

EECS 4612: Final Project

Harman Goraya

218692608

1. Testing of units and 1-bit ALU and Synthesis Simulation Results

1.1: Arithmetic Unit

The arithmetic unit uses a full adder to compute the sum Di and carry-out Couti from three inputs: Ai, a modified version of Bi, and Cini. A 4-to-1 multiplexer selects the second operand—0, Bi, ~Bi, or 1—based on control signals S1 and S0, enabling operations such as transfer, increment, addition, subtraction, and decrement.

The truth table defines how each combination of S1, S0, and Cini maps to a specific operation (e.g., A, A+1, A+B, A+B+1, A+~B, A+~B+1, A-1). The selected operand (Bi_after) feeds the full adder, where the sum is computed using XOR logic among Ai, Bi_after, and Cini, and the carry-out is generated using the standard full-adder majority function.

Cini	S1	S0	Operation	Function
0	0	0	F=A	Throughput of A
1	0	0	F=A+1	Increment A
0	0	1	F=A+B	Input Addition
1	0	1	F=A+B+1	Input Addition with Carry
0	1	0	F=A+~B	Input Subtraction with Borrow
1	1	0	F=A+~B+1	Input Subtraction
0	1	1	F=A-1	Decrement A
1	1	1	F=A	Throughput of A (addition of 2 1's plus a 1 from Cin are equal to a throughput of either input)

Table 1: Signal Combinations and Arithmetic Operations

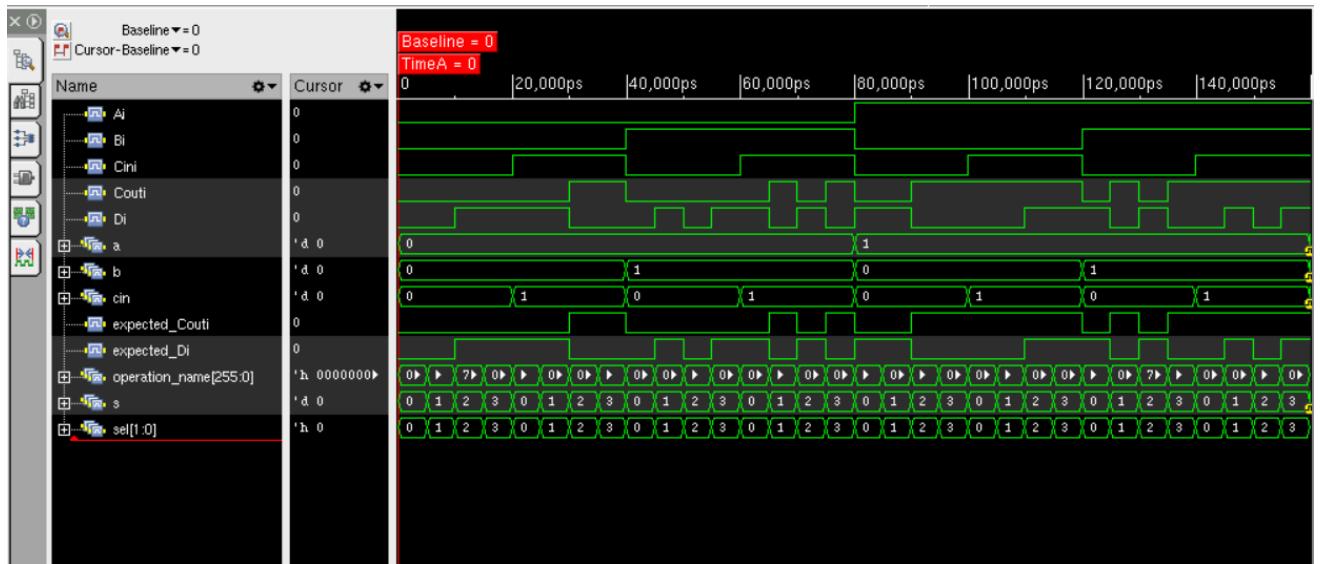


Figure 1: Waveform of the Arithmetic Unit

As you can see from the waveform above, we can see the operations being performed as well as the inputs and outputs of all values. With our waveforms, they are matching the outputs. The test goes through all operations in table 1 to ensure the code works great. When we run the testbench as well, we can see our results which match the test expected results.

