FINAL PROJECT

CE 6325.001 VLSI DESIGN

TEAM MEMBERS:

ALEXANDRA EDWINRAJ (NET ID: AXE210023)

HEENISHA REDDY BACHUGUDEM (NET ID: HXR210022)

KANUPRIYA SHARMA (NET ID :KXS22001

Introduction:

The arithmetic Logic unit (ALU) is the fundamental building block of many computing circuits, including the CPU of computers. The inputs of the ALU are the data to be operated on, called operands, and the code to specify which operations have to be performed, the output is the result of the process completed.

Our ALU design has two 16-bit inputs, a single 32-bit output, and a 4-bit select line to select the operations performed on the inputs. For our project, we designed ALU to perform various arithmetic and logic operations such as Addition, Subtraction, Multiplication, Division, Logical shift left, Logical shift right, Logical AND, Logical OR, Set if equal, etc. In total, we will be performing 16 operations. The Operation will be performed at the positive edge of the clock and active low reset to reset the ALU once the operation is completed the output is stored in the register variable and is passed to the D flip-flop to store the output and in the next cycle, the output is updated with the recent operation performed on the inputs.

Block diagram

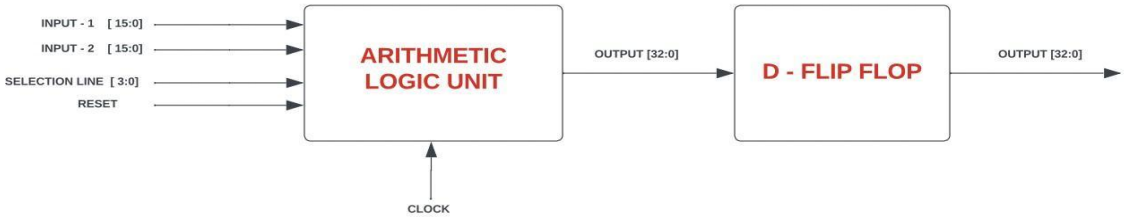


Figure 1 - Block diagram of the ALU with D-flip flop

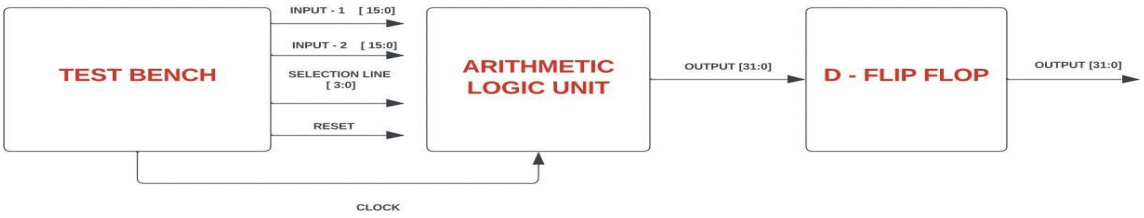


Figure 2 - Block diagram of the test bench connection with ALU

DESIGN PROCESS

PROJECT 1

In project 1, we chose ALU as our arbitrary design and designed a behavioural code and a test bench in Verilog. We simulated the design using ModelSim. As you can see in the above diagram, the 3-bit select line is responsible for selecting the operation that has to be performed on the operands, such as addition, subtraction, multiplication, division, left shift, right shift, rotate to the left, rotate to the right, and various other logical operations as mentioned below. Our design was a positive edge triggered and active low reset. When the output is generated it is stored inside the register variable and passed on to the d flip flop to store the output. In every cycle, as the inputs keep coming in, the outputs are generated and stored.

Behavioural Code Output:

S.NO	FUNCTION	INPUT - X	INPUT - Y	OUTPUT - Z
1.	Reset	0000	0000	00000000
2.	Addition	101000	010100	111100
3.	Subtraction	101000	010100	010100
4.	Multiplication	101000	010100	100000
5.	Division	101000	010100	000010
6.	Logical shift left	101000	010100	010000
7.	Logical shift right	101000	010100	010100
8.	Rotate left	101000	010100	010000
9.	Rotate right	101000	010100	010100
10.	AND	101000	010100	000000
11.	OR	101000	010100	111100
12.	NOR	101000	010100	000011
13.	NAND	101000	010100	11111
14.	XOR	101000	010100	111100
15.	XNOR	101000	010100	000011
16.	Less than	28	14	0000

