Sharif University of Technology
Department of Computer Engineering

System on Chip Desing Projects

End-to-End ASIC Design and Verification of a Mixed ALU Using Cadence Tools

Mahbod Afarin

January 2017

1- Introduction

Our goal in this project is to become familiar with synthesis using the **Design Compiler** tool and to analyze **post-synthesis simulation** using **ModelSim**. Next, we will get introduced to the **SoC Encounter** tool, which is used for **ASIC design** and ultimately provides us with a **GDSII file**—a file that can be sent to the foundry for chip fabrication. Finally, we will perform **post-layout simulation**. In the **second section**, we explain the synthesis process and its post-synthesis simulation. In the **third section**, we describe the layout design. And finally, in the **last section**, we examine the post-layout simulation.

2- Synthesis and Post-Synthesis Simulation

In this section, we describe the synthesis process and its post-synthesis simulation.

2-1- Circuit Synthesis Using Design Compiler

In this step, we use the Design Compiler tool to synthesize the ALU code designed in the previous assignment. The synthesis is completed successfully. A sample script used for this process is shown below.

```
set target_library "/home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db"
set link_library "/home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db"
set symbol_library "/home/icic/Desktop/test/library/tsmc/tsmc18.sdb"


set my_toplevel Comlex_ALU

    set my_input_delay_ns 0
    set my_output_delay_ns 0

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

    analyze -f verilog -library work /home/icic/Desktop/test/source/Complex_ALU.v
    analyze -f verilog -library work /home/icic/Desktop/test/source/complex_mul.v



elaborate $my_toplevel

current_design $my_toplevel


list_designs
uniquify
compile
write -h
set power_preserve_rtl_hier_names true
#rtl2saif -output Mux8.saif -design file_1_1

compile -incremental

remove_unconnected_ports -blast_buses [find -hierarchy cell "*"]

report_cell > /home/icic/Desktop/test/out/area.txt
report_area > /home/icic/Desktop/test/out/area.txt
report_port > /home/icic/Desktop/test/out/port.txt
report_timing > /home/icic/Desktop/test/out/time.txt
report_power > /home/icic/Desktop/test/out/power.txt

write -f verilog -output /home/icic/Desktop/test/out/Comlex_ALU_netlist.v -hierarchy
write sdf out/Comlex_ALU.sdf
```

Figure 1 shows the schematic of the synthesized circuit.

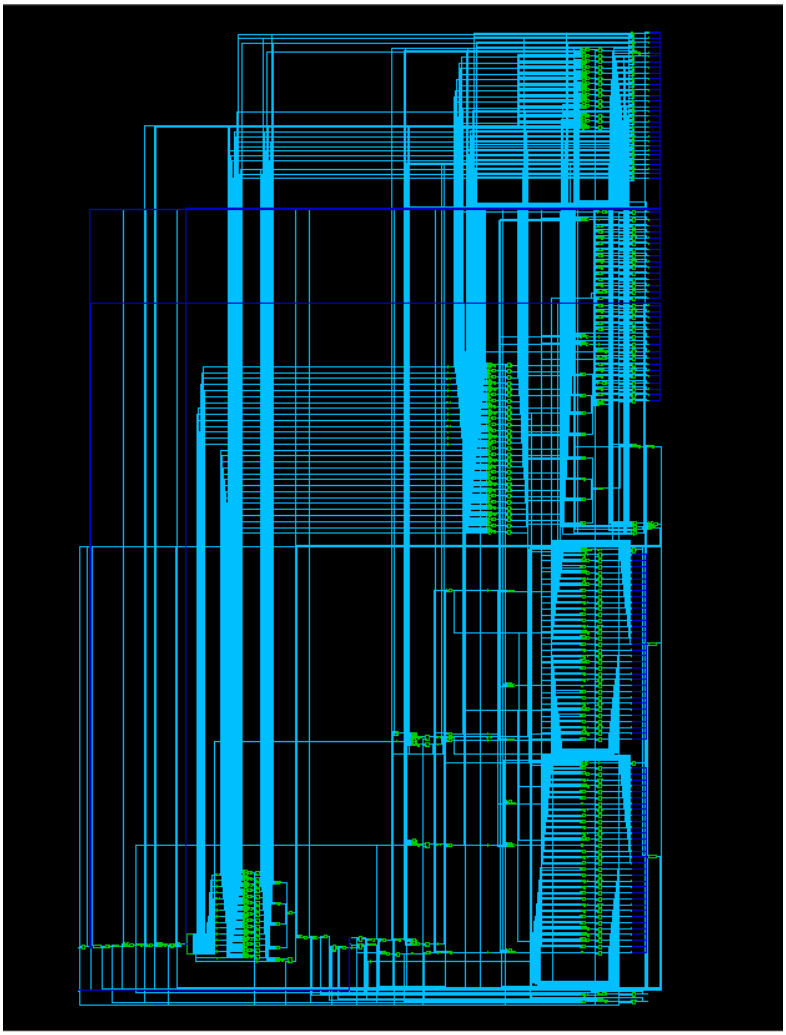

Figure 1: Schematic of the Synthesized Circuit

2-2- Generating Circuit Reports, SDF, and Output Netlist

In this step, we generate the circuit reports, which include area, delay, and power consumption. These reports are produced using the following commands in the script.

```
report_cell > /home/icic/Desktop/test/out/area.txt
report_area > /home/icic/Desktop/test/out/area.txt
report_port > /home/icic/Desktop/test/out/port.txt
report_timing > /home/icic/Desktop/test/out/time.txt
report_power > /home/icic/Desktop/test/out/power.txt
```

The results are shown in the following figures.

```
****************************************
Report : area
Design : Comlex_ALU
Version: C-2009.06-SP5
Date   : Tue Jan 31 11:48:43 2017
****************************************

Library(s) Used:

    typical (File: /home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db)

Number of ports:              88
Number of nets:              505
Number of cells:             469
Number of references:         27

Combinational area:       66015.735331
Noncombinational area:    9204.148960
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          75219.884291
Total area:                 undefined
```

Figure 2: Area Report

```
****************************************
Report : timing
         -path full
         -delay max
         -max_paths 1
Design : Comlex_ALU
Version: C-2009.06-SP5
Date   : Tue Jan 31 11:48:43 2017
****************************************

Operating Conditions: typical   Library: typical
Wire Load Model Mode: top

  Startpoint: valid_reg (rising edge-triggered flip-flop)
  Endpoint: valid (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  valid_reg/CK (DFFHQX1)                   0.00       0.00 r
  valid_reg/Q (DFFHQX1)                    0.19       0.19 r
  valid (out)                              0.00       0.19 r
  data arrival time                                   0.19
  -----------------------------------------------------------
  (Path is unconstrained)
```

Figure 3: Timing Report

```
****************************************
Report : power
        -analysis_effort low
Design : Comlex_ALU
Version: C-2009.06-SP5
Date   : Tue Jan 31 11:48:45 2017
****************************************


Library(s) Used:

    typical (File: /home/icic/Desktop/test/library/tsmc/tsmc_0.18u.db)


Operating Conditions: typical   Library: typical
Wire Load Model Mode: top


Global Operating Voltage = 1.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 592.5060 uW   (92%)
  Net Switching Power  =  54.0776 uW    (8%)
                         ---------
Total Dynamic Power    = 646.5836 uW  (100%)
```

Figure 4: Power Report

Finally, the results are organized in **Table 1**.

Table1 : Synthesis Results

	Power			
Condition	Leakage	Dynamic	Area	Delay
typical	451.1447 nw	646.5836 uw	75219.884291	0.19 ns

In the next step, the netlist and SDF file must be generated, which is done using the following code snippet in the script.

```
write -f verilog -output
/home/icic/Desktop/test/out/Comlex_ALU_netlist.v -hierarchy
write_sdf out/Comlex_ALU.sdf
```

2-3- Considering Different Operating Conditions for Synthesis

By considering the operating conditions **low**, **typical**, and **fast**, the results are obtained as shown in **Table 2**.

