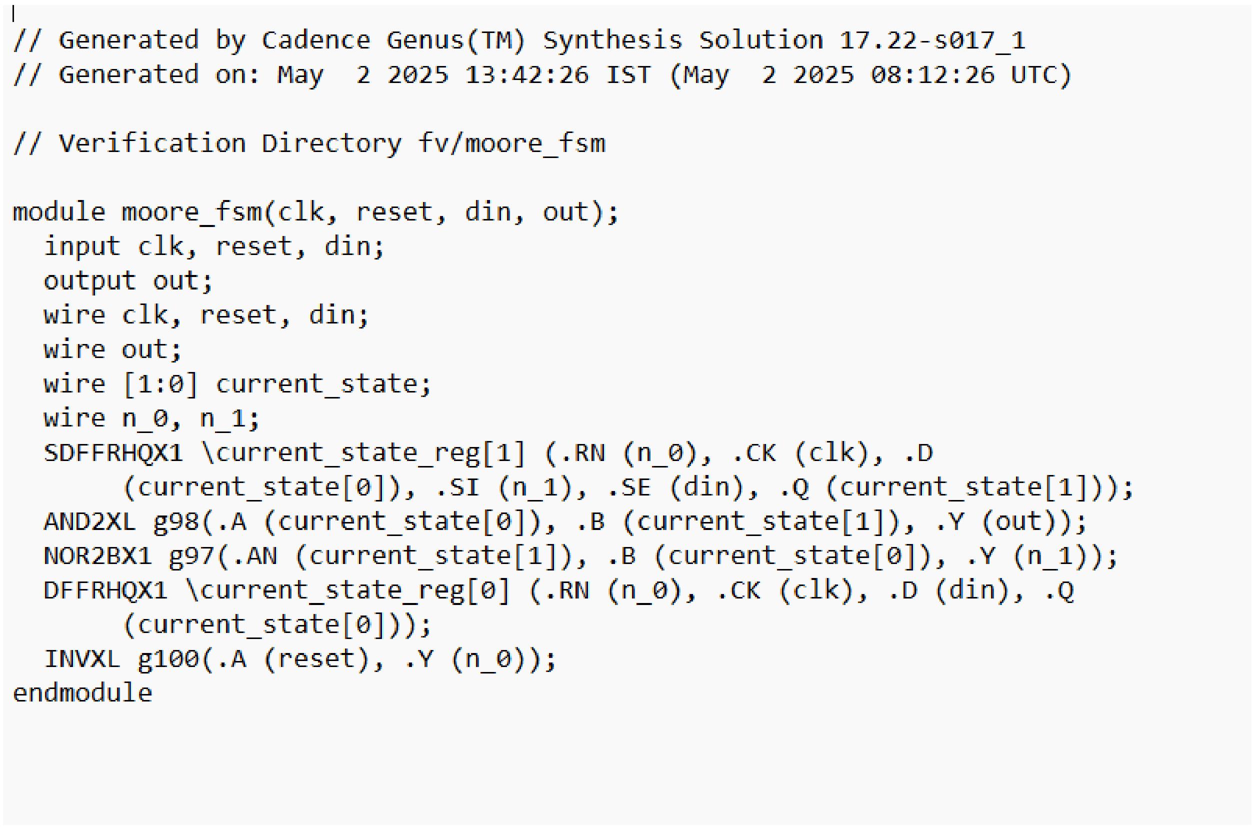


Figure 2: RTL Simulation Result in Vivado



```
// Generated by Cadence Genus(TM) Synthesis Solution 17.22-s017_1
// Generated on: May  2 2025 13:42:26 IST (May  2 2025 08:12:26 UTC)

// Verification Directory fv/moore_fsm

module moore_fsm(clk, reset, din, out);
  input clk, reset, din;
  output out;
  wire clk, reset, din;
  wire out;
  wire [1:0] current_state;
  wire n_0, n_1;
  SDFFRHQX1 \current_state_reg[1] (.RN (n_0), .CK (clk), .D
       (current_state[0]), .SI (n_1), .SE (din), .Q (current_state[1]));
  AND2XL g98(.A (current_state[0]), .B (current_state[1]), .Y (out));
  NOR2BX1 g97(.AN (current_state[1]), .B (current_state[0]), .Y (n_1));
  DFFRHQX1 \current_state_reg[0] (.RN (n_0), .CK (clk), .D (din), .Q
       (current_state[0]));
  INVXL g100(.A (reset), .Y (n_0));
endmodule
```

Figure 3: Synthesis Netlist after genus

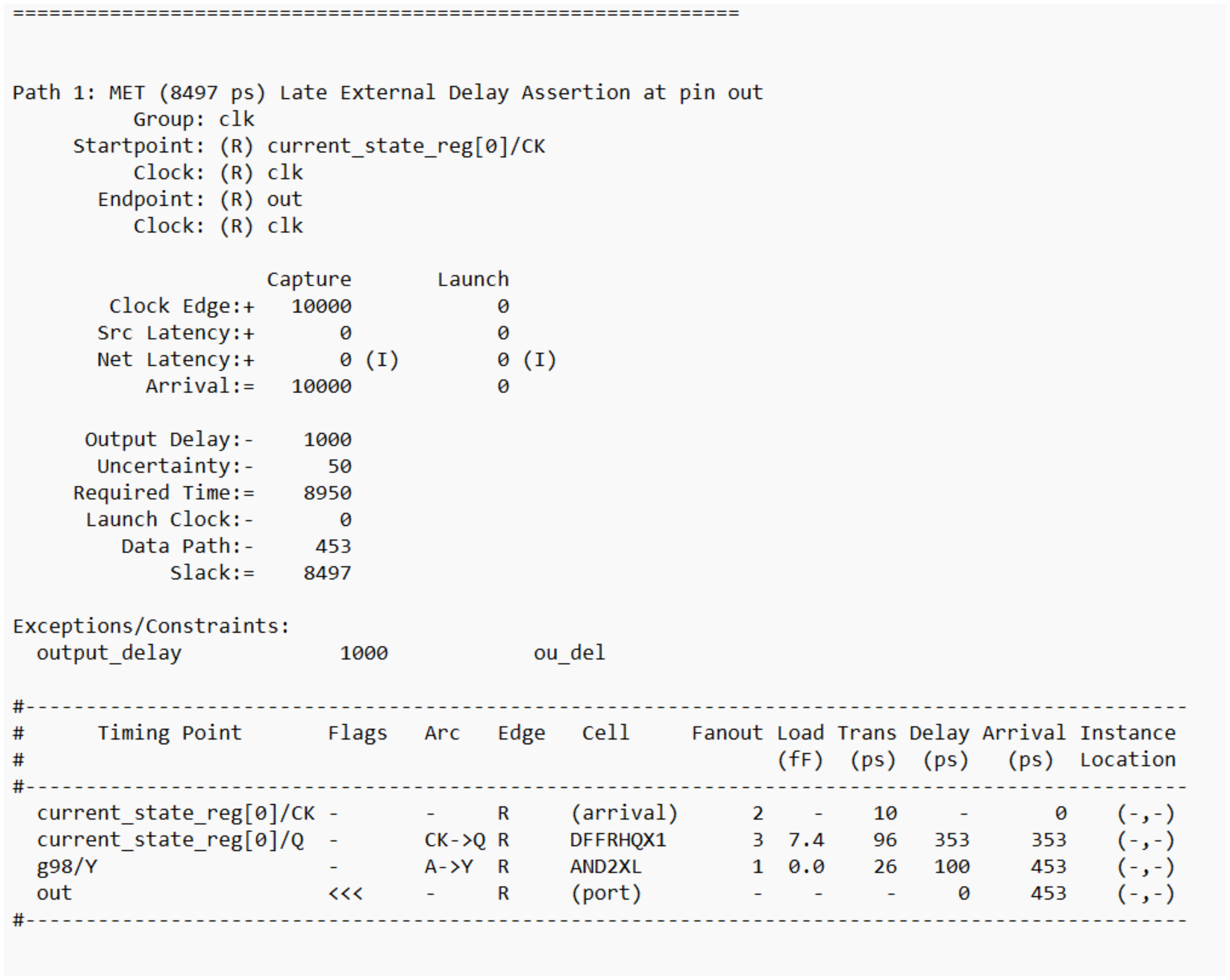
Fast.lib (also called best-case corner): This file models the fastest performance — for example, when voltage is high and temperature is low. It's used to check for hold violations, where signals arrive too early. Used for hold timing checks.

.v (Netlist File): The .v file is a Verilog netlist generated after synthesis, containing gate-level connections of the design. It shows how standard cells are connected but no longer includes high-level logic like if or case statements.

.sdc (Synopsys Design Constraints): The .sdc file defines design constraints like clock definitions, input/output delays, and timing exceptions. It guides the synthesis and place-and-route tools to meet timing and performance goals.

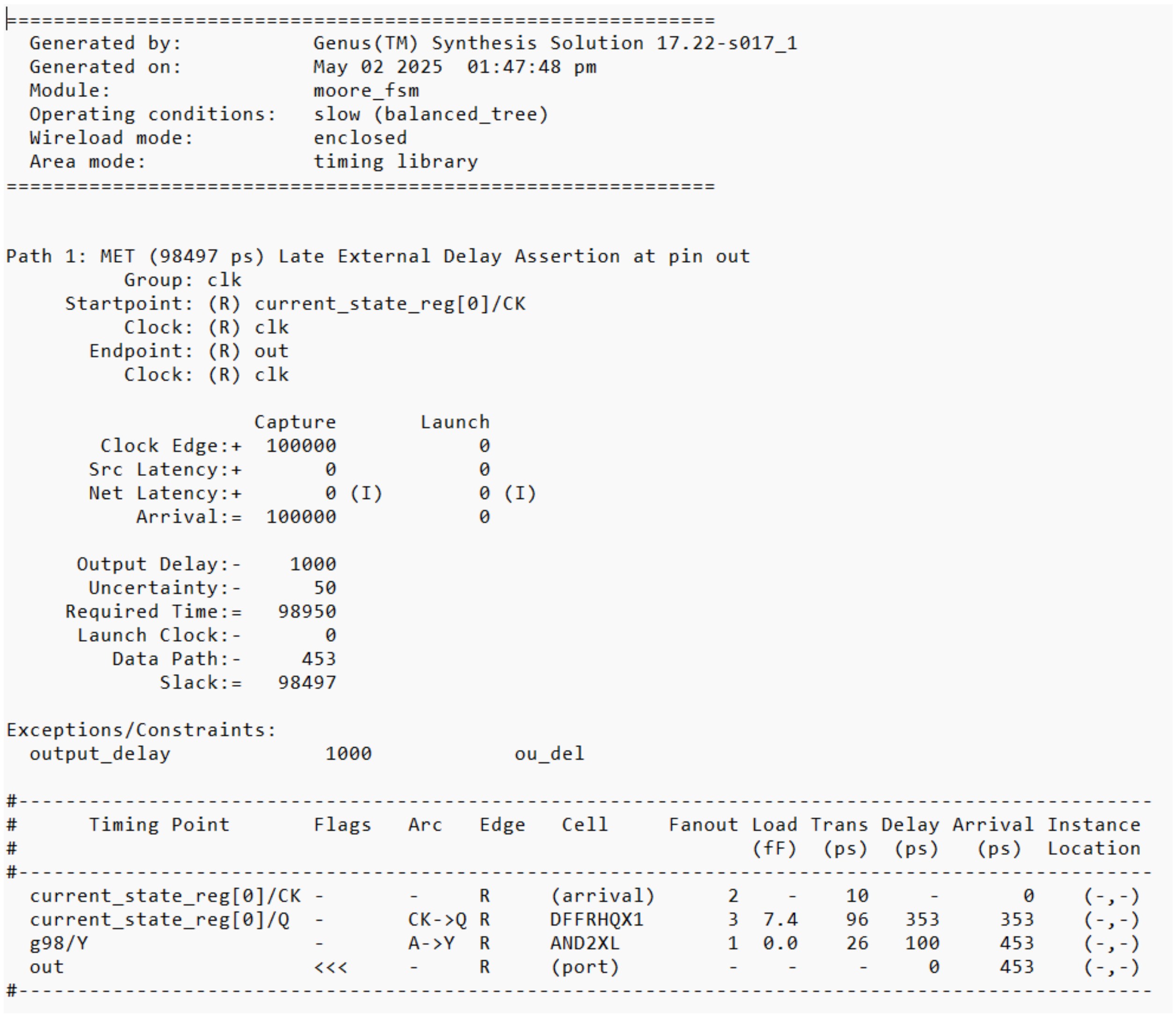
Timing report checks if the design meets the required timing constraints, such as a 10 ps or 100 clock period. It shows setup and hold timing for all paths, helping to ensure the circuit works correctly at the target speed.