PROJECT 2

We used Synopsys Design Vision to generate a mapped netlist based on our library cells. During this Project we got a better idea of the complexity of our design as well as an exact cell count. We have found the cell count for the ALU Module using Synopsys Design Vision (CAD Tool) as 3536. The Design Vision software read our Behavioral Verilog code and then used the library to generate a mapped netlist file. This mapped the netlist file along with the concatenated header.v code was simulated again in ModelSim using the same testbench code. The waveform obtained for the mapped netlist code is compared with the waveforms from the behavioral code. Using the .spfiles generated from Virtuoso, we generated the .libfile using Siliconsmart. We converted the .libfile into .dbfile to extract the synthesized netlist by characterizing the Verilog code using the standard cell library created.

PROJECT 3

While working on the 3rd project, we got to learn how to use the Cadence Virtuoso tool to generate a schematic and layout of the Inverter. Our design objective was to minimize the inverter layout's bounding box area (H*W) while minimizing the energy-delay product (EDP). We also made sure that the pin pitch and offset standards were met. We then ran the DRC and LVs checks. Initially, we came across some errors but we learned to rectify them. We also ran PEX and generated the netlist. Using the netlist, we simulated the THL and TLH waveforms using Waveview.

PROJECT 4

While working on project 4, we developed a keen understanding on schematic to layout conversion for any kind of cell. With the understanding of the Virtuoso tools, we created a standard library of cells of INV, NAND2, NOR2, XOR2, MUX2:1, AOI211, OAI21, AOI22. We repeated all the steps of project 3 again for all the cells, including abstract view. They passed all the DRC and LVS checks successfully. We also made sure that all the cells were of the same height i.e, 7.535um to make them align in a straight row.

PROJECT 5

We drew a layout for the D-Flip Flop schematic given by the professor, and implemented it on Cadence Virtuoso. We then repeated the same steps for D-Flip Flop.

PROJECT 6

During this Final Design, we took the Verilog code which we used during project 2 and laid out the cells that we generated in our cell library. We generated an abstract view of all the cells including filler and characterized them. We then combined all the .lib files that were generated in the process. We finally used the .lib file and the netlist files generated in the second project for automatic placement and route on Innovus. We converted this file to Cadence and ran the LVS/DRC checks.

TRADE-OFFS

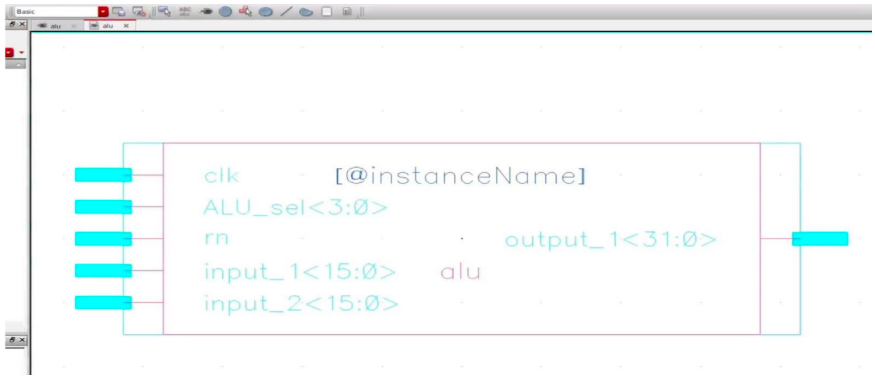
- Our design consumed a power of 1.041 e-05W for 10us. To reduce the power, we increased the period to 11us and it consumed a power of 7.690 e-06 W. Power consumption was reduced by 30%.
- We increased the clock period to which we were getting maximum slack so that the power of the cell is reduced. For each 1us of clock period increment we reduce the power consumption by 30%.
- Our design is efficient for high clock periods but the power is the limiting factor. Therefore, our design is efficient for a higher clock period.
- Height of D-Flip Flop was decreased in order to match with all the other cell heights. This ensures that we have the standard library of cells.