Table2 : Comparison of Fast, Slow, and Typical Synthesis Results

	Power			
Condition	Leakage	Dynamic	Area	Delay
low	3.3255 uw	522.5579 uw	75219.884291	0.31 ns
typical	451.1447 nw	646.5836 uw	75219.884291	0.19 ns
fast	2.2312 uw	783.0043 uw	75219.884291	0.14 ns

These conditions are presented in the table below.

Table3 : Different operational conditions

Operating Conditions

Name	Library	Temp	Volt
Slow	TSMC 180nm	125	1.62
Typical	TSMC 180nm	25	1.8
Fast	TSMC 180nm	0	1.98

The target circuit was successfully synthesized using all three libraries (slow, typical, and fast), and the results of each synthesis were saved and analyzed separately. The results showed that there was **no noticeable difference in area** across the three conditions. This is because the circuit is relatively small, which reduces the likelihood of varying synthesis results. For instance, in the **fast** case, if there were an opportunity to reduce the number of gates in the critical path, such a synthesis would be beneficial. However, due to the small size of the circuit, only one synthesis strategy was applicable across all conditions, which explains the identical area results.

However, to **increase or decrease the circuit speed**, other parameters can be used. One such parameter is **temperature**. Lowering the temperature reduces the threshold voltage (V_th), which—based on the transistor current equation—increases the current and thus the **dynamic power consumption**. On the other hand, increasing the temperature leads to higher **leakage current**. Another way to control performance is by adjusting the **supply voltage (V_dd)**. Increasing V_dd significantly increases **dynamic power** (quadratically) and **static power** (linearly). However, a higher V_dd also boosts the circuit's current, which improves speed.

In conclusion, **as circuit speed increases, power consumption also increases**, which fully explains the observations in the results above.

2-4- Analyzing the Impact of Optimization Constraints

Optimization of **area**, **power**, and **speed** is performed using the following commands.

```
set_max_delay 0.15 -from valid_reg -to valid
set_max_area 0
set_max_dynamic_power 0
set_max_leakage_power 0
```

Table4 : Comparison of Different Optimizations

	Power			
Optimization	Leakage	Dynamic	Area	Delay
Area	*451.1447 nw*	*646.5836 uw*	*75219.884291*	*0.19 ns*
Speed	*453.4206 nw*	*631.5803 uw*	*75742.129068*	*0.15 ns*
Power	*477.2386 nw*	*619.6754 uw*	*81184.119702*	*0.24 ns*

As observed, **area optimization** results in minimized area. In **speed optimization**, the **delay is minimized**, but the **area increases**. In **power optimization**, both **area and delay increase**, but the **power consumption is reduced**.

2-5- Post-Synthesis Simulation

In this section, we perform **post-synthesis simulation** using the **netlist** and **SDF file**. For this purpose, we use the **ModelSim** software. The obtained results are shown in the figure below. As can be seen, the simulation is more realistic. This is because the **SDF file contains the timing information** of the design, making the simulation more accurate.

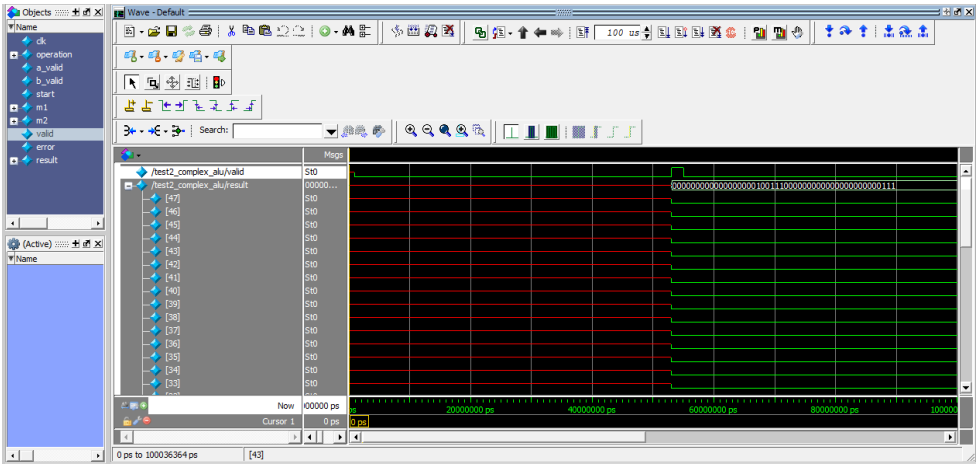

Figure 5: Post-Sysnthesis Simulation

The comparison of results shows that the post-synthesis simulation is accurate, which indicates that the synthesis stage was carried out correctly.

3- Layout Design and RC Extraction of the Circuit

In this section, we use the **Encounter** tool to perform the **layout design** of the circuit, ultimately generating a **GDSII file**. In Section 2, using the **Design Compiler (DC)** tool, we converted the high-level code into a **netlist**, resulting in a **gate-level circuit**. Essentially, this step mapped the high-level design onto the technology cells.

In this project, we feed the circuit into **SOC Encounter**, which performs **placement and routing**. The output of this step can be sent to the foundry for **IC fabrication**.

Note that in this exercise, we do **not perform optimization** using the DC tool. This is because optimization may result in the removal of certain gates, potentially leaving unconnected wires in the final layout.

The **first step** is to generate the **SDF file** for the circuit. This file is produced using the **DC tool** with the following command.

```
write_sdf out/Comlex_ALU.sdf
```

We navigate to the specified directory and provide the **netlist** generated from **DC** as input. Next, we specify the **top module name**, add the **SDC file** (which defines the timing constraints), and then add the **LEF files**. The LEF (Library Exchange Format) file contains the **physical technology information**, such as how many metal layers each cell has, their pitch, and other physical attributes.

From the **Design → Import Design** menu, we proceed to import the files. For the LEF files, we add **two**:

1. The LEF file for the **standard cells**.
2. The LEF file related to **antenna effects**.

The **antenna effect** refers to the phenomenon where long wires act like antennas, causing unwanted **parasitic capacitance** that may lead to noise or damage during fabrication. To manage this, we use an **antenna library**, which includes information about how each cell is affected by antenna rules.

Then, under the **Advanced** tab and in the **Power** section, we define the **global VDD and GND** connections for the circuit. Every standard cell needs its **VDD and VSS (GND)** defined. Since VDD and GND come from external sources, we must specify the global power rails for the entire design.

Next, as shown in the figure, we go to **Specify Floorplan**, activate the **Die Size by** option, and set a die size **larger than the default** to give enough room for placement and routing.

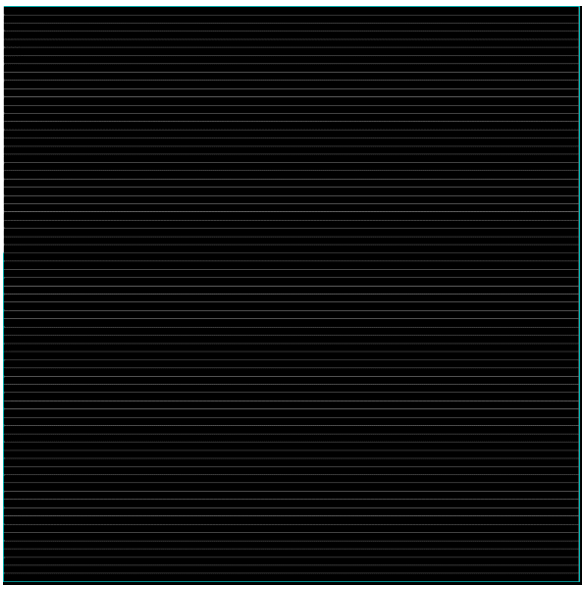

Figure 6: Steps Followed in Encounter

Next, to connect the **core to VSS and VDD**, two complete **power rings** (one for VSS and one for VDD) are created around the core. The internal VDD and VSS of the core are then connected to these rings. This approach helps to **reduce power consumption**, **increase noise immunity**, and **optimize routing**.

Therefore, in the **"Core to Left"** and **"Core to Right"** settings, a margin is added to make room for the power rings. The result of this configuration is shown in the figure below.