Cell report lists all the standard cells used after synthesis and gives details like cell delay, area, and function. With tighter timing (e.g., 10 ps), faster and sometimes larger cells may be used to meet timing.



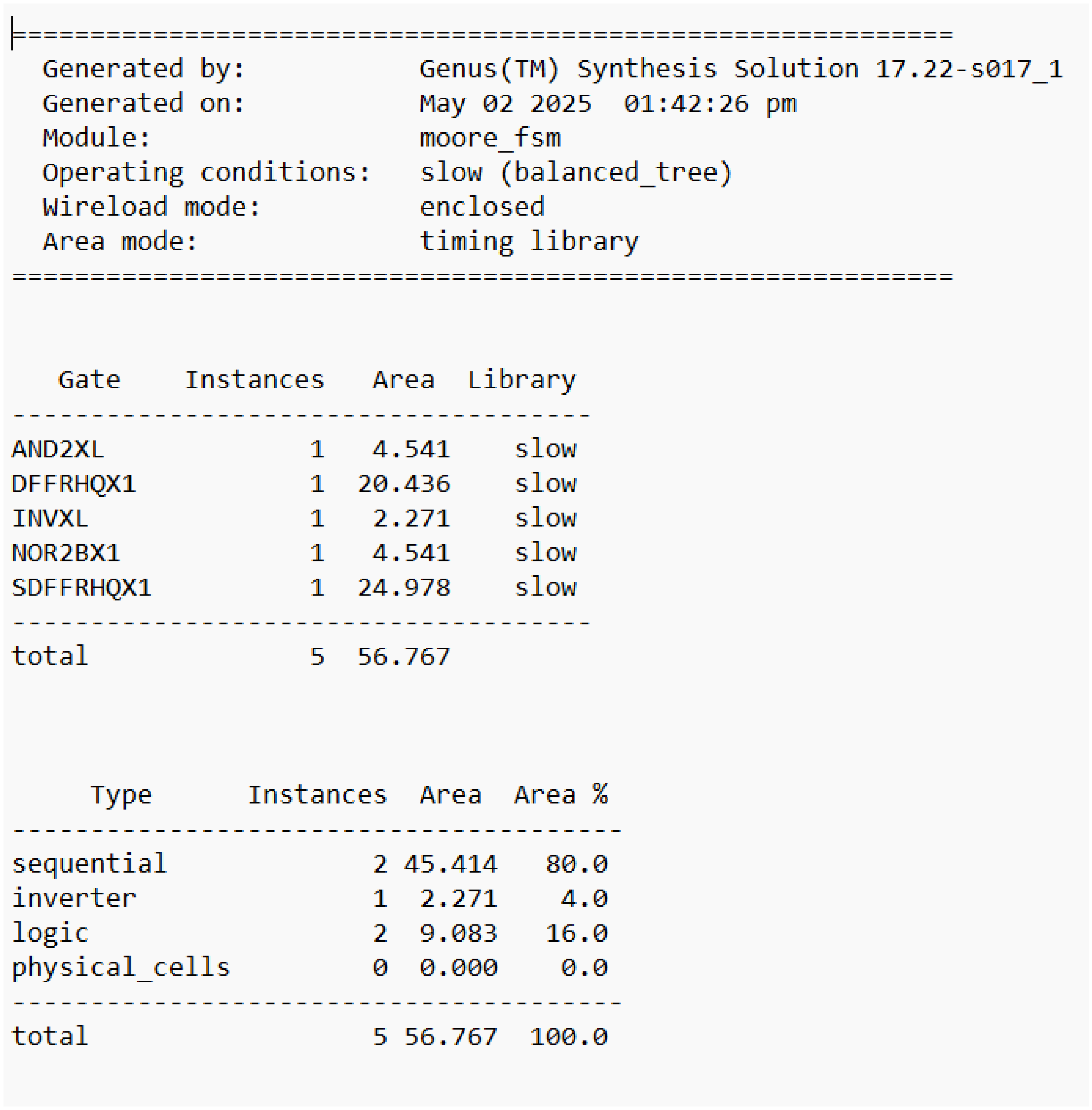
```
============================================================


Path 1: MET (8497 ps) Late External Delay Assertion at pin out
          Group: clk
     Startpoint: (R) current_state_reg[0]/CK
          Clock: (R) clk
       Endpoint: (R) out
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+   10000            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=   10000            0

      Output Delay:-    1000
       Uncertainty:-      50
     Required Time:=    8950
      Launch Clock:-       0
         Data Path:-     453
             Slack:=    8497

Exceptions/Constraints:
  output_delay             1000            ou_del

#-----------------------------------------------------------------------------------------------
#      Timing Point       Flags    Arc    Edge    Cell      Fanout  Load Trans Delay Arrival Instance
#                                                                   (fF)  (ps)  (ps)   (ps)  Location
#-----------------------------------------------------------------------------------------------
  current_state_reg[0]/CK -        -      R      (arrival)       2    -    10     -       0    (-,-)
  current_state_reg[0]/Q  -        CK->Q  R      DFFRHQX1        3  7.4    96   353     353    (-,-)
  g98/Y                   -        A->Y   R      AND2XL          1  0.0    26   100     453    (-,-)
  out                     <<<      -      R      (port)          -    -     -     0     453    (-,-)
#-----------------------------------------------------------------------------------------------
```

Figure 4: Timing Report @10



```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           May 02 2025  01:47:48 pm
  Module:                 moore_fsm
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


Path 1: MET (98497 ps) Late External Delay Assertion at pin out
          Group: clk
     Startpoint: (R) current_state_reg[0]/CK
          Clock: (R) clk
       Endpoint: (R) out
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+  100000            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=  100000            0

      Output Delay:-    1000
       Uncertainty:-      50
     Required Time:=   98950
      Launch Clock:-       0
         Data Path:-     453
             Slack:=   98497

Exceptions/Constraints:
  output_delay             1000            ou_del

#-------------------------------------------------------------------------------------------------
#      Timing Point       Flags   Arc   Edge   Cell      Fanout Load Trans Delay Arrival Instance
#                                                               (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------
  current_state_reg[0]/CK -       -     R     (arrival)      2    -    10     -       0    (-,-)
  current_state_reg[0]/Q  -       CK->Q R     DFFRHQX1       3  7.4    96   353     353    (-,-)
  g98/Y                   -       A->Y  R     AND2XL         1  0.0    26   100     453    (-,-)
  out                     <<<     -     R     (port)         -    -     -     0     453    (-,-)
#-------------------------------------------------------------------------------------------------
```

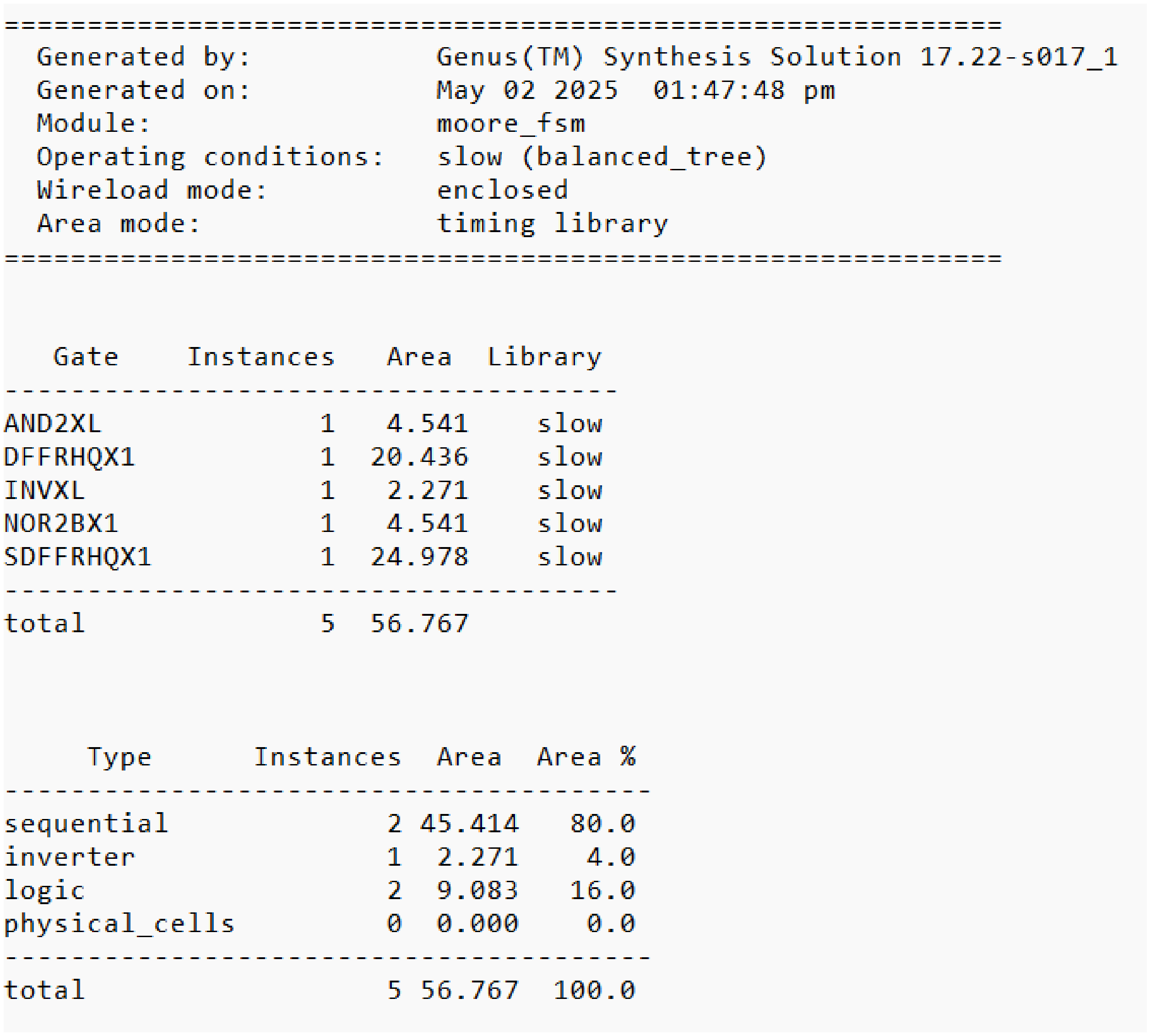
Figure 5: Timing Report @100



```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           May 02 2025  01:42:26 pm
  Module:                 moore_fsm
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================


   Gate    Instances   Area  Library
-------------------------------------
AND2XL             1    4.541    slow
DFFRHQX1           1   20.436    slow
INVXL              1    2.271    slow
NOR2BX1            1    4.541    slow
SDFFRHQX1          1   24.978    slow
-------------------------------------
total              5   56.767



     Type      Instances  Area  Area %
---------------------------------------
sequential             2 45.414   80.0
inverter               1  2.271    4.0
logic                  2  9.083   16.0
physical_cells         0  0.000    0.0
---------------------------------------
total                  5 56.767  100.0
```

Figure 6: Cell Report @10

Power report shows how much power the chip consumes, including dynamic, static, and leakage power. Designs targeting faster speeds like 10 ps usually consume more power due to increased switching and use of higher drive-strength cells.



```
============================================================
  Generated by:          Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:          May 02 2025  01:47:48 pm
  Module:                moore_fsm
  Operating conditions:  slow (balanced_tree)
  Wireload mode:         enclosed
  Area mode:             timing library
============================================================


   Gate    Instances   Area  Library
-------------------------------------
AND2XL             1   4.541    slow
DFFRHQX1           1  20.436    slow
INVXL              1   2.271    slow
NOR2BX1            1   4.541    slow
SDFFRHQX1          1  24.978    slow
-------------------------------------
total              5  56.767



     Type      Instances  Area  Area %
---------------------------------------
sequential             2 45.414   80.0
inverter               1  2.271    4.0
logic                  2  9.083   16.0
physical_cells         0  0.000    0.0
---------------------------------------
total                  5 56.767  100.0
```

Figure 7: Cell Report @100

5 NCLAUNCH

nclaunch is a graphical user interface (GUI) tool in Cadence used to manage simulation setups for digital designs.