TESTING PROCESS

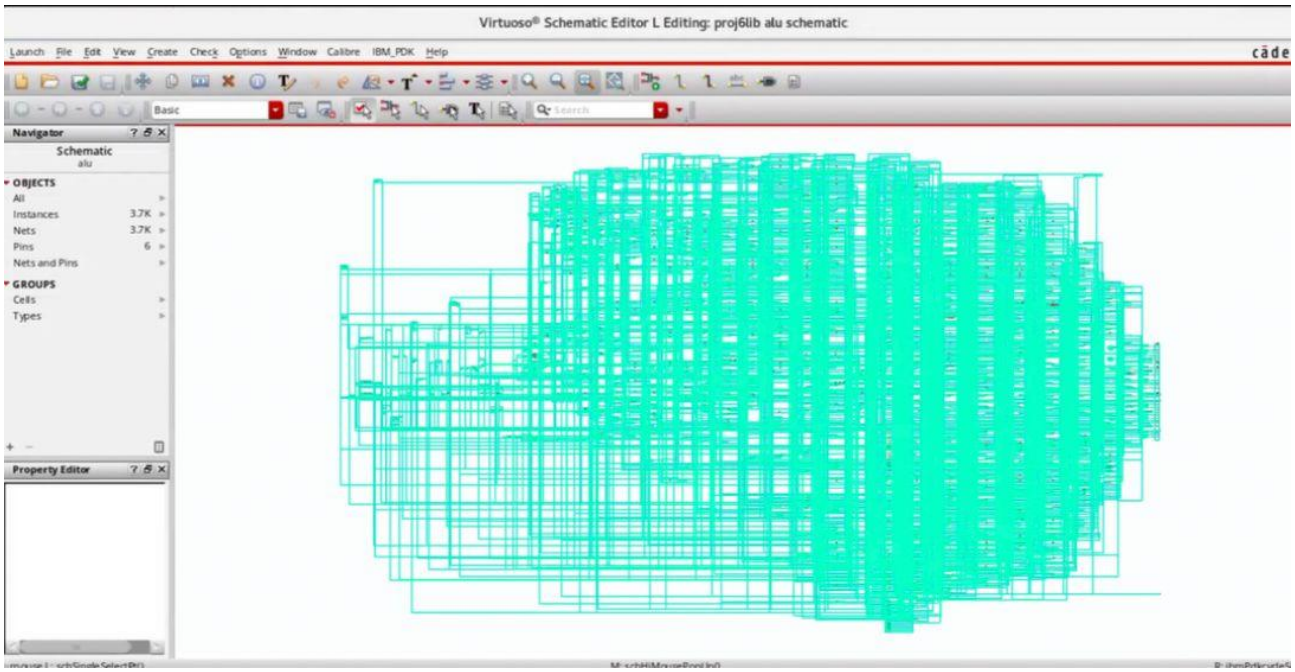
- Throughout the flow of the design, testing was a constant step to be taken care of before proceeding to the next step.
- Initially, we tested the verilog code by simulating the waveforms on ModelSim and approving it only when we knew that the ALU was performing a function we intended to perform.
- In Cadence, we ran tests like LVS AND DRC to make sure that it's error free before generating its PEX files.
- We used DRC to test whether the design is following the design rules
- We ran the LVS to make sure our layout matched the schematic functionality.
- We extracted the netlist and tested the cell post layout using HSPICE.
- Eight out of eight cells were verified before proceeding to final layout.
- We tested the verilog code on our cells to make sure it was working and still had the same logical function.
- DRC and LVS was performed again to make sure that the layout matched the verilog schematic.
- We ran the final test with the extracted layout using the same tests we used to test the original code.
- Once the post-layout simulation matched the original test bench, we were certain our design was functioning as designed.

FINAL SCHEMATIC :

SYMBOLS :