1.2: Full Adder

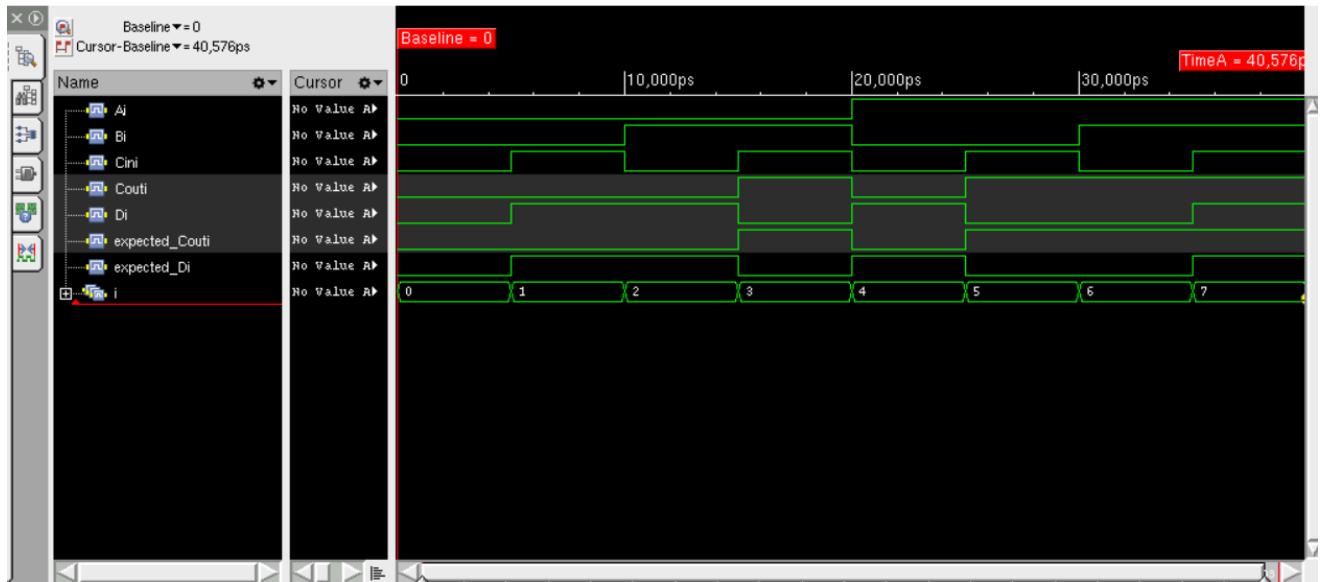


Figure 2: Full Adder Waveform

The full adder should function as the following table.

A	B	Cin	D	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 2: Full Adder Truth Table

In the Full Adder waveform, we can see the design go through each signal. We can see that the outputs match our expected values from the table. The goal with the full adder is to produce 1 output with a carry out from 2 inputs and a carry in.

1.3: Logic Unit

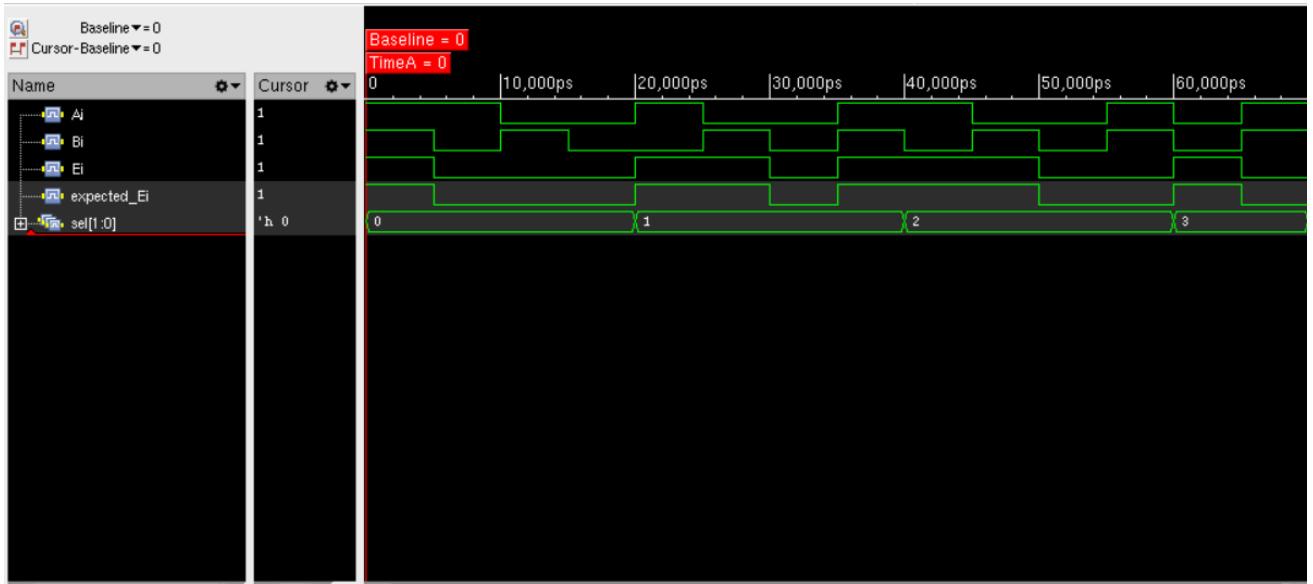


Figure 3: Logic Unit Waveform

The logic unit's main function is to perform logical operations. The `sel[1:0]` controls the logic operation that should be used. `Ai` and `Bi` are the inputs while the `Ei` is the output. Within the logic unit, the multiplexer is also used to calculate what logic operation should go through the system as `Ei`.