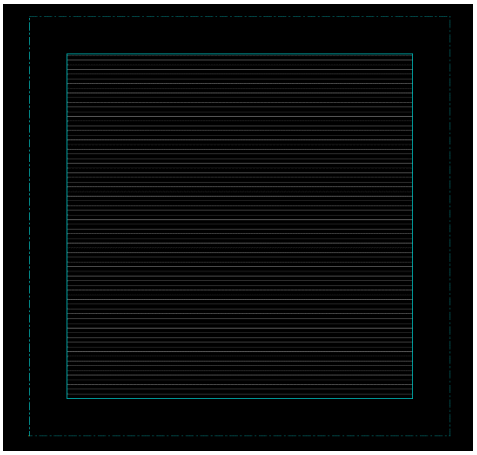

Figure 7: Steps Followed in Encounter

As shown in the figure above, a **gap has been created between the core and the die**. Next, we go to the **Connect Global Net** section and connect the **power (VDD)** and **ground (GND)** cells to the global power and ground networks.

The **tie high** concept refers to a technique in digital circuits where a node is not directly connected to VDD to represent logic '1'. This is because any **noise on the VDD line** could propagate to the node, causing instability. Instead, special cells called **tie high** and **tie low** are used. A **tie high** cell generates a noise-resistant digital logic '1', and a **tie low** cell generates a noise-resistant logic '0'.

Then, we navigate to **Power → Power Planning → Add Ring**. **Power rings** are usually implemented using **intermediate metal layers**, as they need to connect to all parts of the design. The necessary modifications are made as shown in the following figure.

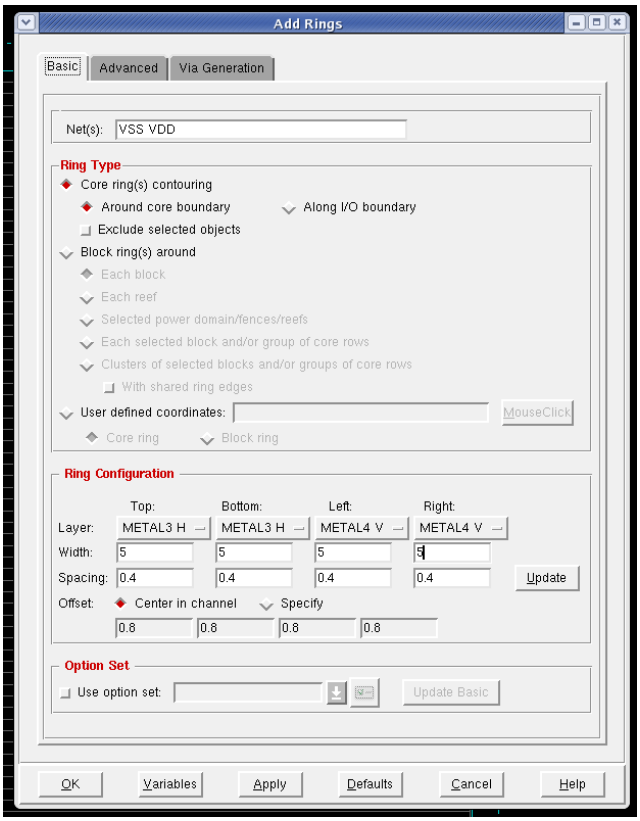

Figure 8: Steps Followed in Encounter

In the figure below, it can be seen that the **power rings have been added**.

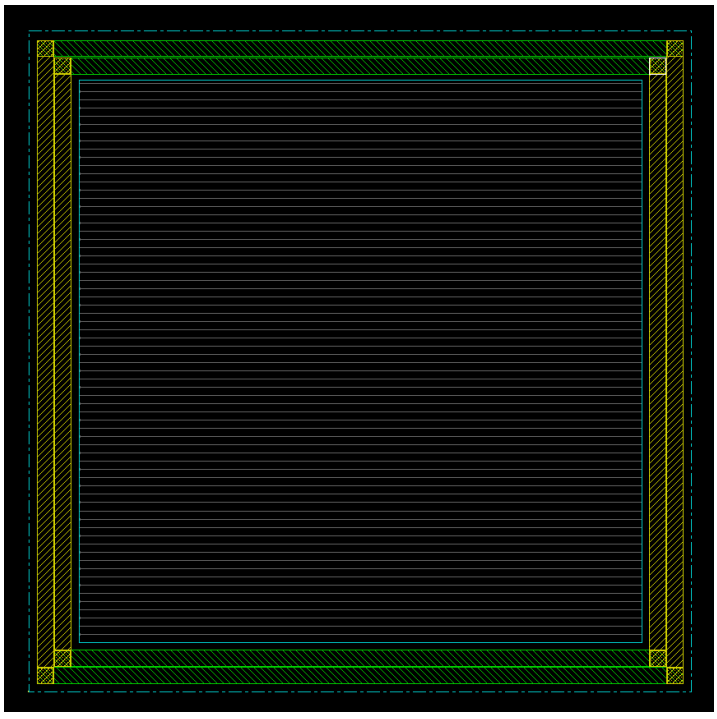

Figure 9: Steps Followed in Encounter

By zooming in, the **connections can also be observed**.

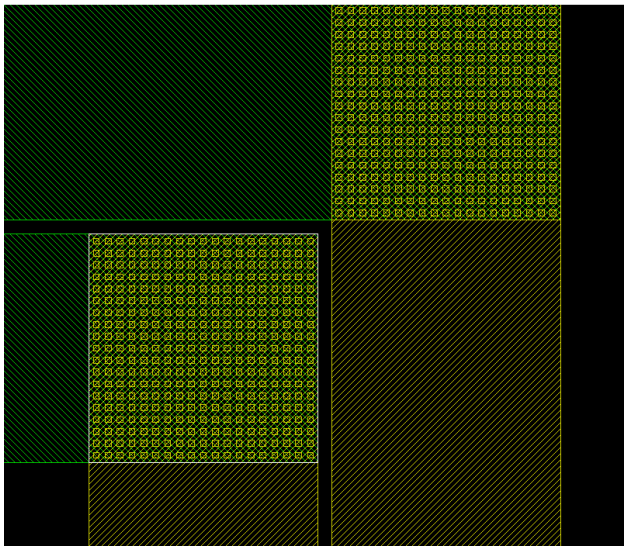

Figure 10: Steps Followed in Encounter

Next, we need to add **strips** to the circuit—this means drawing additional metal lines that are connected to **VDD and GND**. If there is a cell located in the **middle of the core** that needs to be connected to power or ground, it will connect through these strips. This ensures a **proper distribution of power lines** across the design.

This step can be done by navigating to:
Power → Power Planning → Add Strip

The figure below illustrates this process.

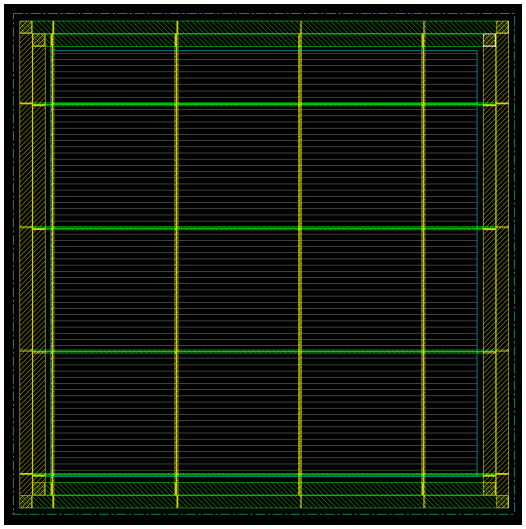

Figure 11: Steps Followed in Encounter

Next, we perform a **special route**. The output after this step is shown in the figure below.

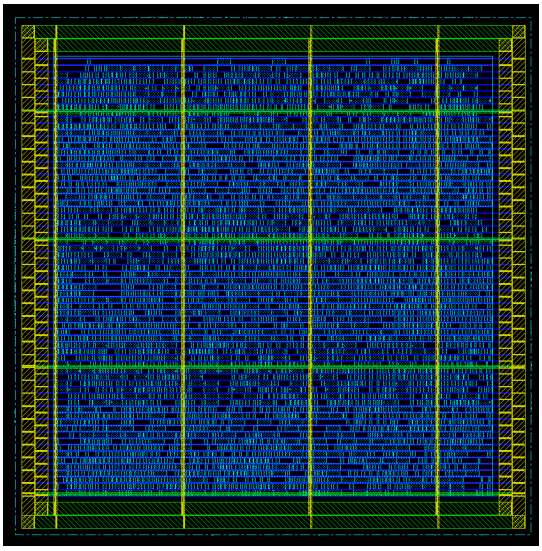

Figure 12: Steps Followed in Encounter

The next step is to perform the **final routing** and insert **metal fillers** to complete the final layout of the design.