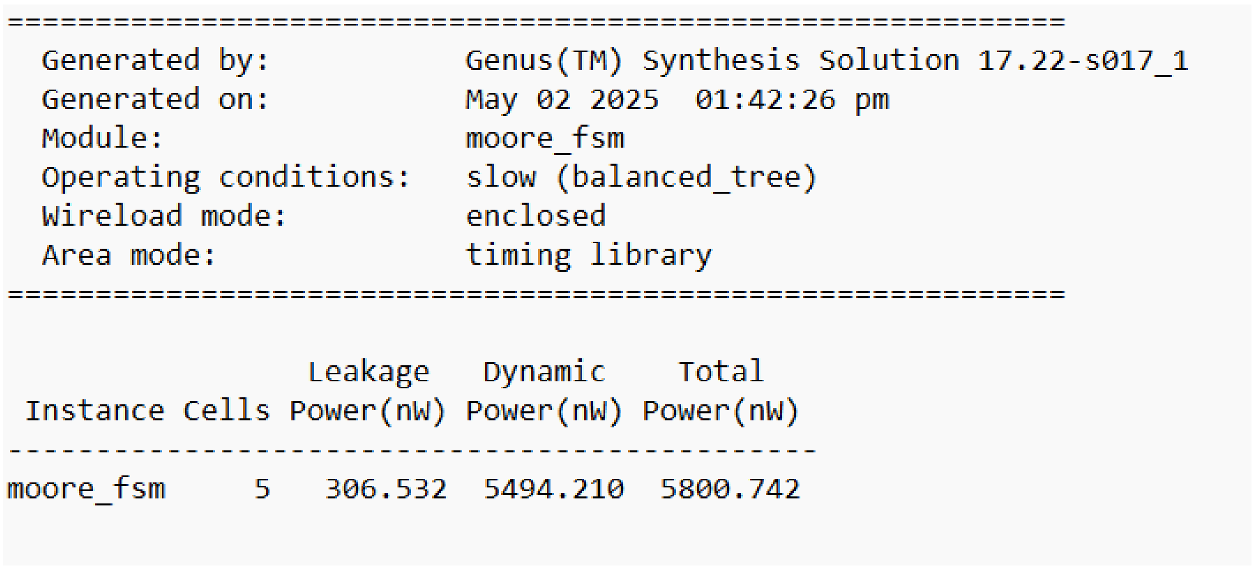
Step 1: In the Terminal write the nclaunch to open the tool.

Step 2: Copy the file in the download folder of slow.v from the directory saved in vlog folder.

Step 3: Send the slow.v , synthesis after netlist.v and the testbench .v file in the compiler hdl and the result store in the Worklib file.

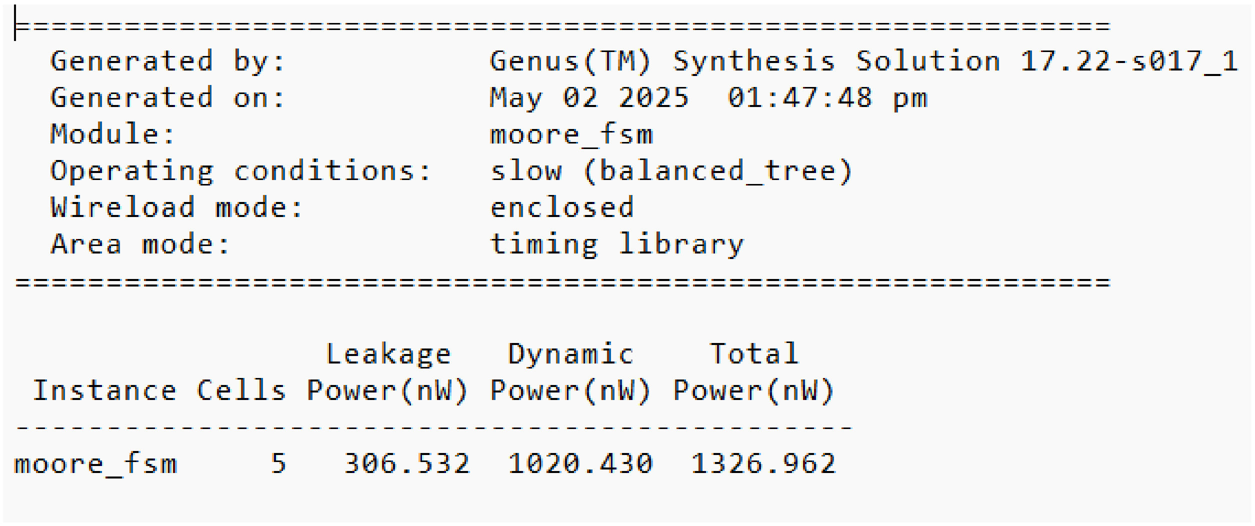
Step 4: Open the workLib file and select the module name file in .v format same for testbench .v format and add the elaborater on each .v file.

Step 5: Open the snapshots, select the generated testbench file and open the gui graphics.



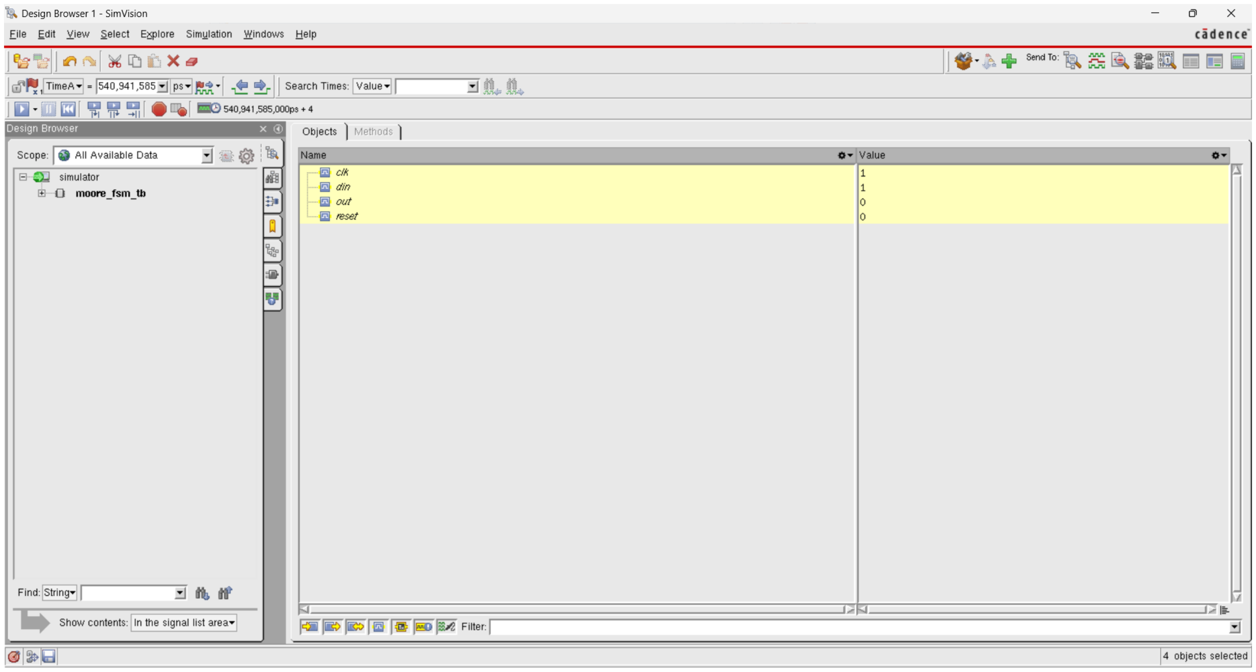
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           May 02 2025  01:42:26 pm
  Module:                 moore_fsm
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                 Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
----------------------------------------------
moore_fsm     5   306.532  5494.210  5800.742
```

Figure 8: Power report @10