1.4: Mux 4to1

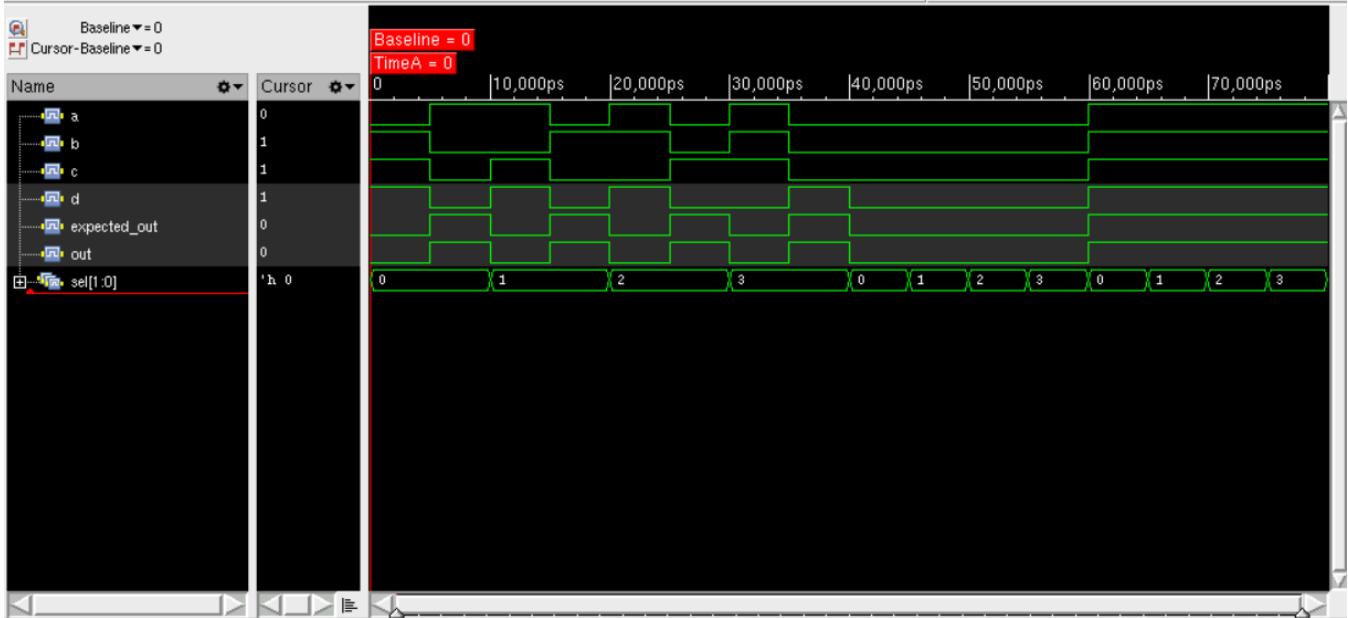


Figure 4: Mux Waveform

Below, we can see the table for a 4 to 1 multiplexer. From our waveform, we can see that it matches the truth table. Our tests went over the key inputs, such as the basic routes, having all inputs zero, having all inputs 1, and an inverted selector edge case. This is tied in with the logic unit, where this selects the operation that would be performed. It gives us 4 different options for the operation.