After placing the components, we move on to **routing**. To begin with, we need to **route the clock (clk)**. For this, certain cells are added to the circuit to **prevent clock skew**, **reduce clock voltage drop**, and address other clock-related issues.

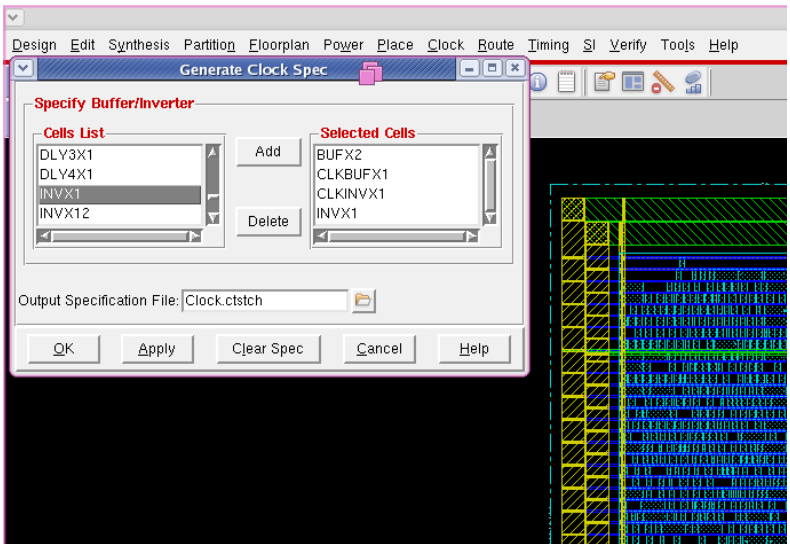

Figure 13: Steps Followed in Encounter

After that, the **clk**, **VDD**, and **VSS** signals are connected to the cells.

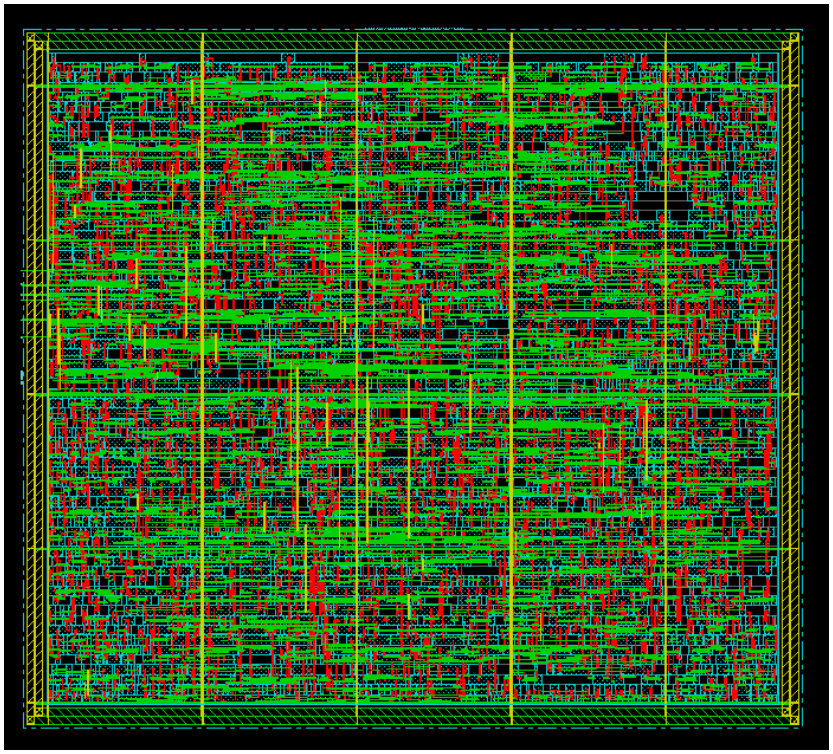

Figure 14: Steps Followed in Encounter

Now, we perform **timing analysis**.

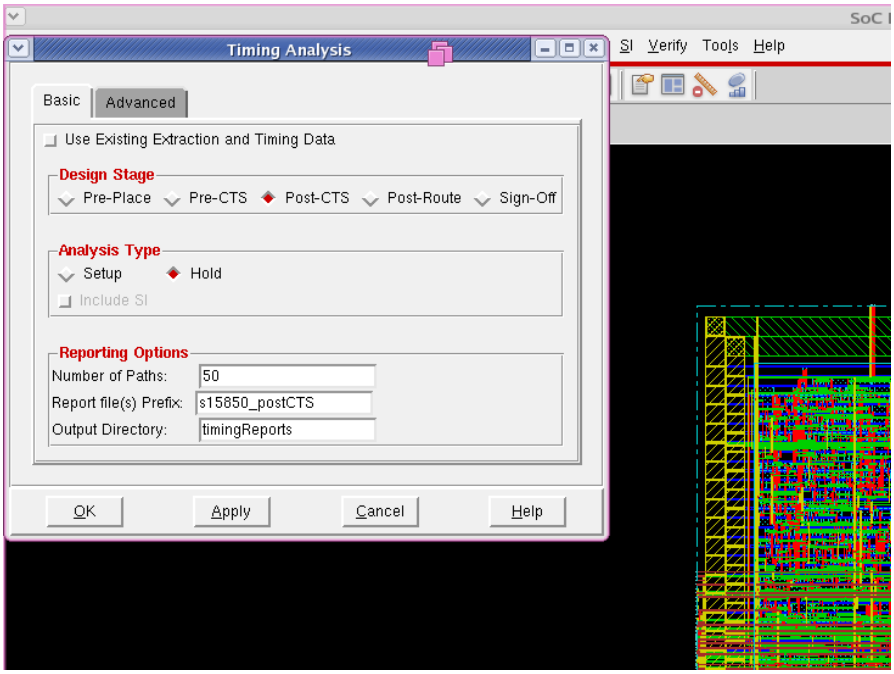

Figure 15: Steps Followed in Encounter

The results are as follows:

```
+----------------+---------------------------+---------------+
|                |           Real            |     Total     |
|    DRVs        +---------------+-----------+---------------|
|                |Nr nets(terms)| Worst Vio  |Nr nets(terms)|
+----------------+---------------+-----------+---------------+
|   max_cap      |    1 (1)      |   -0.114  |    1 (1)      |
|   max_tran     |    6 (6)      |   -3.273  |    6 (6)      |
|   max_fanout   |    0 (0)      |     0     |    0 (0)      |
+----------------+---------------+-----------+---------------+

Density: 70.394%
Routing Overflow: 0.00% H and 0.00% V
------------------------------------------------------------
Reported timing to dir timingReports
Total CPU time: 0.99 sec
Total Real time: 1.0 sec
Total Memory Usage: 245.890625 Mbytes
```

Figure 16: Steps Followed in Encounter

At this stage, we need to use **optimization techniques** to try and **improve the setup time**.