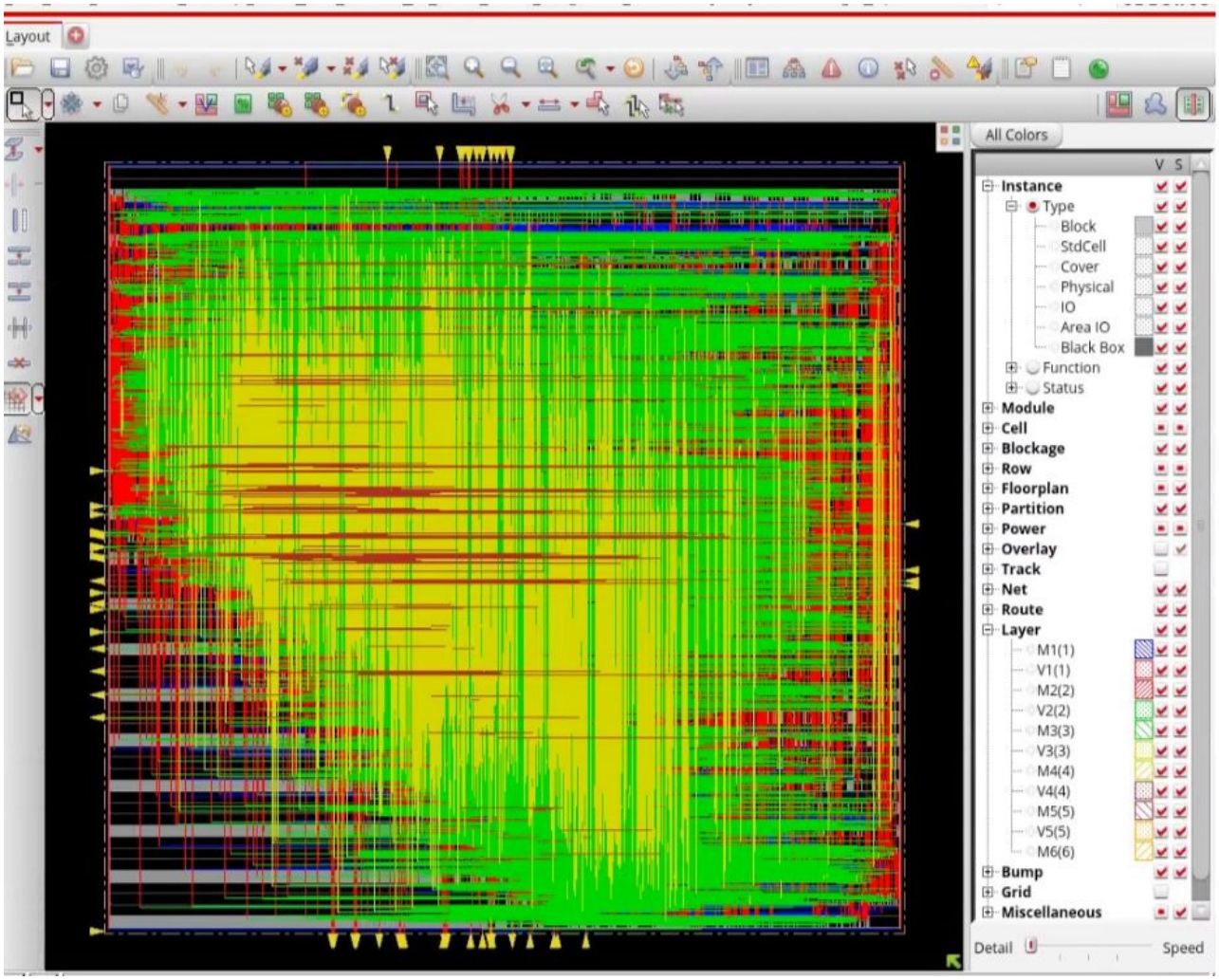


FINAL SCHEMATIC:



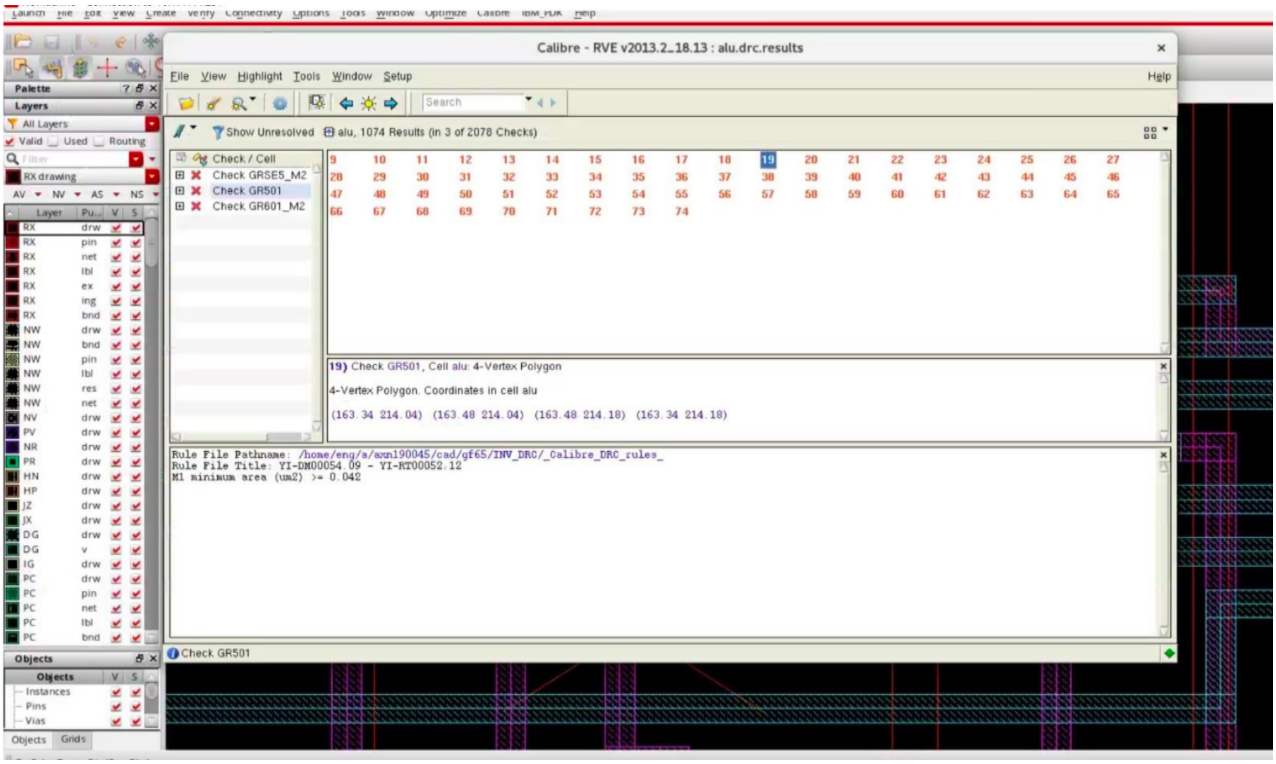
LAYOUT (Virtuoso)

LAYOUT Innovus :