S1	S0	Input X	Input Y
0	0	A	A
0	1	B	B
1	0	C	C
1	1	D	D

Table 3: 4to1 Multiplexer Truth Table

1.5: 1-bit ALU

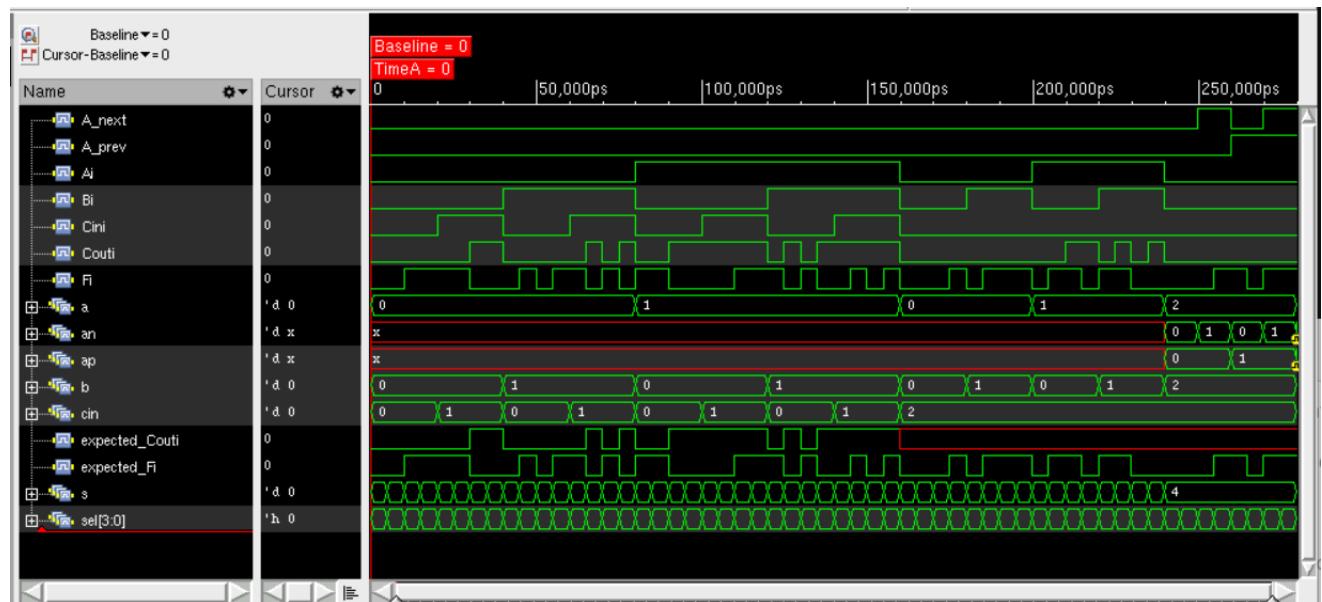


Figure 5: 1bit ALU

The below table shows all the different operations that can be performed on the ALU. There were a variety of verification tests that were run on the design to ensure that everything goes smoothly and we have 0 errors. In figure 5, we can see the simulation results match the truth table.

Sel[3]	Sel[2]	Sel[1]	Sel[0]	Cin	Operation
0	0	0	0	0	$F = A$
0	0	0	0	1	$F = A + 1$
0	0	0	1	0	$F = A + B$
0	0	0	1	1	$F = A + B + 1$
0	0	1	0	0	$F = A + B'$
0	0	1	0	1	$F = A + B' + 1$

0	0	1	1	0	$F = A - 1$
0	0	1	1	1	$F = A$
0	1	0	0	X	$F = A \text{ and } B$
0	1	0	1	X	$F = A \text{ or } B$
0	1	1	0	X	$F = A \text{ XOR } B$
0	1	1	1	X	$F = \sim A$
1	0	X	X	X	$F = A(i-1)$
1	1	X	X	X	$F = A(i+1)$