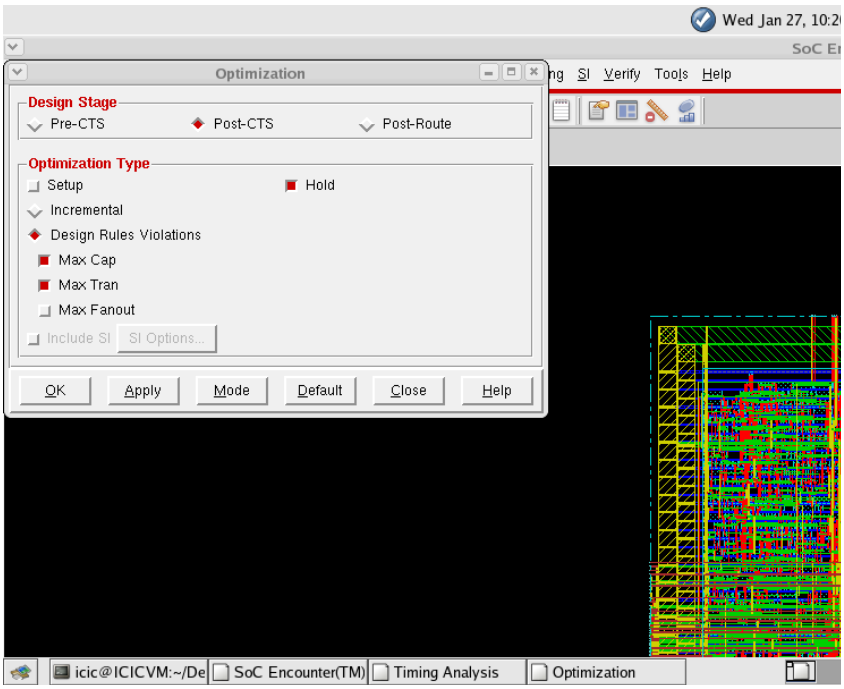

Figure 17: Steps Followed in Encounter

Now, we re-evaluate the results.

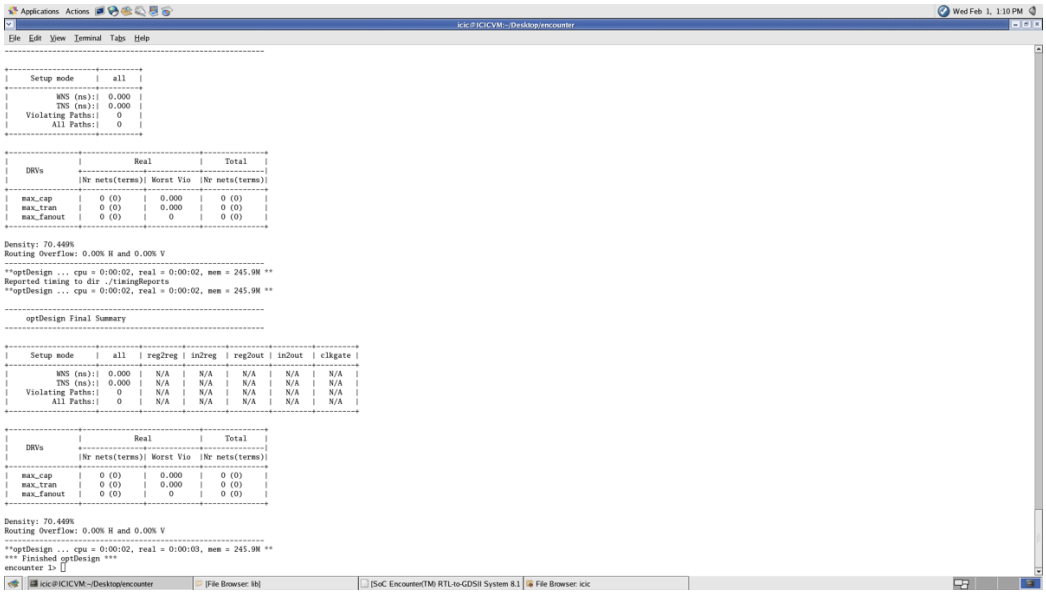

Figure 18: Steps Followed in Encounter

As shown in the image above, there are **no negative values for setup time**, indicating that the setup timing requirements are met.

Now, let's analyze the **hold time**.

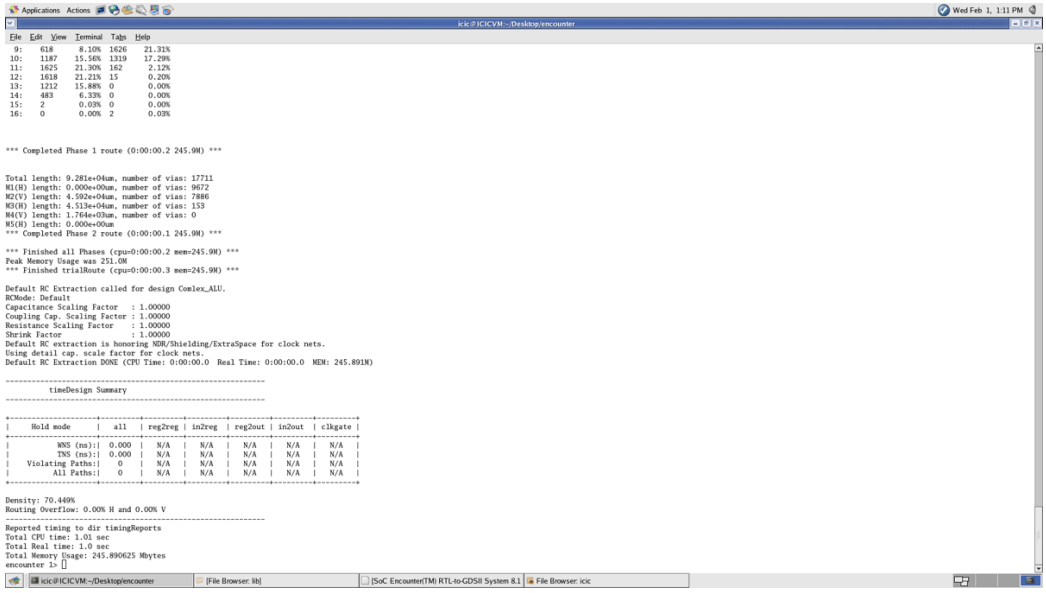

Figure 19: Steps Followed in Encounter

In the output, we observe that **there are no negative values**, which means the **hold time requirements are also satisfied**.

After this, we perform **detailed routing** of the circuit, leading to the result shown in the following figure.

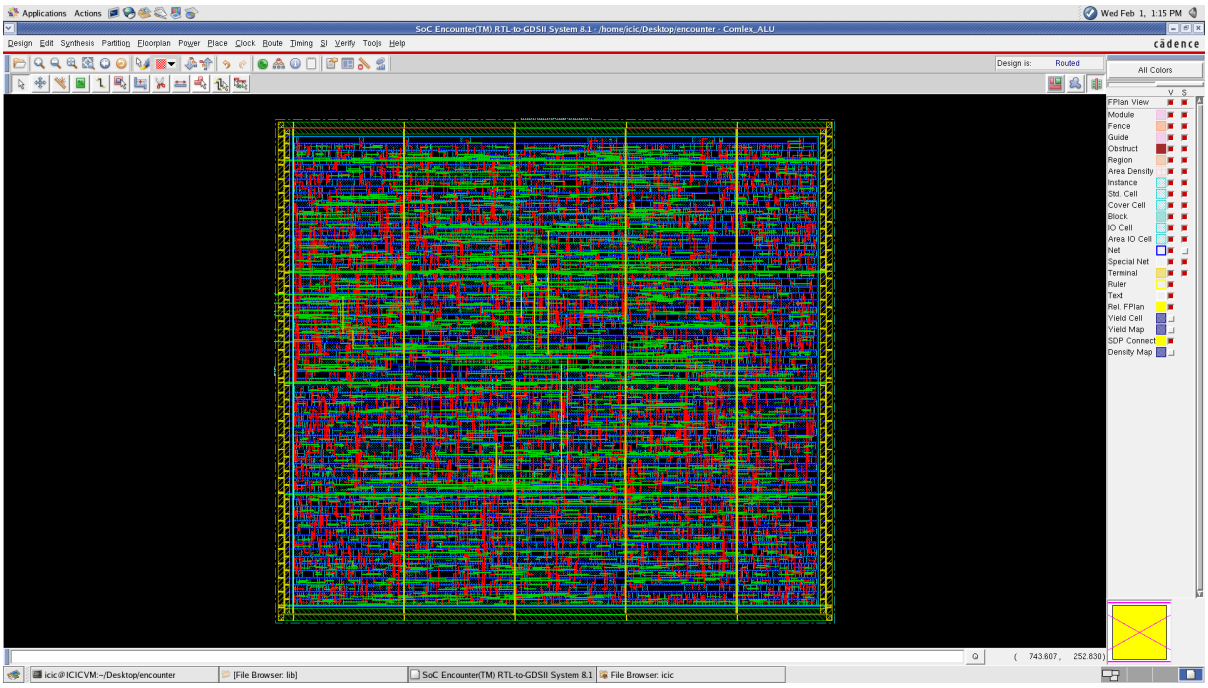

Figure 20: Steps Followed in Encounter

Close-up View of the Routing Paths:

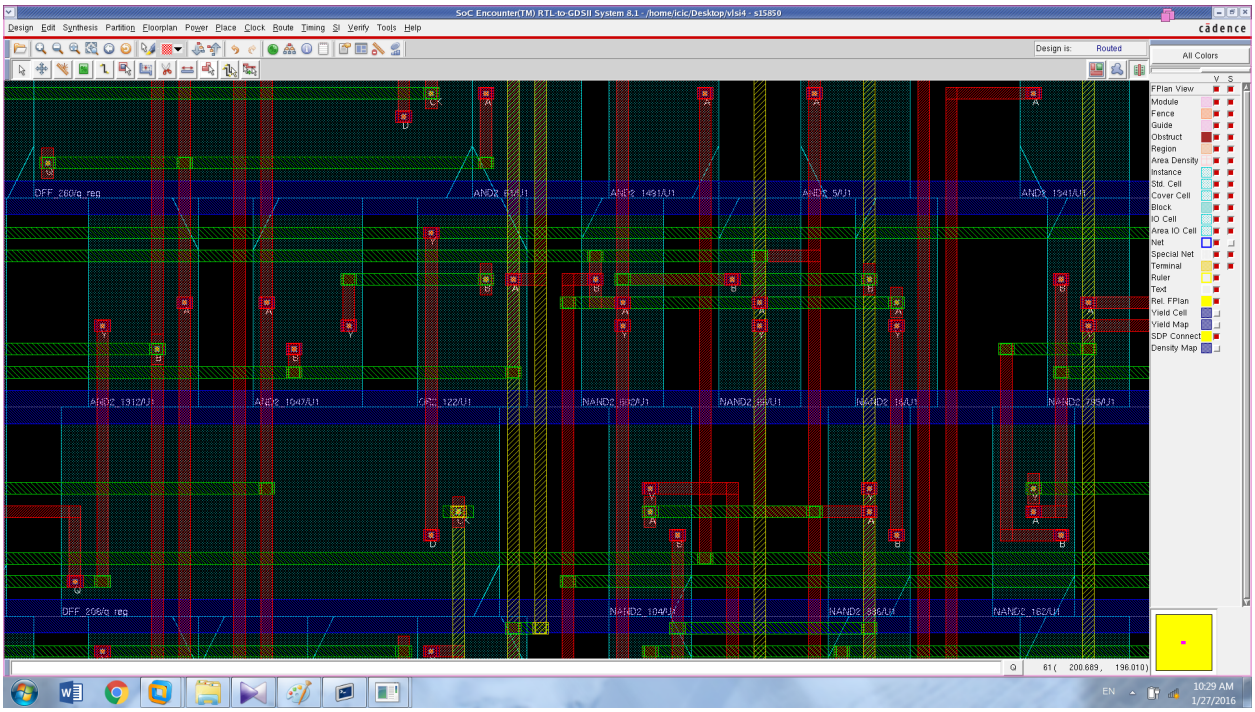

Figure 21: Steps Followed in Encounter

Next, we perform **post-route timing analysis** for the **setup time**.

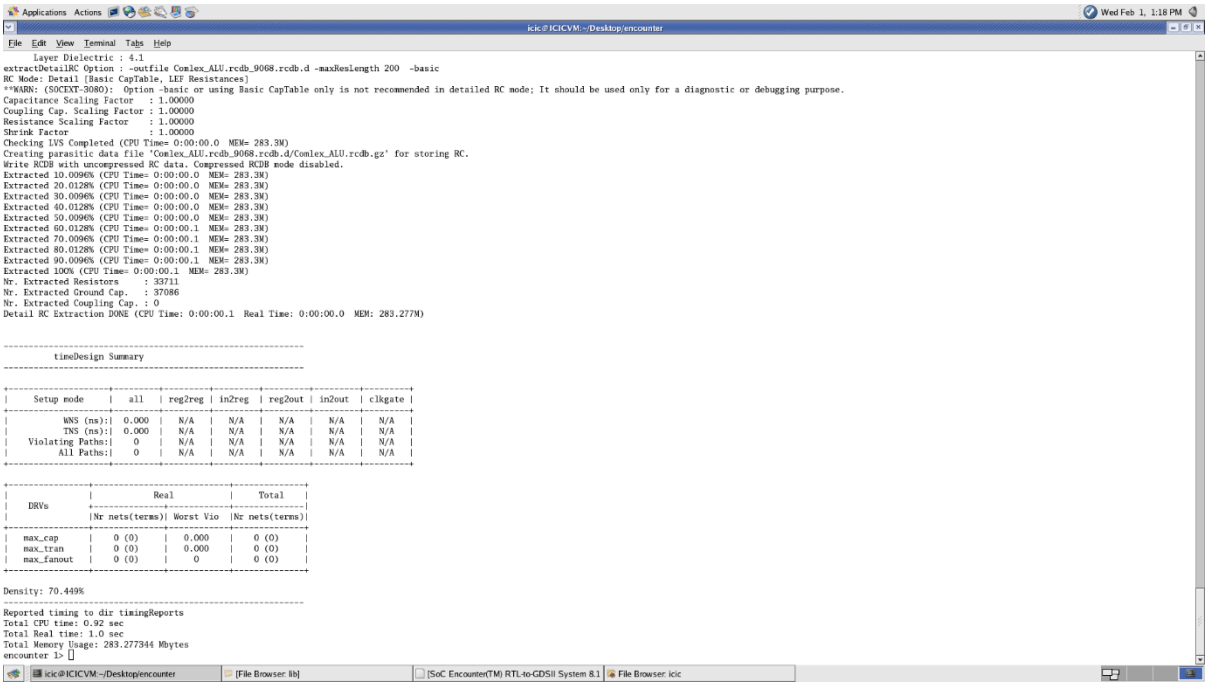

Figure 22: Steps Followed in Encounter

And we perform the **same analysis for hold time** as well.

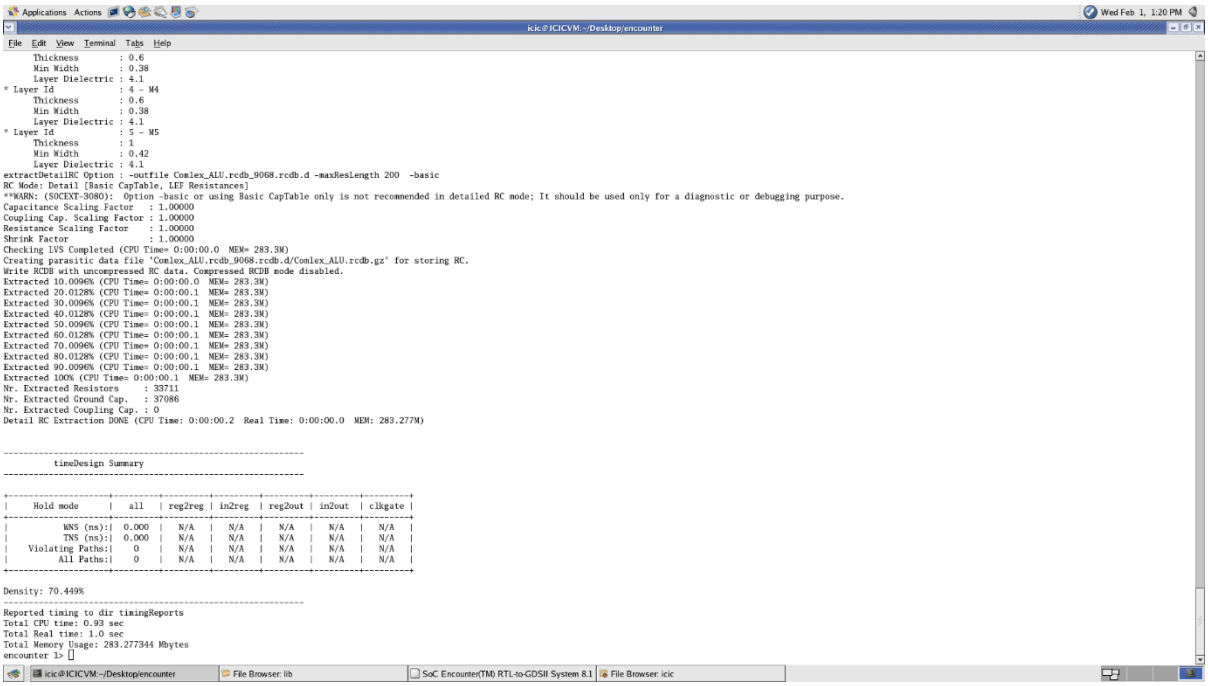

Figure 23: Steps Followed in Encounter

Next, we perform the **filler cell insertion** step.