```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           May 02 2025  01:47:48 pm
  Module:                 moore_fsm
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                 Leakage   Dynamic    Total
 Instance Cells Power(nW) Power(nW) Power(nW)
----------------------------------------------
moore_fsm     5   306.532  1020.430  1326.962
```

Figure 9: Power Report @100



Name	Value
clk	1
din	1
out	0
reset	0

Figure 10: nclaunch variable selection

Step 6 : In new window, Select the Testbench File which is mentioned in .v format select all its input,output,clock and reset.

Step 7 : Run the testbench and observe the result.

Step 8 : Select all the data variables and sent to the waveform to check the functionality.

Step 9: (if needed) If error occur check the verilog code from the scratch and keep the functionality intact.

Note: Keep the timescale format in the testbench as well as in synthesis netlist

If all these steps works properly with no error we can now move to Physcial Design on Innovus Tool.

6 Genus Schematic

The Genus Schematic Viewer enables visualization of the synthesized gate-level design, displaying standard cells, connections, and signal flow. It helps in debugging, understanding logic implementation, and analyzing design hierarchy. Users can trace paths, examine control logic, and ensure correctness before moving to physical design, enhancing overall design verification efficiency.

7 Physcial Design

Physical design is about placing the gates and connecting them with wires on a chip. It includes steps like floorplanning, placement, clock tree synthesis (CTS), routing, and signoff checks. The goal is to make sure the design fits on the chip, works fast, uses less power, and meets all timing rules before manufacturing.

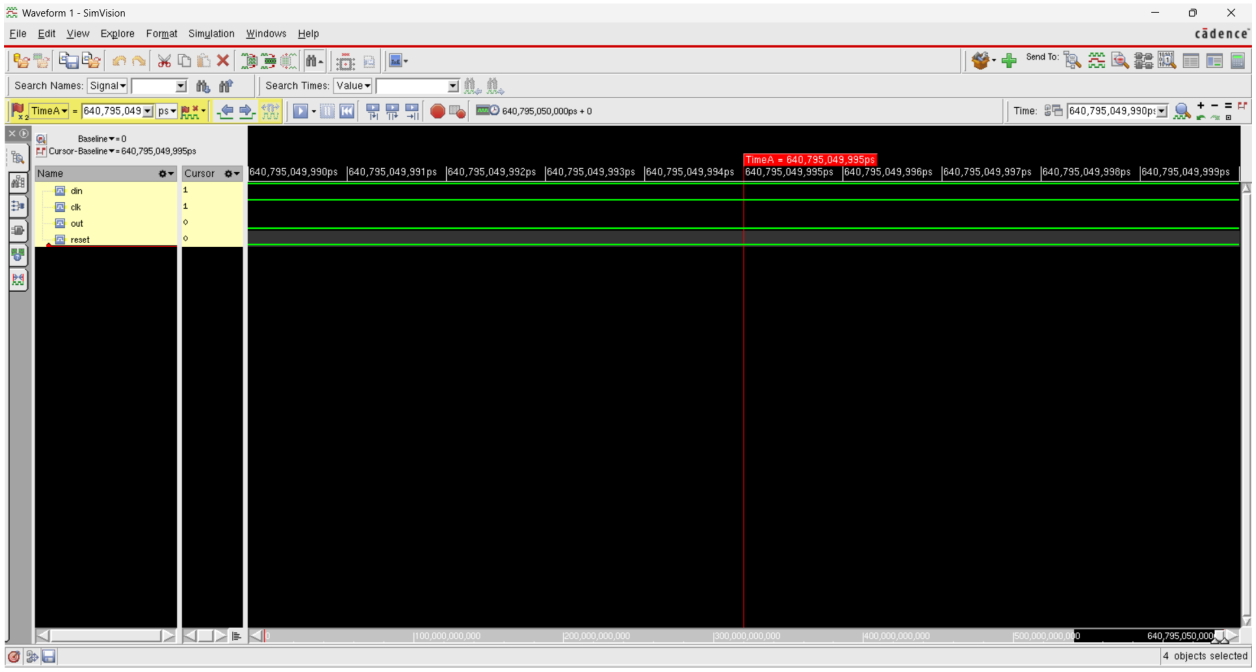


Figure 11: nclaunch simulation

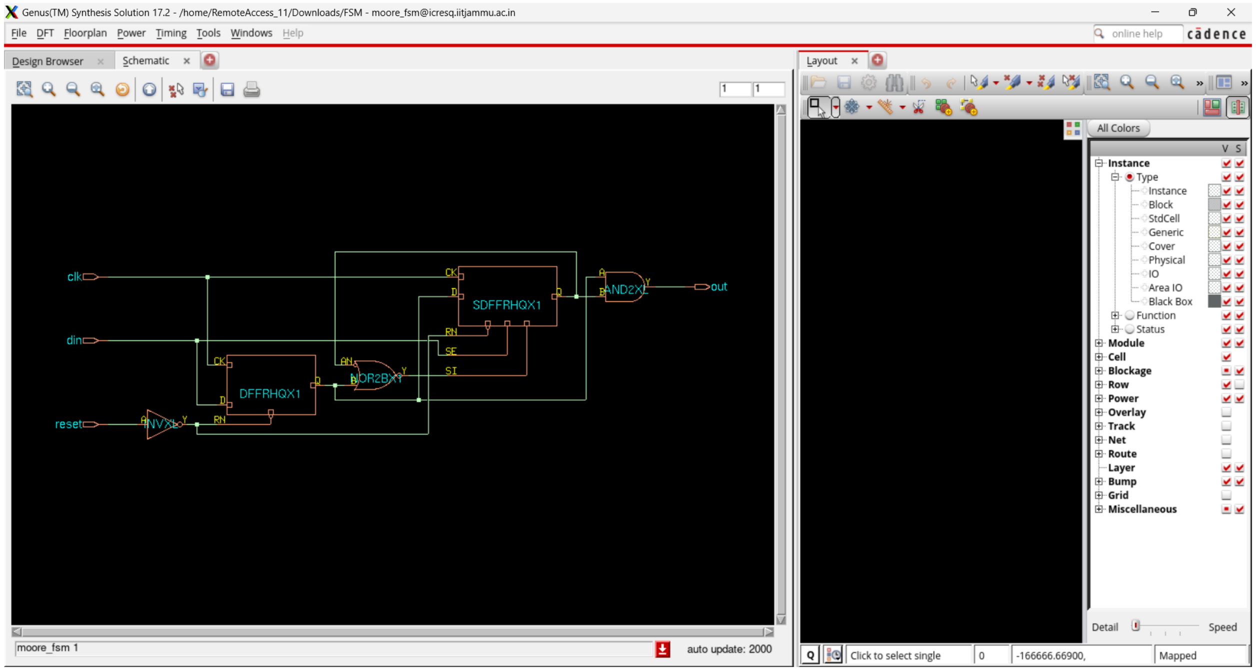


Figure 12: Genus FSM Schematic

Defines the layout area, places major blocks (like memories, macros), and sets up power planning. It's like deciding where rooms go before building a house.

Standard cells from the netlist are placed in rows to minimize wire length and meet timing. No routing yet—just positioning the gates efficiently.

Builds a balanced tree of buffers/inverters to evenly distribute the clock signal to all flip-flops. This helps reduce clock skew and timing problems.

Wires are drawn to connect the placed cells based on the netlist connections. The goal is to avoid congestion, shorts, and delays.