Table 4: ALU Operations

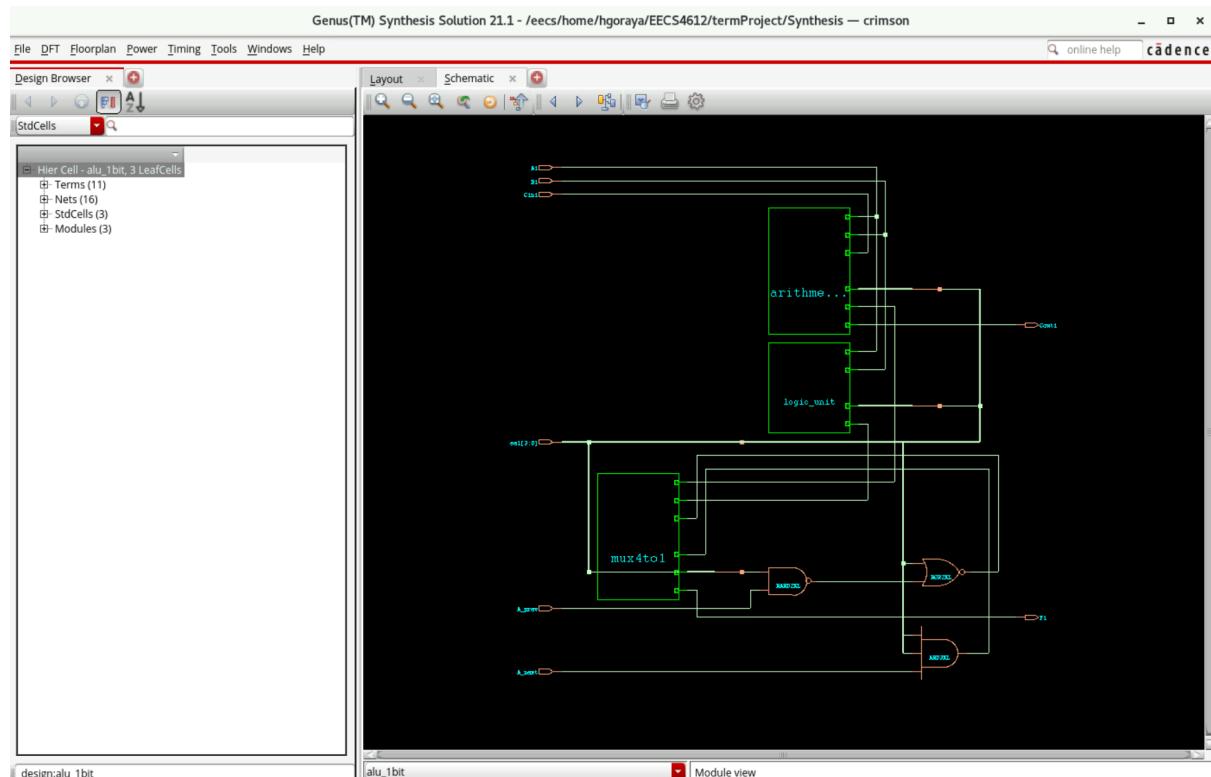


Figure 6: Schematic of the Synthesis

In figure 6, we can see the synthesis of the 1 bit ALU. We can see the mux, arithmetic and logic unit all tied together, with 6 inputs and 2 outputs.

Area report:

```
=====
Generated by:      Genus(TM) Synthesis Solution 21.17-s066_1
Generated on:      Dec 11 2025 01:09:04 pm
Module:           alu_1bit
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode:    enclosed
Area mode:        timing library
```

```
=====
```

Gate	Instances	Area	Library
------	-----------	------	---------

AND3XL	1	2.052	slow_vdd1v0
NAND2XL	1	1.026	slow_vdd1v0
NOR2XL	1	1.026	slow_vdd1v0
total	3	4.104	

Type	Instances	Area	Area %
------	-----------	------	--------

unresolved	3	0.000	0.0
logic	3	4.104	100.0
physical_cells	0	0.000	0.0
total	6	4.104	100.0

Power Report:

Instance: /alu_1bit

Power Unit: W

PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
----------	---------	----------	-----------	-------	------

memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	4.99960e-11	9.56083e-09	6.19647e-09	1.58073e-08	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	4.99960e-11	9.56083e-09	6.19647e-09	1.58073e-08	100.00%
Percentage	0.32%	60.48%	39.20%	100.00%	100.00%

Layout:

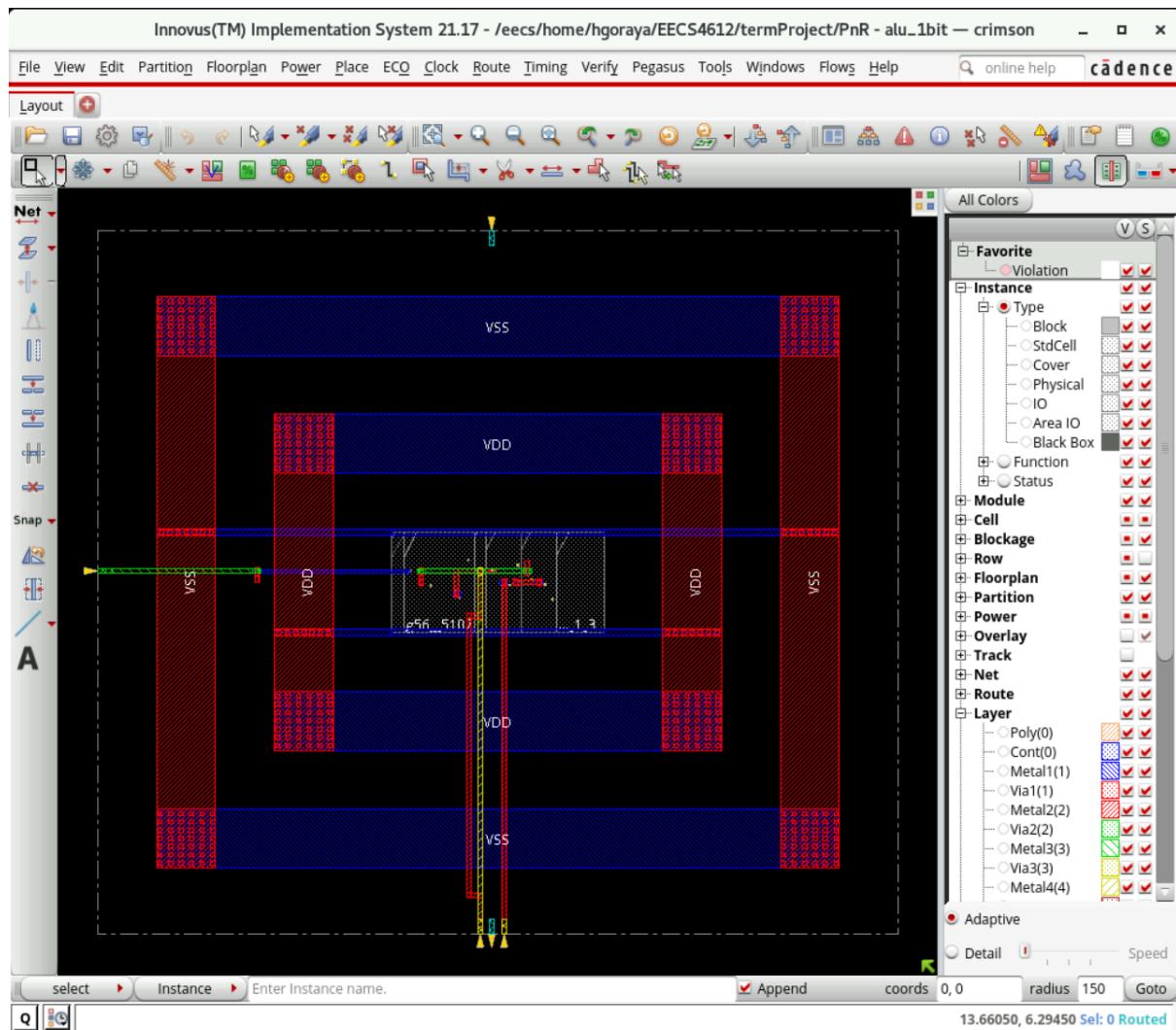


Figure 7: Design of 1 bit ALU

In figure 7, we can see the final design of the 1 bit ALU. This includes all the inputs, outputs as well as the current supply.

DRC Verification Check Results:

*** Starting Verify DRC (MEM: 1970.5) ***

```

VERIFY DRC ..... Starting Verification
VERIFY DRC ..... Initializing
VERIFY DRC ..... Deleting Existing Violations
VERIFY DRC ..... Creating Sub-Areas
VERIFY DRC ..... Using new threading
VERIFY DRC ..... Sub-Area: {0.000 0.000 13.600 11.970} 1 of 1
VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.

```

Verification Complete : 0 Viols.

*** End Verify DRC (CPU: 0:00:00.0 ELAPSED TIME: 0.00 MEM: 256.1M) ***

Verification Results:

***** Start: VERIFY CONNECTIVITY *****

Start Time: Thu Dec 11 13:32:41 2025

Design Name: alu_1bit

Database Units: 2000

Design Boundary: (0.0000, 0.0000) (13.6000, 11.9700)

Error Limit = 1000; Warning Limit = 50

Check all nets

Begin Summary

Found no problems or warnings.

End Summary

End Time: Thu Dec 11 13:32:41 2025

Time Elapsed: 0:00:00.0

***** End: VERIFY CONNECTIVITY *****

Verification Complete : 0 Viols. 0 Wrngs.

(CPU Time: 0:00:00.0 MEM: 0.000M)

Timing Report:

```
S#####
#####
# Design Stage: PostRoute
# Design Name: alu_1bit
# Design Mode: 90nm
# Analysis Mode: MMMC OCV
# Parasitics Mode: SPEF/RCDB
# Signoff Settings: SI On
#####
#####
AAE_INFO: 1 threads acquired from CTE.
Start delay calculation (fullDC) (1 T). (MEM=2264.55)
Initializing multi-corner resistance tables ...
Total number of fetched objects 7
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
AAE_INFO-618: Total number of nets in the design is 18, 61.1 percent of the nets selected for SI analysis
End delay calculation. (MEM=2280.78 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=2280.78 CPU=0:00:00.1 REAL=0:00:00.0)
Loading CTE timing window with TwFlowType 0...(CPU = 0:00:00.0, REAL = 0:00:00.0, MEM = 2272.8M)
Add other clocks and setupCteToAAEClockMapping during iter 1
```