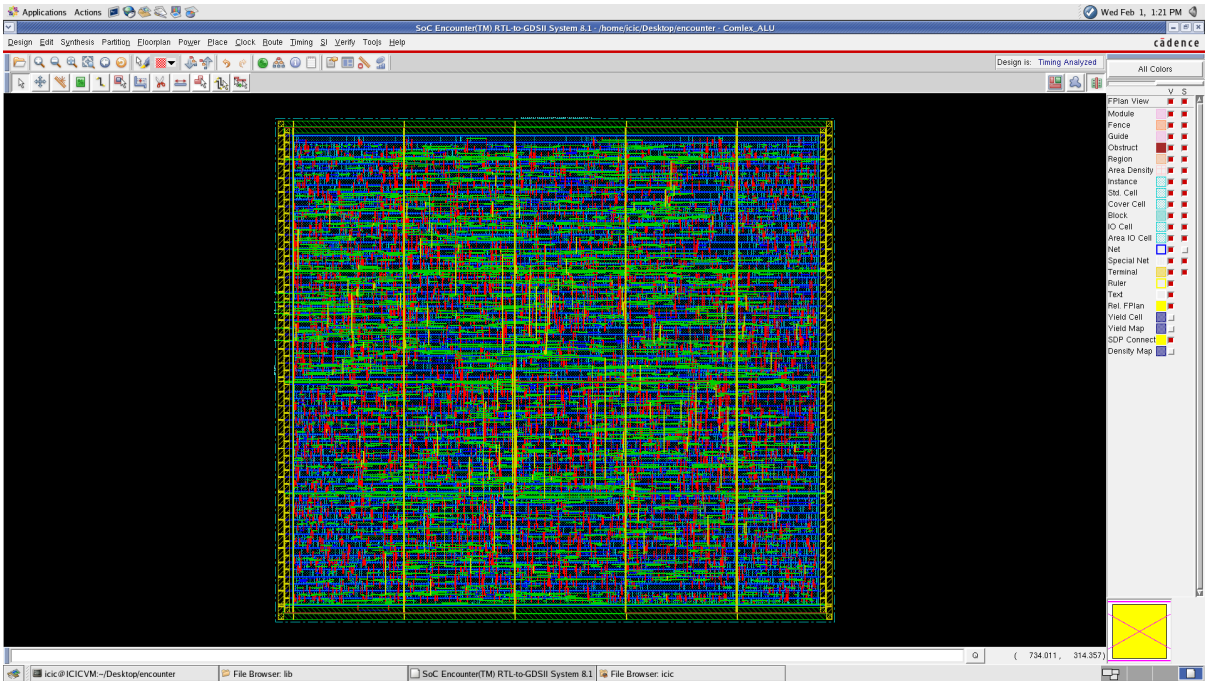

Figure 24: Steps Followed in Encounter

In the next step, we proceed to **verify the geometry and connectivity** of the design. The **positive result** of each check is shown below.

Geometry:

```
encounter 1>  *** Starting Verify Geometry (MEM: 302.4) ***

  VERIFY GEOMETRY ..... Starting Verification
  VERIFY GEOMETRY ..... Initializing
  VERIFY GEOMETRY ..... Deleting Existing Violations
  VERIFY GEOMETRY ..... Creating Sub-Areas
                  ..... bin size: 3680
  VERIFY GEOMETRY ..... SubArea : 1 of 1
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 2.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.

**********End: VERIFY GEOMETRY**********
 *** verify geometry (CPU: 0:00:01.4  MEM: 34.5M)
```

Figure 25: Steps Followed in Encounter

Connectivity:

```
encounter 1>
******** Start: VERIFY CONNECTIVITY ********
Start Time: Wed Feb  1 13:25:18 2017

Design Name: Comlex_ALU
Database Units: 1000
Design Boundary: (0.0000, 0.0000) (504.4770, 452.6000)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Wed Feb  1 13:25:18 2017
******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.2  MEM: 0.000M)
```

Figure 26: Steps Followed in Encounter

No issues are observed in the **Violation Browser** menu either.

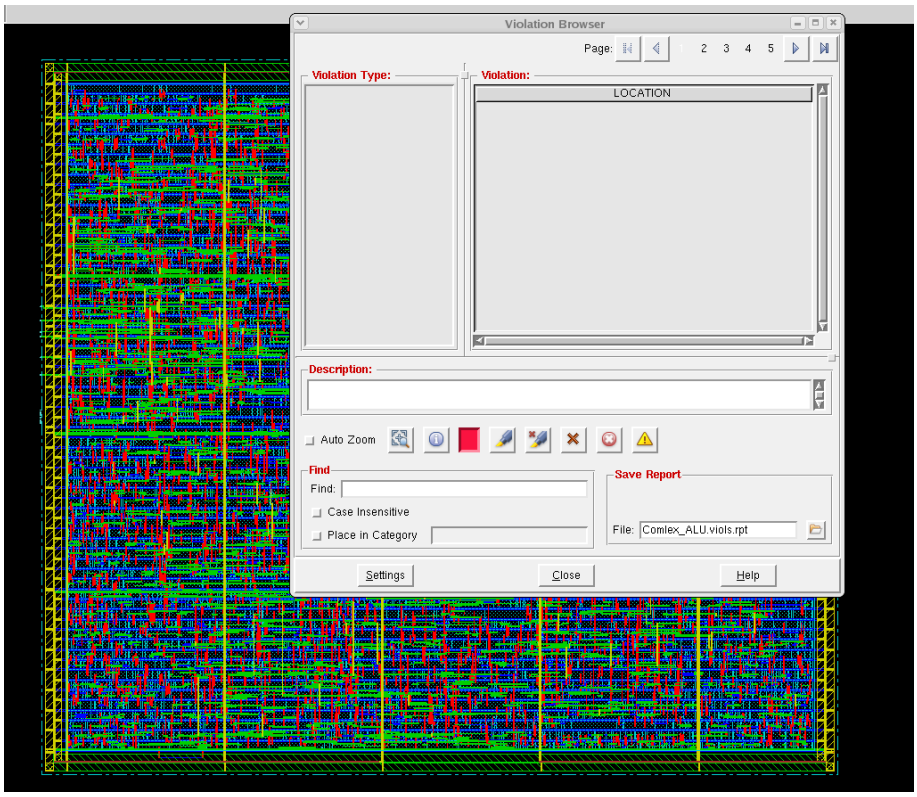

Figure 27: Steps Followed in Encounter

After this step, we proceed to **Add Filler**, which fills the gaps between cells to make the layout **manufacturable by the foundry**.

After performing this step, the gaps between the cells are filled as shown below.