Step 1 : In the terminal type the command innovus to open the tool,then a new window of cadence open.

Step 2 : Go to the file, open the design, now fill all the essential files ,LEF files , netkist after synthesis.v files, and in MMMC add the max Timing with slow lib and min timing with fast lib, max delay with max timing and vice versa,add the setup and hold condition,in last all the netlist and design floor occur on the screen. The format will be of .view and .global.

Step 3 : Open the specify floorplan fill the necessary information like core utilization and add the aspect ratio and core to IO boundary to 10,click ok.

Step 4 :Do the power planning add the strips, rings with appropriate metals selection both horizontal and vertical with high metals.

Step 5 : Add the nets,by clicking on route and then special route and click on all metal layers.

Step 6: Go to place,physical cell, add encp. select the filler and add later the well tap instance with appropriate setting to rows to get the design.

Step 7 : Again Go to place,std. cell, then place io pins and click on ok, all the std. cell come into the die are.

Step 8 : In digital analysis, Go to report timing , preCTS to setup all information occur on the screen, then same will done for the hold .

Step 9: Go to the Clock Debugger click on CTS and apply ,path will be generated and save the design with .enc format.

Step 10:Go to Route ,Nano Route ,route add the timing driven add congestion click on terminal to get the result.

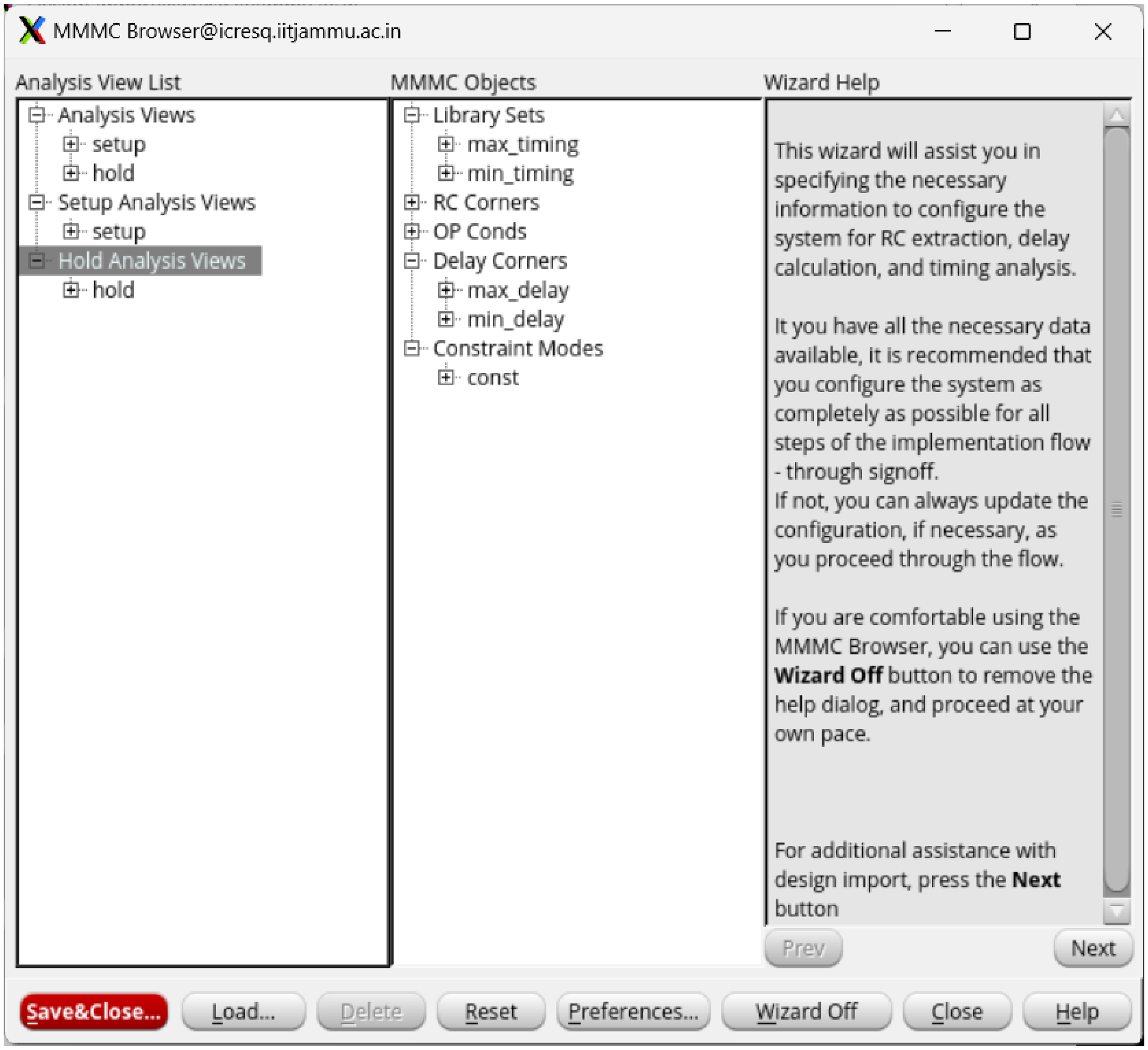
Step 11 : Go to placement add the physical cell add the filler(Select all filters) the fillers get added on the vacant space.

Step 12: Go to ECO,optimization Design post route add ok.On window we get the optimized design.

Step 13: Go to file save the design in GDS2 with stream.out format and our physcial design is completed.

8 Conclusion:

The Moore Finite State Machine was successfully designed and implemented using the ASIC design flow. With output depending only on the current state, the Moore FSM provided a stable and predictable response, making it suitable for timing-critical applications. Using



This wizard will assist you in specifying the necessary information to configure the system for RC extraction, delay calculation, and timing analysis.

It you have all the necessary data available, it is recommended that you configure the system as completely as possible for all steps of the implementation flow - through signoff.