Loading CTE timing window is completed (CPU = 0:00:00.0, REAL = 0:00:00.0, MEM = 2272.8M)
Starting SI iteration 2
Start delay calculation (fullDC) (1 T). (MEM=2025.69)
Glitch Analysis: View worst_case -- Total Number of Nets Skipped = 0.
Glitch Analysis: View worst_case -- Total Number of Nets Analyzed = 7.
Total number of fetched objects 7
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
AAE_INFO-618: Total number of nets in the design is 18, 0.0 percent of the nets selected for SI analysis
End delay calculation. (MEM=2065.86 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=2065.86 CPU=0:00:00.0 REAL=0:00:00.0)
*** Done Building Timing Graph (cpu=0:00:00.3 real=0:00:00.0 totSessionCpu=0:01:05 mem=2065.9M)

timeDesign Summary

Setup views included:

worst_case

+-----+-----+-----+			
Setup mode all default			
+-----+-----+-----+			
WNS (ns): 0.000 0.000			
TNS (ns): 0.000 0.000			
Violating Paths: 0 0			
All Paths: 0 0			
+-----+-----+-----+			

+-----+-----+-----+				
	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: 66.667%
(100.000% with Fillers)

Reported timing to dir timingReports

Total CPU time: 0.7 sec

Total Real time: 2.0 sec

Total Memory Usage: 2044.183594 Mbytes

Reset AAE Options

*** timeDesign #1 [finish] : cpu/real = 0:00:00.7/0:00:02.1 (0.3), totSession cpu/real =

0:01:04.9/0:23:33.4 (0.0), mem = 2044.2M

0

2. 32-bit ALU

2.1: Modular Version

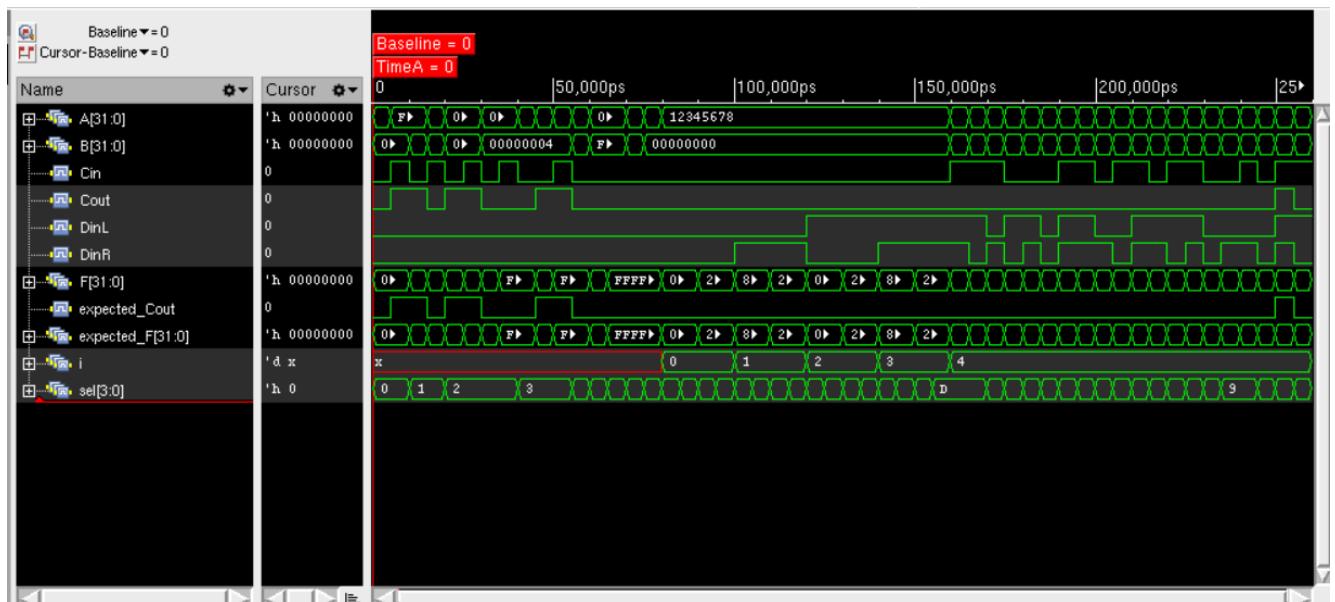


Figure 8: Modular Version waveform results

In the above waveform, we can see the test results of the modular 32 bit ALU. Inputs include A and B (32 bits each), Cin, S0 through S4, and DinL and DinR. Our outputs are F (32 bit output) and Carry Out. For the testbench, I included a variety of tests to ensure the smooth operation of the ALU. Some key tests include transfer A, increment A, addition of A and B with and without A, subtraction of A and B both with and without borrow, Decrement A, all the logic operations and all the shift operations. This ensured our modular version passed all tests and the whole program worked as it should.