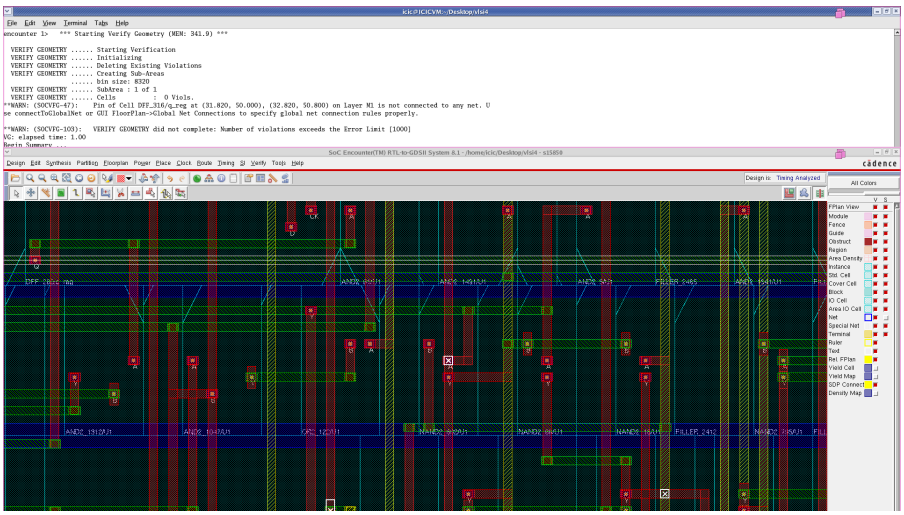

Figure 28: Steps Followed in Encounter

Now, we can perform the **metal fill** step to ensure **balanced metal distribution** across the circuit.

In this step, **all metal layers are selected**, and the empty spaces are filled with the corresponding **metal fill patterns**.

This results in the layout shown below.

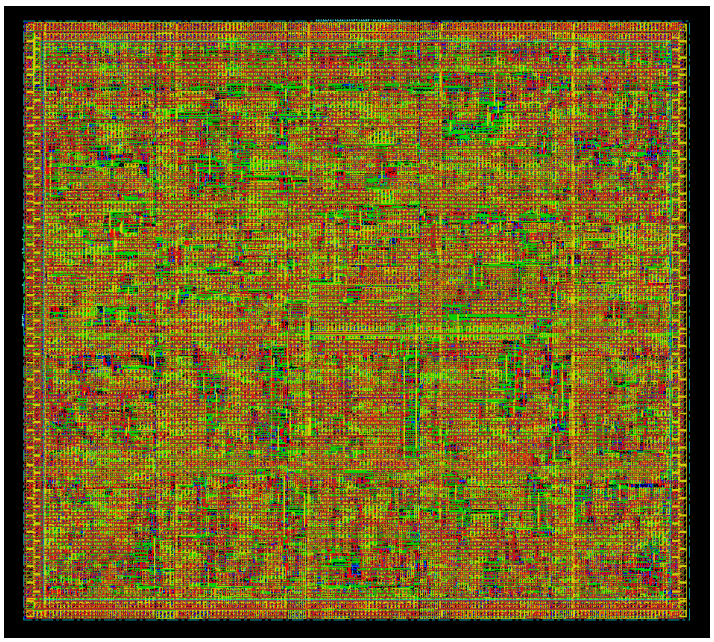

Figure 29: Steps Followed in Encounter

Next, we perform **geometry verification** again.

```
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 10 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 11 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 11 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 12 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 12 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 13 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 13 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 14 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 14 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 15 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 15 complete 0 Viols. 0 Wrngs.
 VERIFY GEOMETRY ..... SubArea : 16 of 16
 VERIFY GEOMETRY ..... Cells          :  0 Viols.
 VERIFY GEOMETRY ..... SameNet        :  0 Viols.
 VERIFY GEOMETRY ..... Wiring         :  0 Viols.
 VERIFY GEOMETRY ..... Antenna        :  0 Viols.
 VERIFY GEOMETRY ..... Sub-Area : 16 complete 0 Viols. 0 Wrngs.
'G: elapsed time: 2.00
egin Summary ...
 Cells       : 0
 SameNet     : 0
 Wiring      : 0
 Antenna     : 0
 Short       : 0
 Overlap     : 0
nd Summary

 Verification Complete : 0 Viols.  0 Wrngs.

*********End: VERIFY GEOMETRY**********
*** verify geometry (CPU: 0:00:02.2  MEM: 2.8M)
```

Figure 30: Steps Followed in Encounter

Another important output from the layout design stage is the **RC Extraction**, which produces a .spef file containing the **parasitic resistance and capacitance** data of the circuit. This file was extracted from the **Timing → Extract RC** menu and has been attached.

In addition, other files such as the **GDS** file and the **netlist** were also extracted and attached.

In the final step, we determine the **smallest possible die size** through a **trial-and-error** approach. Various die sizes were tested starting from the initial stage to find the most suitable size for this design.

The best estimated die size for this circuit was found to be approximately **386×386**. The process involved initially setting the size to **350**, then checking for any **violations**. Since violations were present, the size was gradually increased in steps (typically by doubling or halving) until a size was reached that produced **no violations**.

4- Post-Layout Simulation

To perform this task, the following files are required:

1. **Netlist Output from Encounter**: This file needs some modifications. Specifically, instead of the `Architecture` block, we replace it with the line `initial $nsda_module();`. Since the netlist is written in Verilog, we must remove everything except the input/output definitions. Given the netlist was about **15,000 lines**, we wrote a **Java program** to automate this cleanup. The Java code is included in the project's compressed archive.
2. **RC Extraction Output (.spef)**: This file is generated from Encounter and contains parasitic resistance and capacitance data.
3. **OSU Spice File**: Named `osu018_stdcells.sp`, this file is the required library to convert the netlist into a usable SPICE format.
4. **Transistor-Level Models for PMOS and NMOS**: For this, you can use the `tsmc018.m` file.
5. **Co-Simulation Support between VSIM and HSIM**: Enabled via the `libvpihsim.so` library found in: `/opt/synopsis/HSIM/hsimplus/platform/linux/bin/`
6. **cosim.cfg File**: This configuration file should contain the line: `Set_args top.sp`
7. **top.sp File**: The top-level SPICE file for the design.
8. **SPICE Model of the Main Module**: Required for transistor-level simulation.
9. **Testbench File**: To verify the functional behavior of the design.

Finally, place all these files in a single folder and use **ModelSim** to perform the simulation. The **post-layout simulation results** are shown in the figure below and are in full agreement with the previous results, indicating that the **layout stage was completed successfully**.

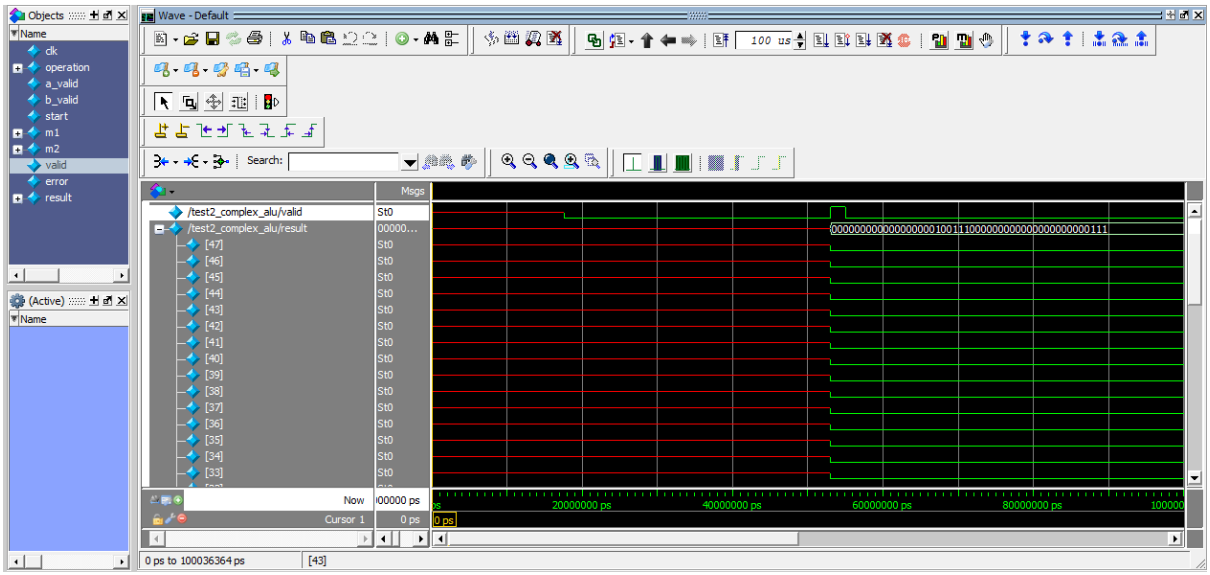

Figure 31: Steps Followed in Encounter