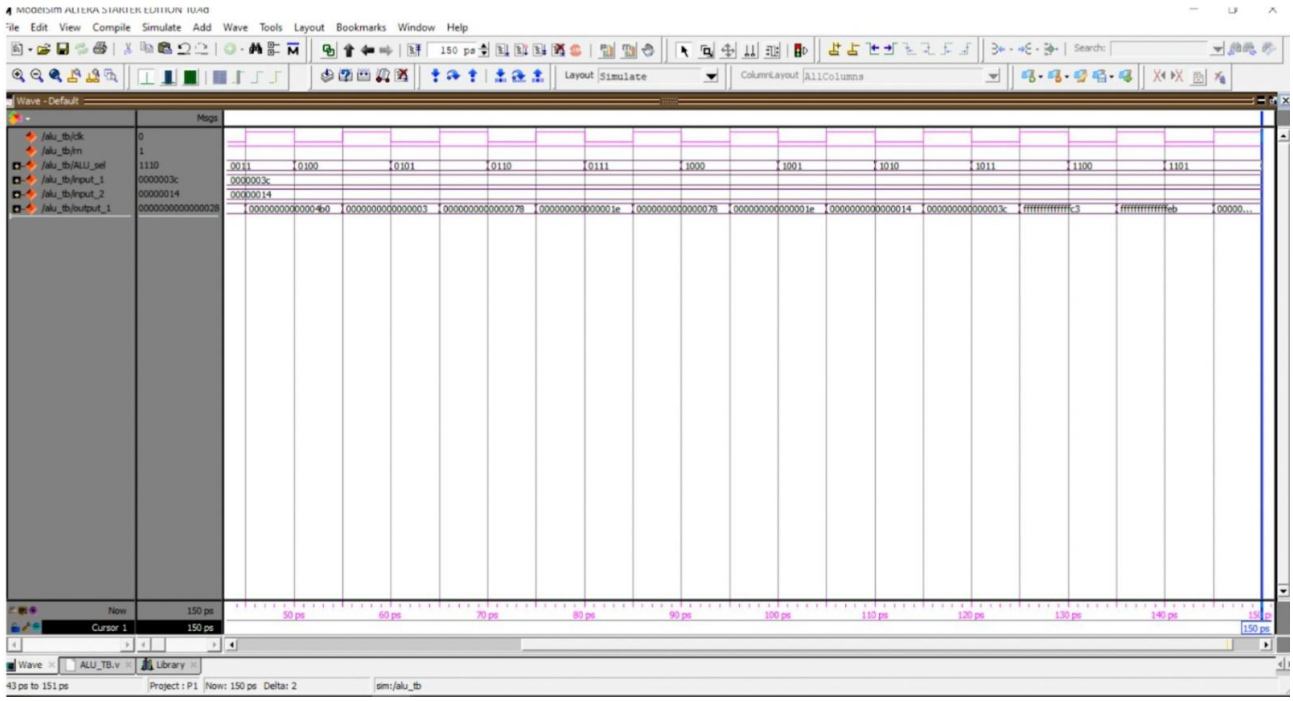


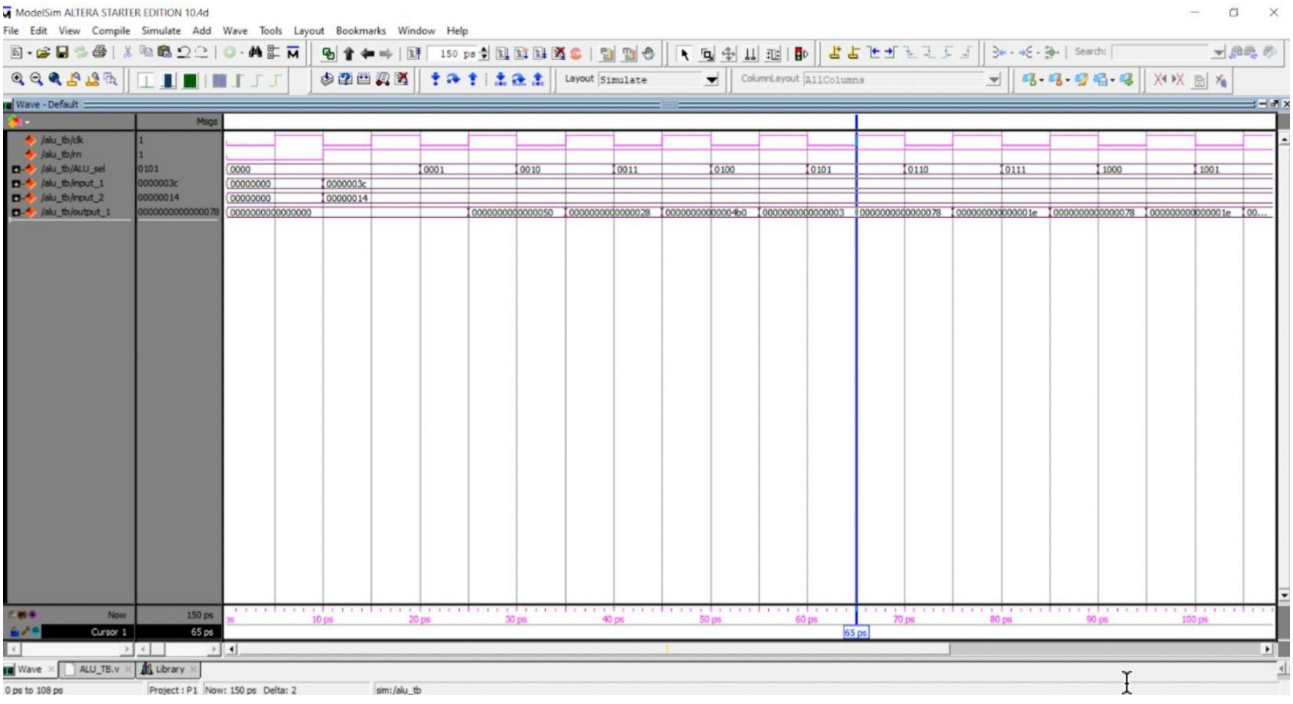
DRC results :

Initially we had 12000 errors when we ran the DRC check.
Were able to rectify majority of them and reached around 1000 errors.