If not, you can always update the configuration, if necessary, as you proceed through the flow.

If you are comfortable using the MMMC Browser, you can use the **Wizard Off** button to remove the help dialog, and proceed at your own pace.

For additional assistance with design import, press the **Next** button

Figure 13: Multi-Mode Multi-Corner (MMMC) Setup

tools like Genus for synthesis and Innovus for physical design, the RTL code was translated into a gate-level netlist and optimized for area, timing, and power. This project enhanced understanding of state-based digital design and demonstrated how FSMs are efficiently realized in silicon through standard ASIC design methodologies.

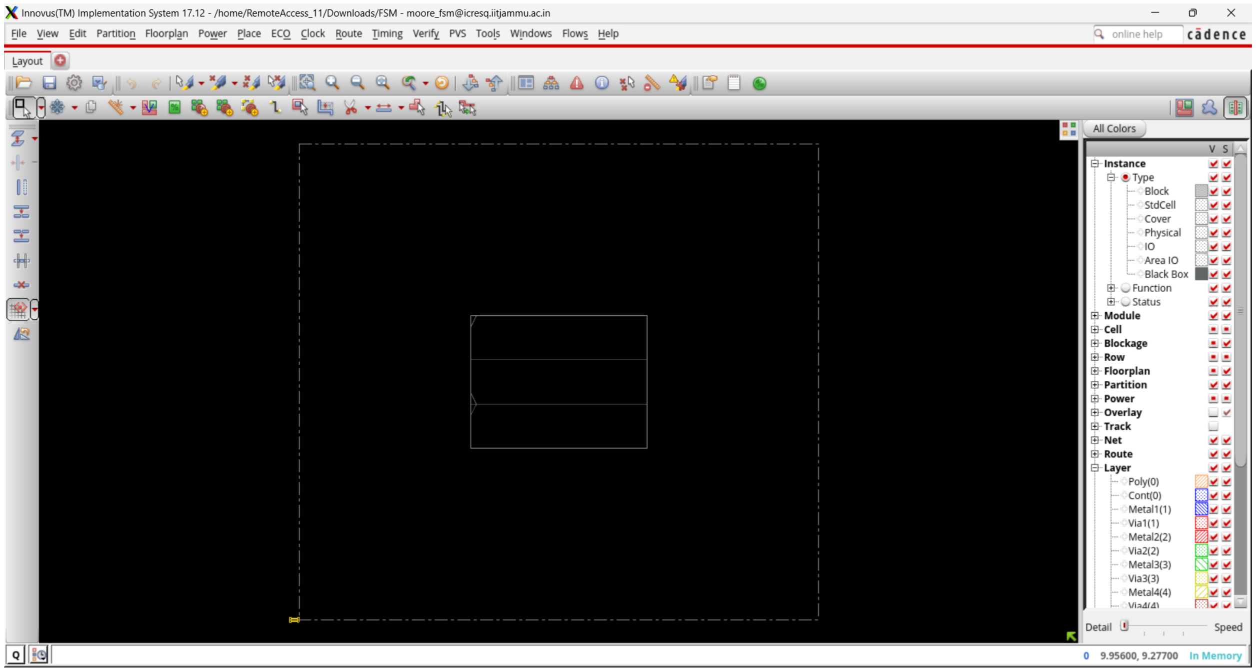


Figure 14: import design

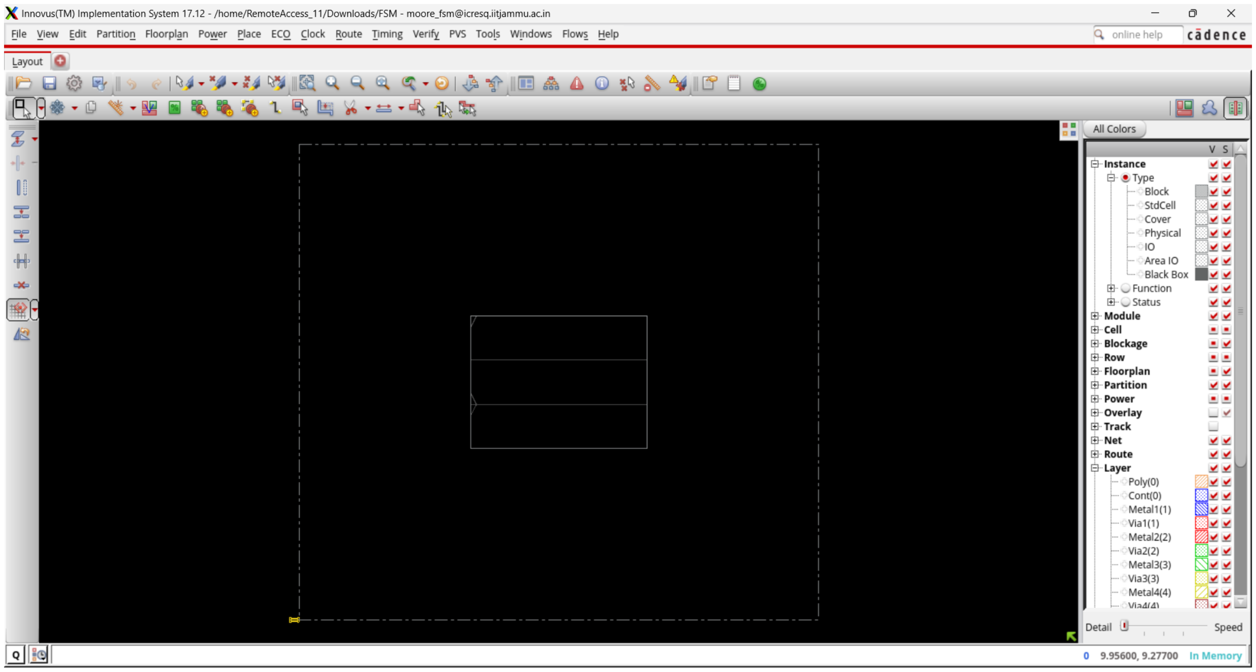


Figure 15: Floorplanning of FSM Design

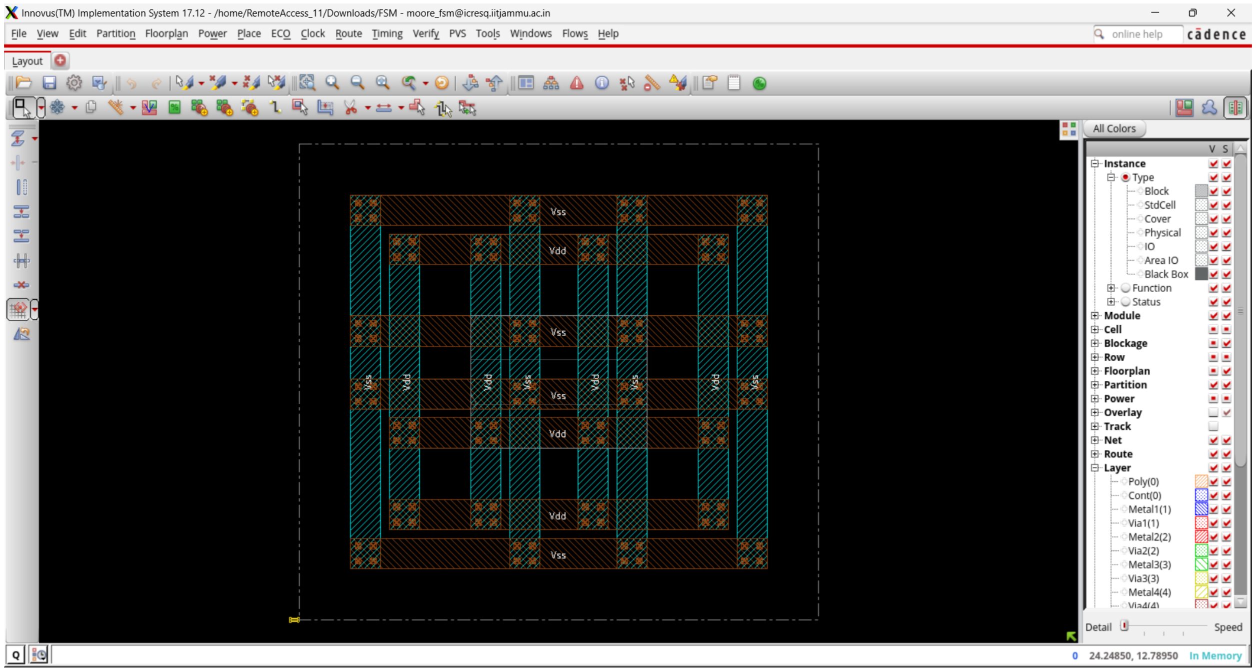
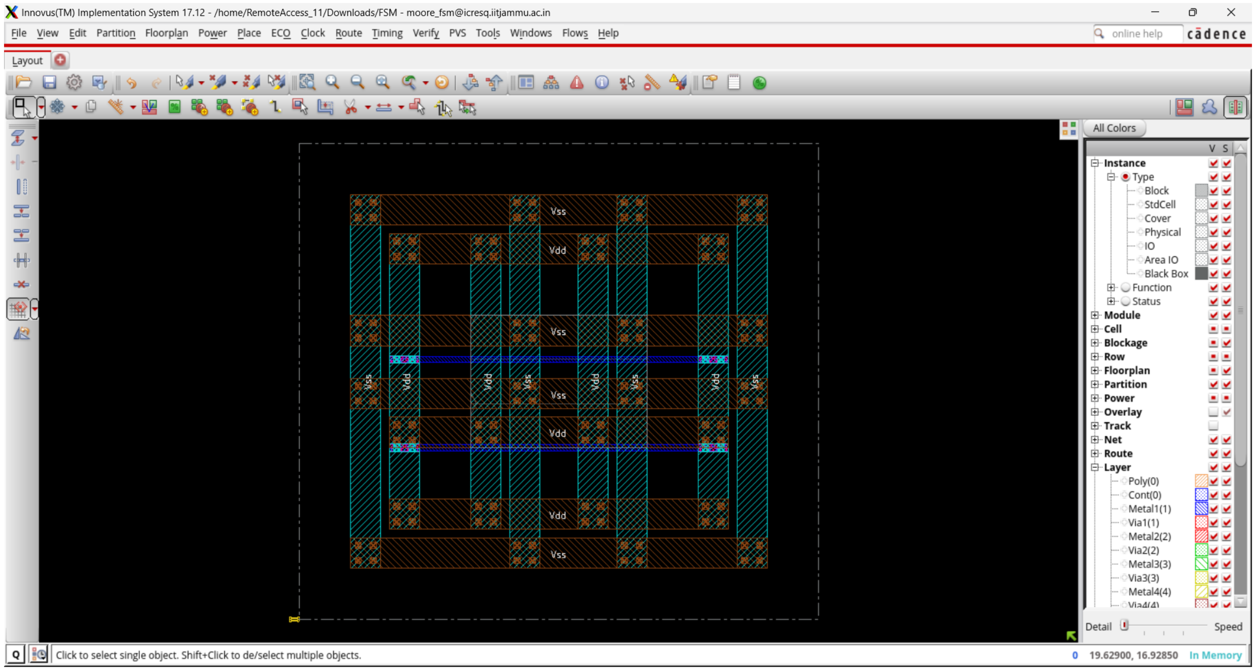
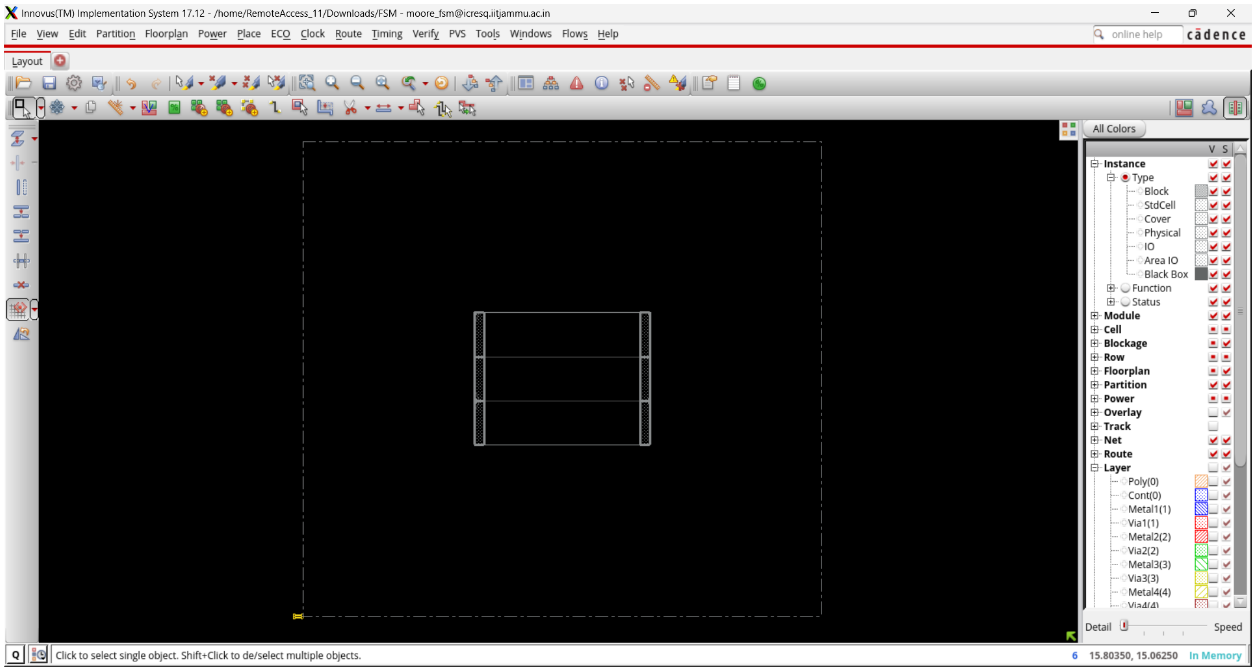


Figure 16: Power Planning



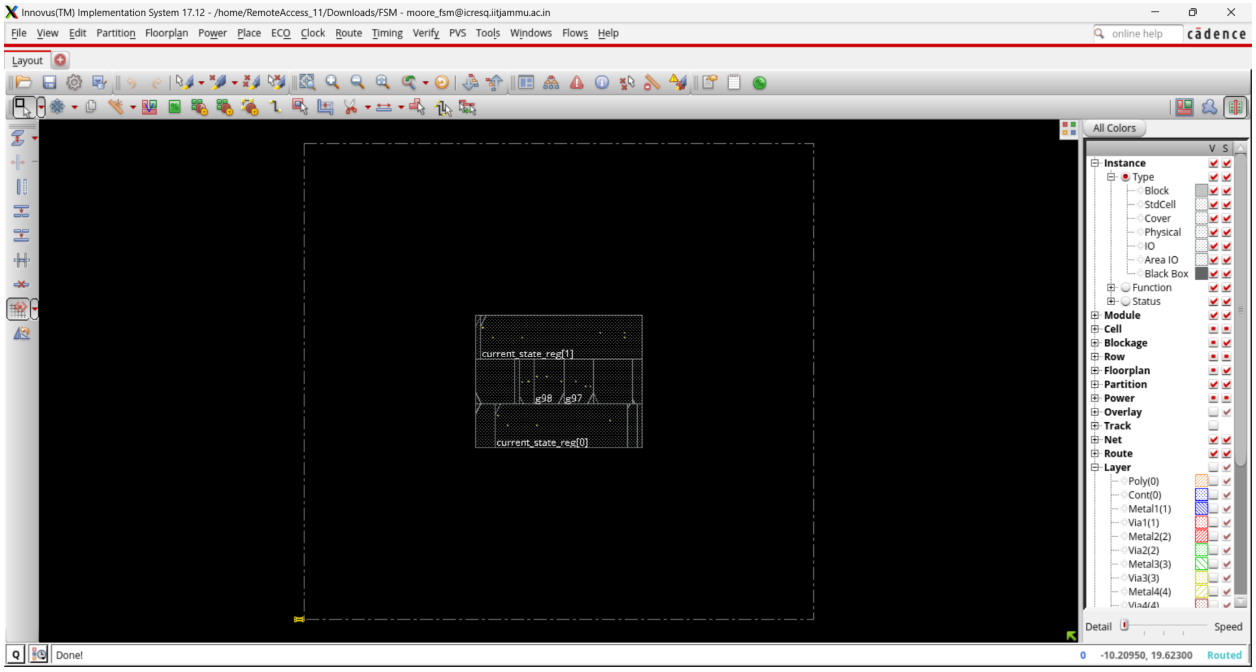
Click to select single object. Shift+Click to de/select multiple objects.

Figure 17: Speical Route



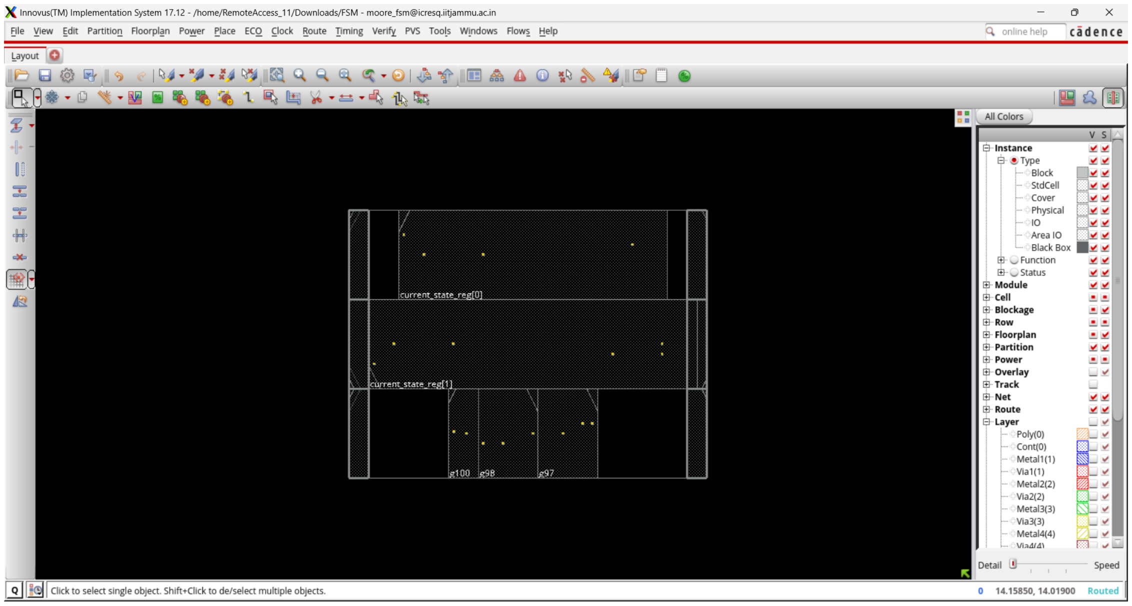
Click to select single object. Shift+Click to de/select multiple objects.

Figure 18: Physical cell encapsulation



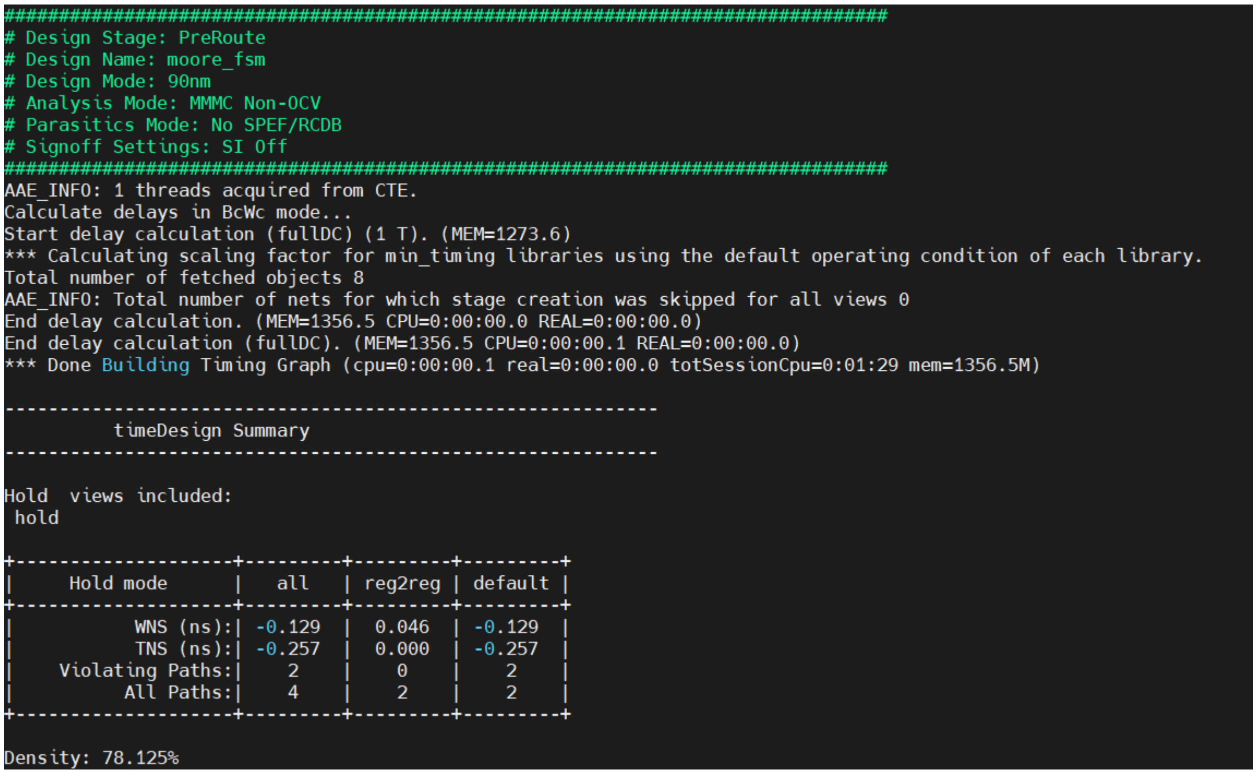
Done!

Figure 19: Add Filler



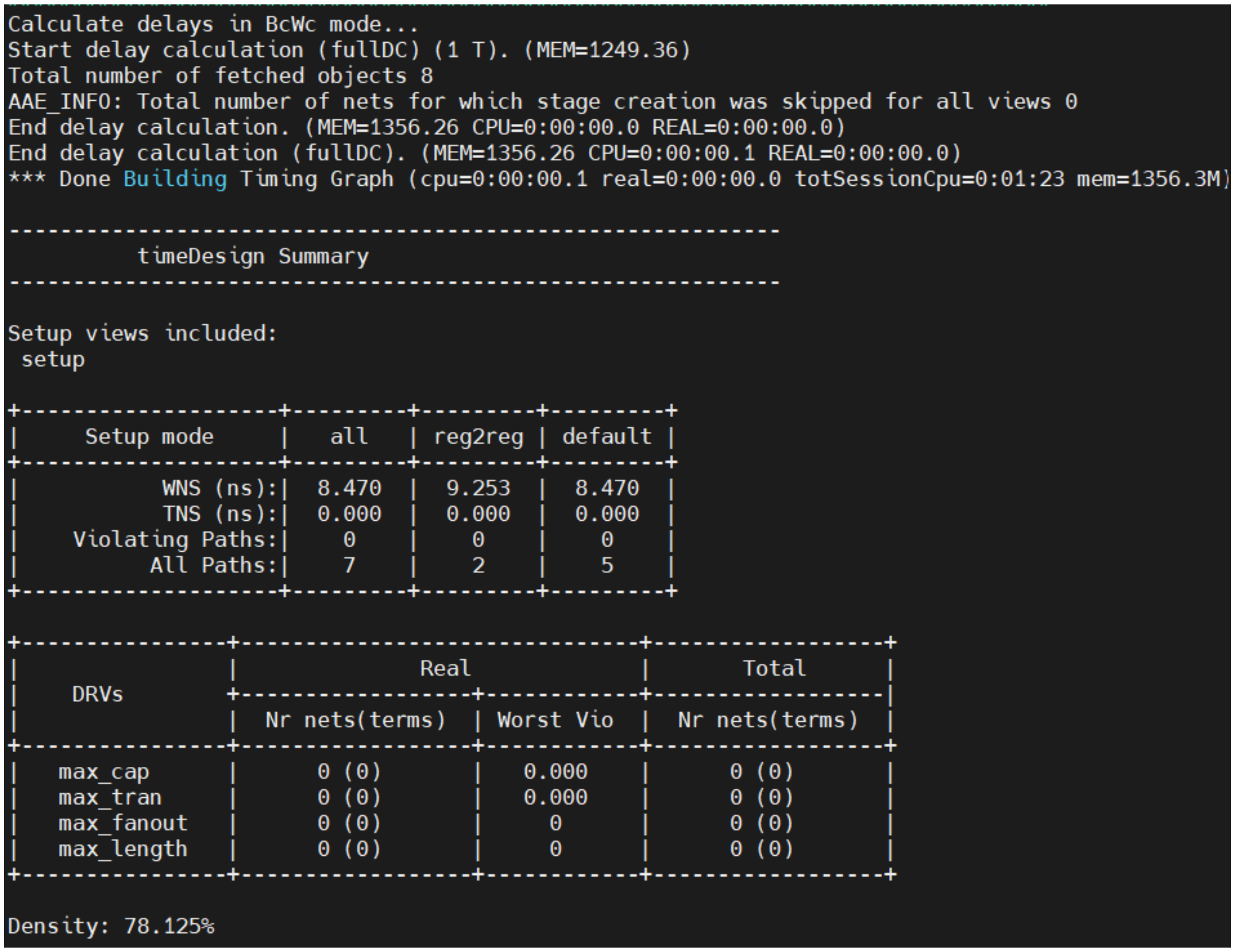
```
Click to select single object. Shift+Click to de/select multiple objects.
0    14.15850, 14.01900
```

Figure 20: Standard Cell View



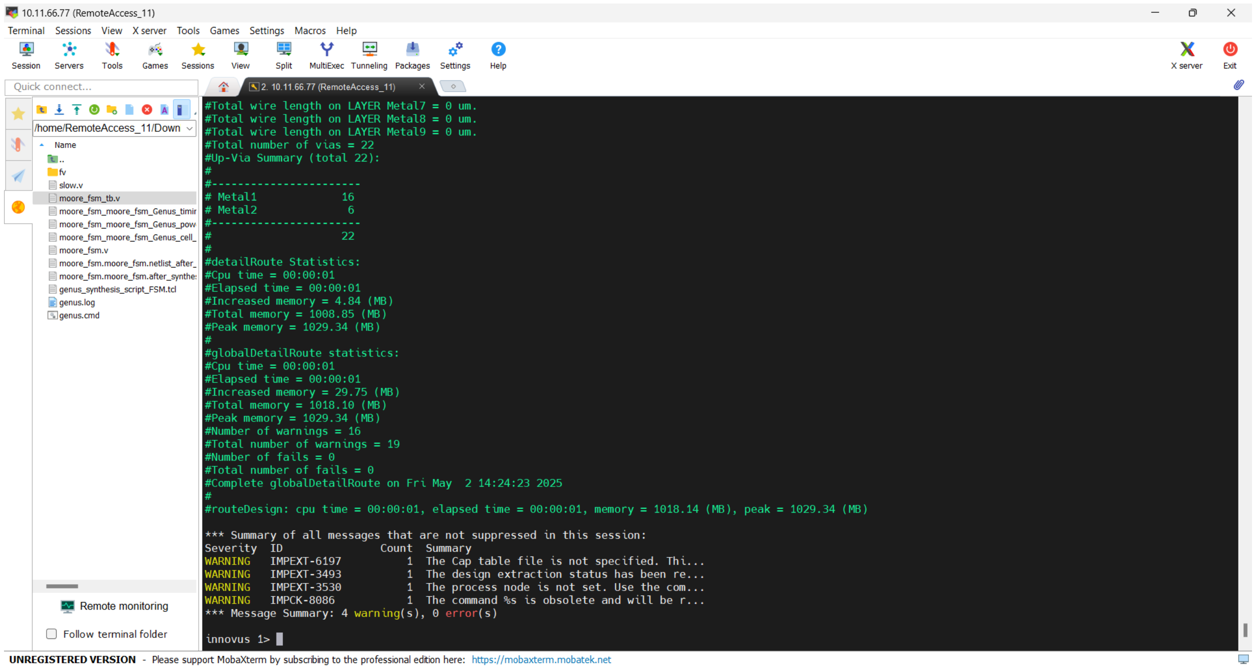
```
##################################################################################
# Design Stage: PreRoute
# Design Name: moore_fsm
# Design Mode: 90nm
# Analysis Mode: MMMC Non-OCV
# Parasitics Mode: No SPEF/RCDB
# Signoff Settings: SI Off
##################################################################################
AAE_INFO: 1 threads acquired from CTE.
Calculate delays in BcWc mode...
Start delay calculation (fullDC) (1 T). (MEM=1273.6)
*** Calculating scaling factor for min_timing libraries using the default operating condition of each library.
Total number of fetched objects 8
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
End delay calculation. (MEM=1356.5 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=1356.5 CPU=0:00:00.1 REAL=0:00:00.0)
*** Done Building Timing Graph (cpu=0:00:00.1 real=0:00:00.0 totSessionCpu=0:01:29 mem=1356.5M)