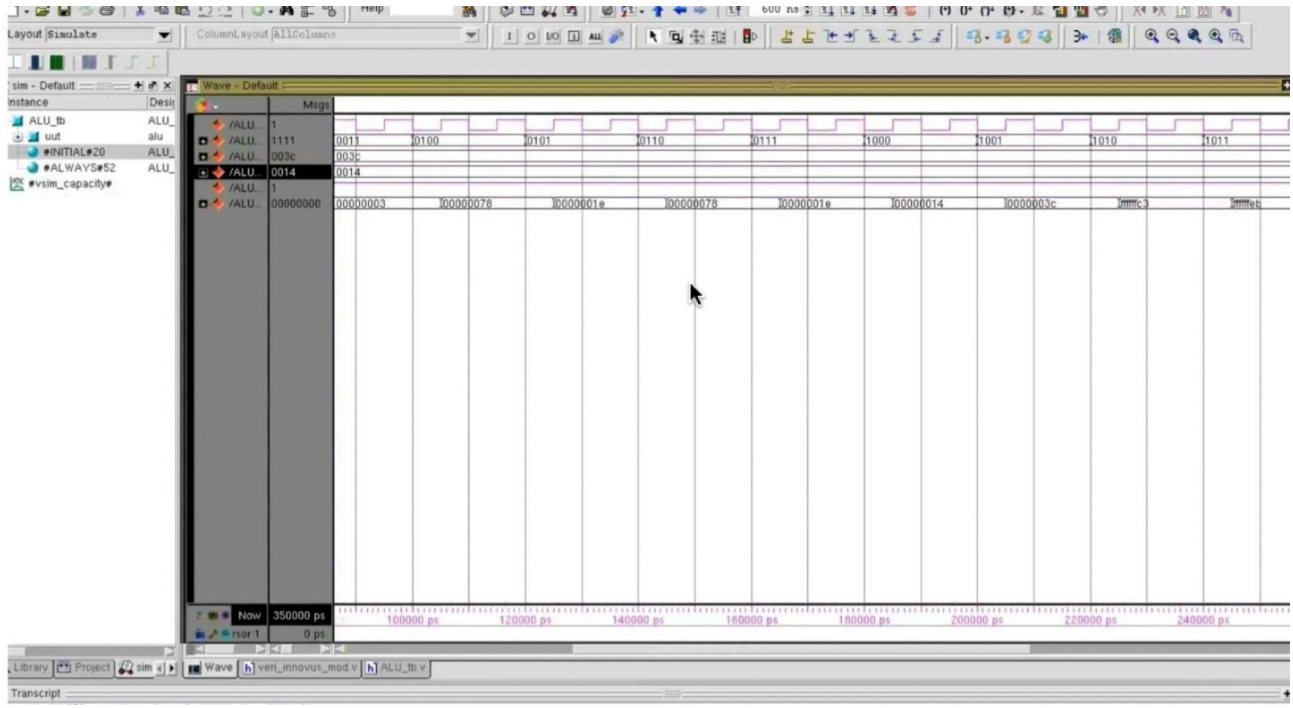


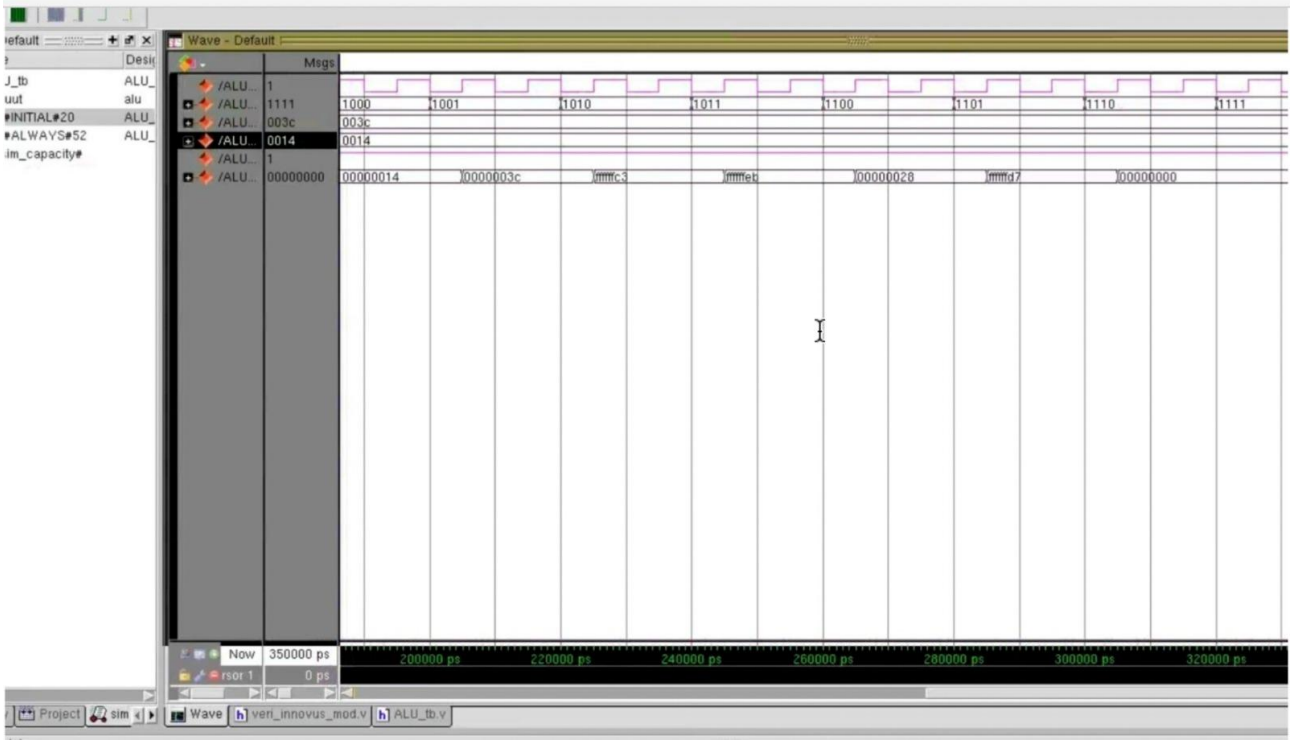
Original Verilog Code Simulation:





Output Waveform of the actual waveform:





Prime Time Analysis :

```
Report : Averaged Power
Design : alu
Version: M-2016.12-SP3-1
Date   : Fri Dec  2 16:02:17 2022
****************************************


  Attributes
  ----------
      i  -  Including register clock pin internal power
      u  -  User defined power group

                          Internal  Switching  Leakage    Total
Power Group               Power     Power      Power      Power   (     %)  Attrs
--------------------------------------------------------------------------------
clock_network             3.016e-10 9.861e-09 3.999e-11 1.020e-08 ( 0.10%)  i
register                  4.975e-08 3.626e-10    0.0000 5.011e-08 ( 0.48%)
combinational             2.438e-07 9.875e-06 2.269e-07 1.035e-05 (99.42%)
sequential                   0.0000    0.0000    0.0000    0.0000 ( 0.00%)
memory                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
io_pad                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box                    0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power  = 9.885e-06   (95.00%)
  Cell Internal Power  = 2.938e-07   ( 2.82%)
  Cell Leakage Power   = 2.270e-07   ( 2.18%)
                         ---------
Total Power            = 1.041e-05  (100.00%)

1
exit
```

```
report_power
****************************************
Report : Averaged Power
Design : alu
Version: M-2016.12-SP3-1
Date   : Fri Dec  2 16:02:17 2022
****************************************


  Attributes
  ----------
      i  -  Including register clock pin internal power
      u  -  User defined power group

                          Internal  Switching  Leakage    Total
Power Group               Power     Power      Power      Power   (     %)  Attrs
--------------------------------------------------------------------------------
clock_network             1.971e-10 7.231e-09 3.999e-11 7.469e-09 ( 0.10%)  i
register                  3.648e-08 2.659e-10    0.0000 3.675e-08 ( 0.48%)
combinational             1.771e-07 7.242e-06 2.269e-07 7.646e-06 (99.43%)
sequential                   0.0000    0.0000    0.0000    0.0000 ( 0.00%)
memory                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
io_pad                       0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box                    0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power  = 7.249e-06   (94.27%)
  Cell Internal Power  = 2.137e-07   ( 2.78%)
  Cell Leakage Power   = 2.270e-07   ( 2.95%)
                         ---------
Total Power            = 7.690e-06  (100.00%)

1
exit
```