------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Hold  views included:
 hold

+--------------------+---------+---------+---------+
|     Hold mode      |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):| -0.129  |  0.046  | -0.129  |
|           TNS (ns):| -0.257  |  0.000  | -0.257  |
|    Violating Paths:|    2    |    0    |    2    |
|          All Paths:|    4    |    2    |    2    |
+--------------------+---------+---------+---------+

Density: 78.125%
```

Figure 21: pre cts hold



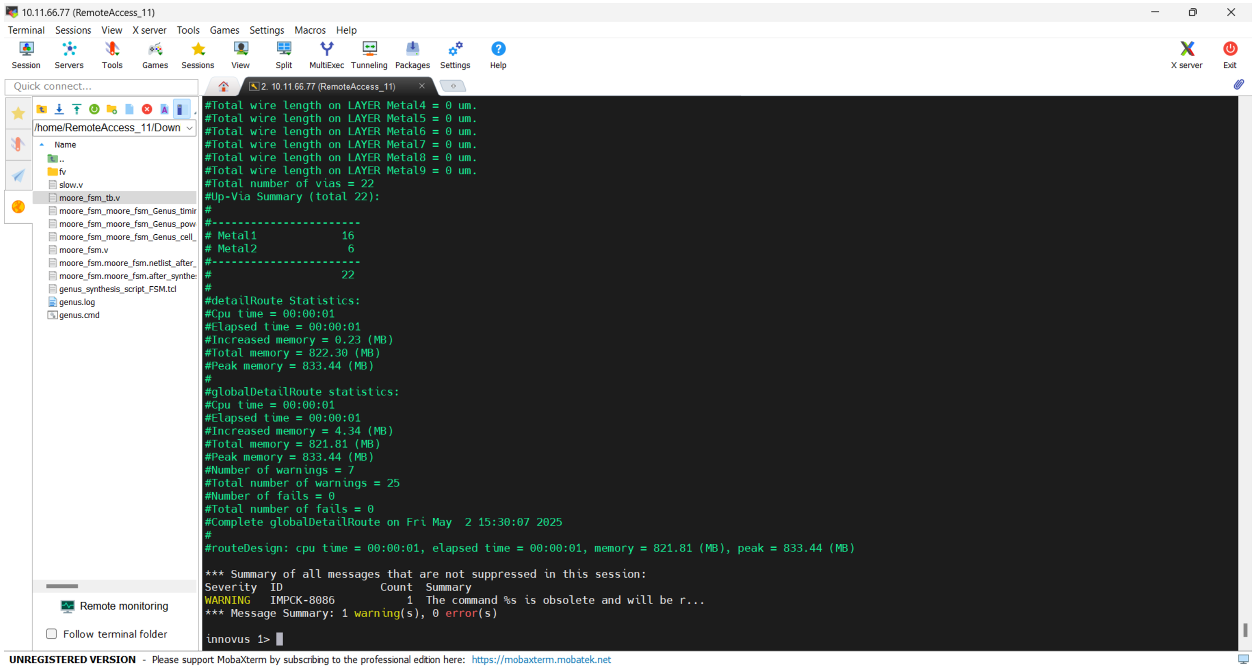
```
Calculate delays in BcWc mode...
Start delay calculation (fullDC) (1 T). (MEM=1249.36)
Total number of fetched objects 8
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
End delay calculation. (MEM=1356.26 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=1356.26 CPU=0:00:00.1 REAL=0:00:00.0)
*** Done Building Timing Graph (cpu=0:00:00.1 real=0:00:00.0 totSessionCpu=0:01:23 mem=1356.3M)

------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 setup

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  8.470  |  9.253  |  8.470  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|    7    |    2    |    5    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  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   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 78.125%
```

Figure 22: pre cts setup


10.11.66.77 (RemoteAccess_11)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 10.11.66.77 (RemoteAccess_11)
/home/RemoteAccess_11/Down
Name
..
fv
slow.v
moore_fsm_tb.v
moore_fsm_moore_fsm_Genus_timir
moore_fsm_moore_fsm_Genus_pow
moore_fsm_moore_fsm_Genus_cell_
moore_fsm.v
moore_fsm.moore_fsm.netlist_after_
moore_fsm.moore_fsm.after_synthe
genus_synthesis_script_FSM.tcl
genus.log
genus.cmd
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
#Total wire length on LAYER Metal7 = 0 um.
#Total wire length on LAYER Metal8 = 0 um.
#Total wire length on LAYER Metal9 = 0 um.
#Total number of vias = 22
#Up-Via Summary (total 22):
#
#-----------------------
# Metal1             16
# Metal2              6
#-----------------------
#                    22
#
#detailRoute Statistics:
#Cpu time = 00:00:01
#Elapsed time = 00:00:01
#Increased memory = 4.84 (MB)
#Total memory = 1008.85 (MB)
#Peak memory = 1029.34 (MB)
#
#globalDetailRoute statistics:
#Cpu time = 00:00:01
#Elapsed time = 00:00:01
#Increased memory = 29.75 (MB)
#Total memory = 1018.10 (MB)
#Peak memory = 1029.34 (MB)
#Number of warnings = 16
#Total number of warnings = 19
#Number of fails = 0
#Total number of fails = 0
#Complete globalDetailRoute on Fri May  2 14:24:23 2025
#
#routeDesign: cpu time = 00:00:01, elapsed time = 00:00:01, memory = 1018.14 (MB), peak = 1029.34 (MB)

*** Summary of all messages that are not suppressed in this session:
Severity  ID               Count  Summary
WARNING   IMPEXT-6197          1  The Cap table file is not specified. Thi...
WARNING   IMPEXT-3493          1  The design extraction status has been re...
WARNING   IMPEXT-3530          1  The process node is not set. Use the com...
WARNING   IMPCK-8086           1  The command %s is obsolete and will be r...
*** Message Summary: 4 warning(s), 0 error(s)

innovus 1>
```

Figure 23: Timing Driven


10.11.66.77 (RemoteAccess_11)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 10.11.66.77 (RemoteAccess_11)
/home/RemoteAccess_11/Down
Name
..
fv
slow.v
moore_fsm_tb.v
moore_fsm_moore_fsm_Genus_timir
moore_fsm_moore_fsm_Genus_pow
moore_fsm_moore_fsm_Genus_cell_
moore_fsm.v
moore_fsm.moore_fsm.netlist_after_
moore_fsm.moore_fsm.after_synthe
genus_synthesis_script_FSM.tcl
genus.log
genus.cmd
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
#Total wire length on LAYER Metal4 = 0 um.
#Total wire length on LAYER Metal5 = 0 um.
#Total wire length on LAYER Metal6 = 0 um.
#Total wire length on LAYER Metal7 = 0 um.
#Total wire length on LAYER Metal8 = 0 um.
#Total wire length on LAYER Metal9 = 0 um.
#Total number of vias = 22
#Up-Via Summary (total 22):
#
#-----------------------
# Metal1             16
# Metal2              6
#-----------------------
#                    22
#
#detailRoute Statistics:
#Cpu time = 00:00:01
#Elapsed time = 00:00:01
#Increased memory = 0.23 (MB)
#Total memory = 822.30 (MB)
#Peak memory = 833.44 (MB)
#
#globalDetailRoute statistics:
#Cpu time = 00:00:01
#Elapsed time = 00:00:01
#Increased memory = 4.34 (MB)
#Total memory = 821.81 (MB)
#Peak memory = 833.44 (MB)
#Number of warnings = 7
#Total number of warnings = 25
#Number of fails = 0
#Total number of fails = 0
#Complete globalDetailRoute on Fri May  2 15:30:07 2025
#
#routeDesign: cpu time = 00:00:01, elapsed time = 00:00:01, memory = 821.81 (MB), peak = 833.44 (MB)

*** Summary of all messages that are not suppressed in this session:
Severity  ID               Count  Summary
WARNING   IMPCK-8086           1  The command %s is obsolete and will be r...
*** Message Summary: 1 warning(s), 0 error(s)

innovus 1>
```

Figure 24: power Driven

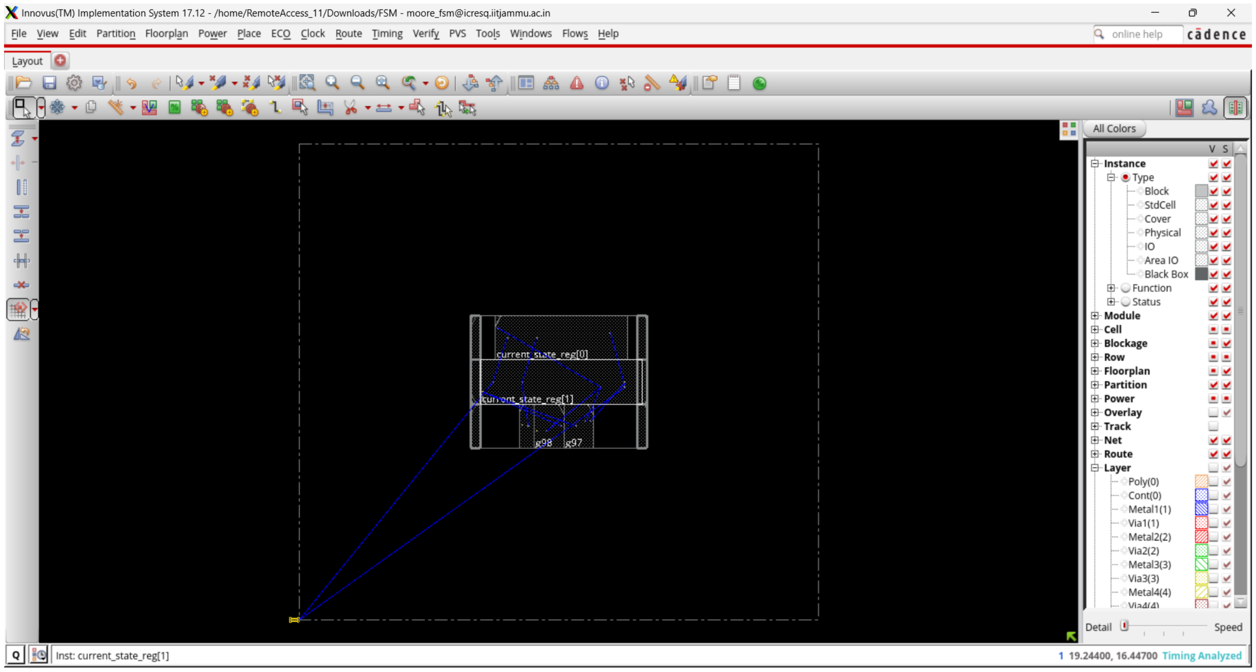
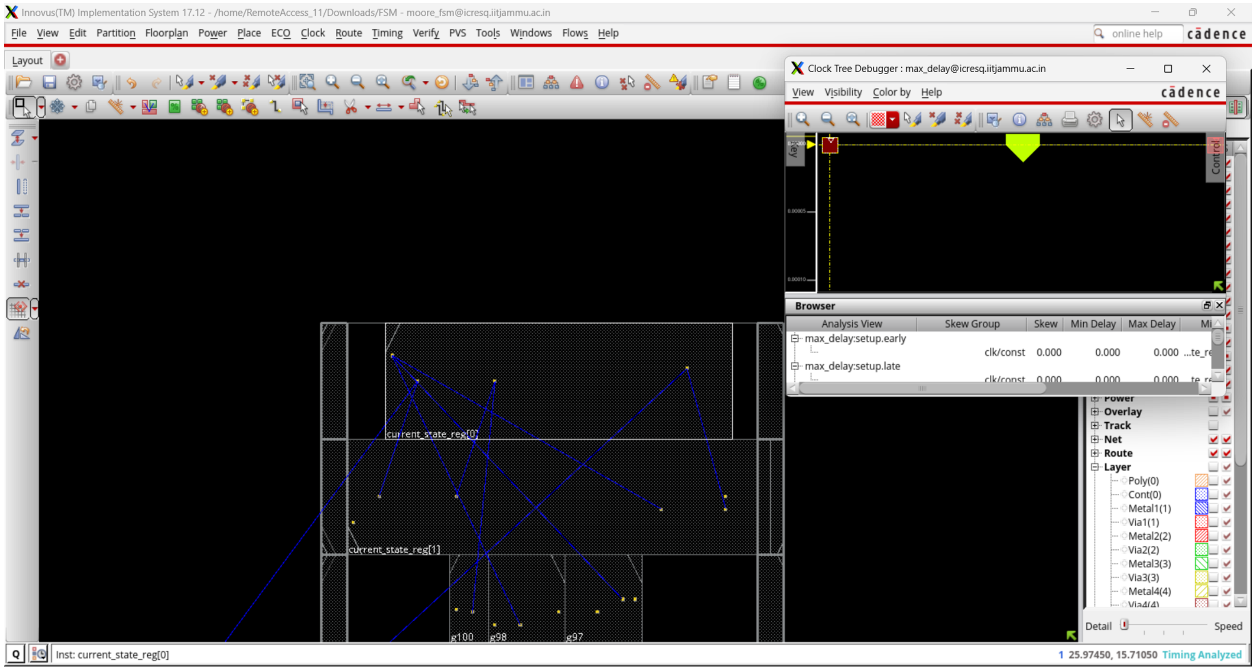
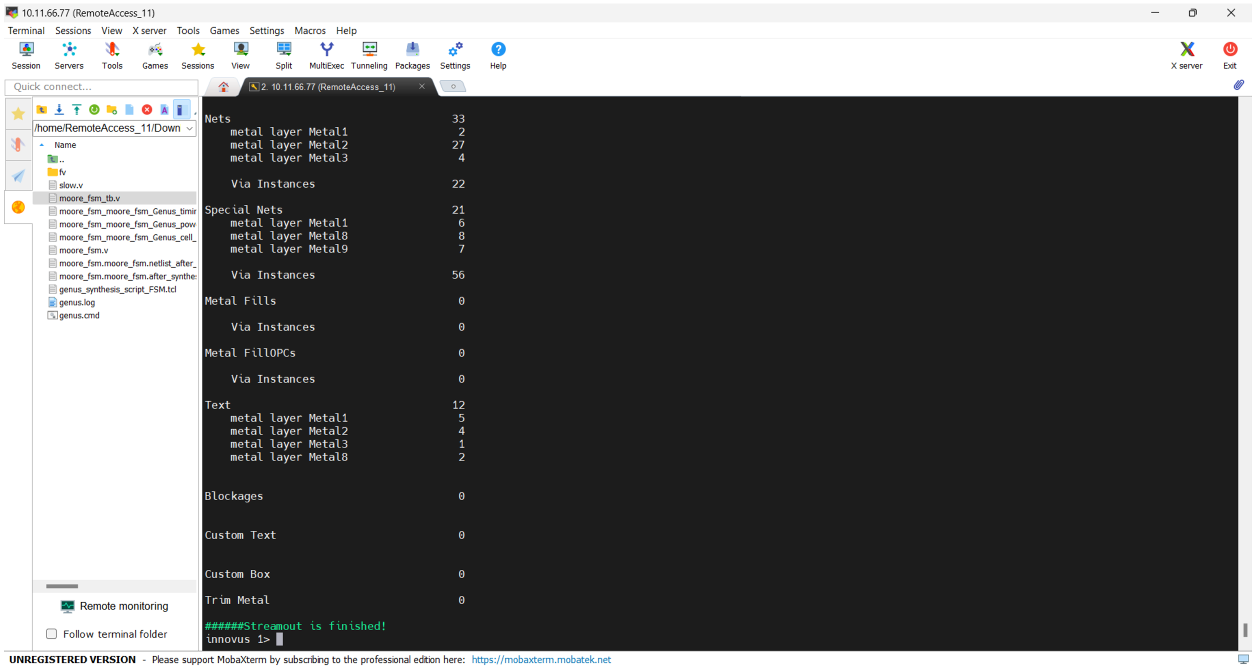


Figure 25: clock debugger


Innovus(TM) Implementation System 17.12 - /home/RemoteAccess_11/Downloads/FSM - moore_fsm@icresq.iitjammu.ac.in
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify PVS Tools Windows Flows Help
Clock Tree Debugger : max_delay@icresq.iitjammu.ac.in
View Visibility Color by Help
Browser
current_state_reg[0]
current_state_reg[1]
g100
g98
g97
Inst: current_state_reg[0]
1 25.97450, 15.71050 Timing Analyzed

Analysis View	Skew Group	Skew	Min Delay	Max Delay	Mi
max_delay:setup.early					
	clk/const	0.000	0.000	0.000	...te_re
max_delay:setup.late					
	clk/const	0.000	0.000	0.000	...te_re

Figure 26: Connecting Clock


10.11.66.77 (RemoteAccess_11)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 10.11.66.77 (RemoteAccess_11)
/home/RemoteAccess_11/Down
Name
..
fv
slow.v
moore_fsm_tb.v
moore_fsm_moore_fsm_Genus_timir
moore_fsm_moore_fsm_Genus_pow
moore_fsm_moore_fsm_Genus_cell_
moore_fsm.v
moore_fsm.moore_fsm.netlist_after_
moore_fsm.moore_fsm.after_synthe
genus_synthesis_script_FSM.tcl
genus.log
genus.cmd
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
Nets                                33
    metal layer Metal1               2
    metal layer Metal2              27
    metal layer Metal3               4

    Via Instances                   22

Special Nets                        21
    metal layer Metal1               6
    metal layer Metal8               8
    metal layer Metal9               7

    Via Instances                   56

Metal Fills                          0

    Via Instances                    0

Metal FillOPCs                       0

    Via Instances                    0

Text                                12
    metal layer Metal1               5
    metal layer Metal2               4
    metal layer Metal3               1
    metal layer Metal8               2


Blockages                            0


Custom Text                          0


Custom Box                           0

Trim Metal                           0

######Streamout is finished!
innovus 1>
```

Figure 27: Post Optimization

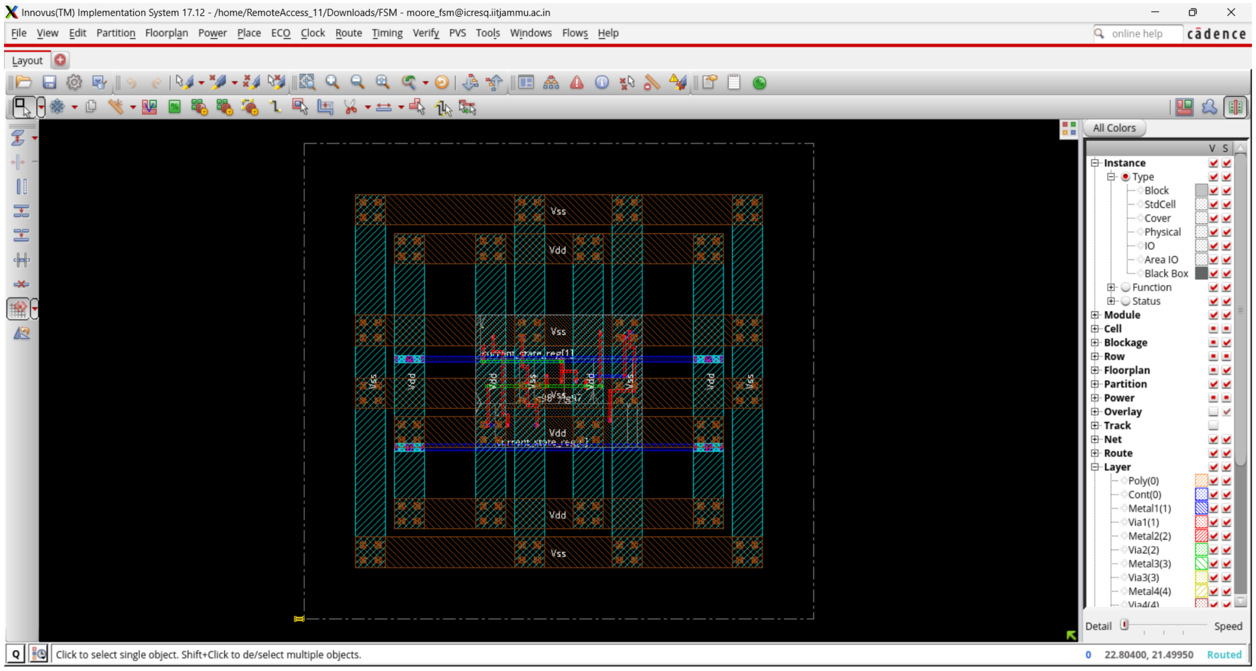
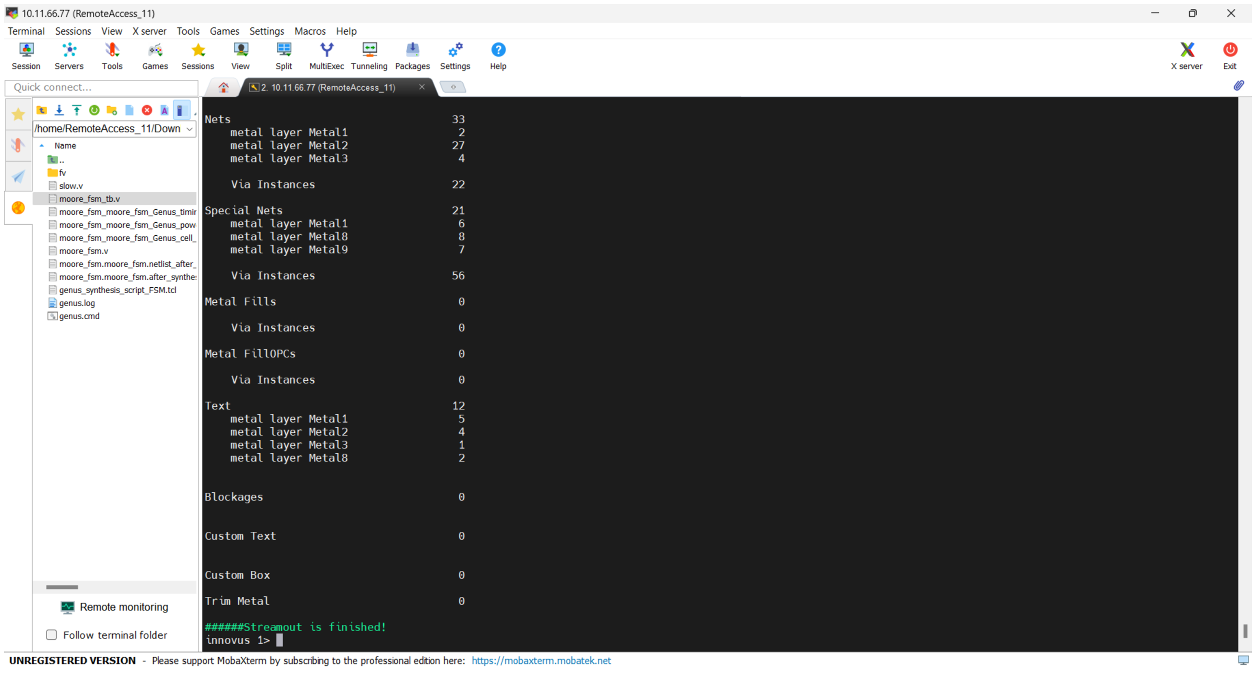


Figure 28: Final Design


10.11.66.77 (RemoteAccess_11)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. 10.11.66.77 (RemoteAccess_11)
/home/RemoteAccess_11/Down
Name
..
fv
slow.v
moore_fsm_tb.v
moore_fsm_moore_fsm_Genus_timir
moore_fsm_moore_fsm_Genus_pow
moore_fsm_moore_fsm_Genus_cell_
moore_fsm.v
moore_fsm.moore_fsm.netlist_after_
moore_fsm.moore_fsm.after_synthe
genus_synthesis_script_FSM.tcl
genus.log
genus.cmd
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
Nets                                      33
    metal layer Metal1                     2
    metal layer Metal2                    27
    metal layer Metal3                     4

    Via Instances                         22

Special Nets                              21
    metal layer Metal1                     6
    metal layer Metal8                     8
    metal layer Metal9                     7

    Via Instances                         56

Metal Fills                                0

    Via Instances                          0

Metal FillOPCs                             0

    Via Instances                          0

Text                                      12
    metal layer Metal1                     5
    metal layer Metal2                     4
    metal layer Metal3                     1
    metal layer Metal8                     2


Blockages                                  0


Custom Text                                0


Custom Box                                 0

Trim Metal                                 0

######Streamout is finished!
innovus 1>
```

Figure 29: Post Layout Design