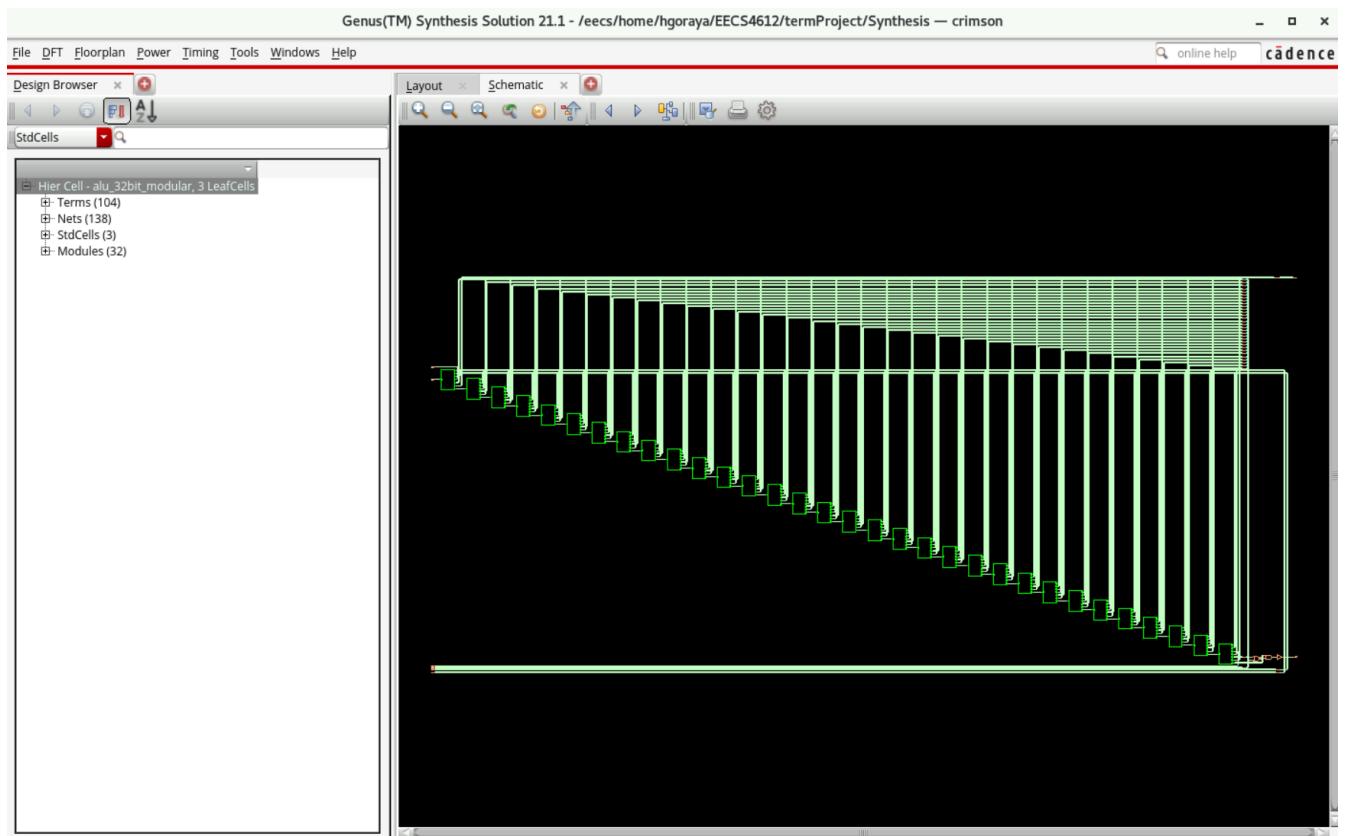


Figure 9: Synthesis of the Modular 32 bit ALU

Above in figure 9, we can see the cascade of the 1 bit ALU to make the 32 bit ALU.

Area report:

This is only the relevant data. Full data is included in the zip file.

Instances	Area
total	485 1451.106

Power Report:

Instance: /alu_32bit_modular

Power Unit: W

PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	4.96023e-08	4.39078e-07	2.51057e-05	2.55943e-05	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%

pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	4.96023e-08	4.39078e-07	2.51057e-05	2.55943e-05	100.00%
Percentage	0.19%	1.72%	98.09%	100.00%	100.00%

From these results, we can see that the area shows 15, but seems to be incorrect calculations. It seems to not include the fact that there are all the 1 bit ALUs. Doing some calculations, the real area consumption would be somewhere between 1400-1500.

Power seems to be very low as well as we should see approximately 2.6e-06 W.

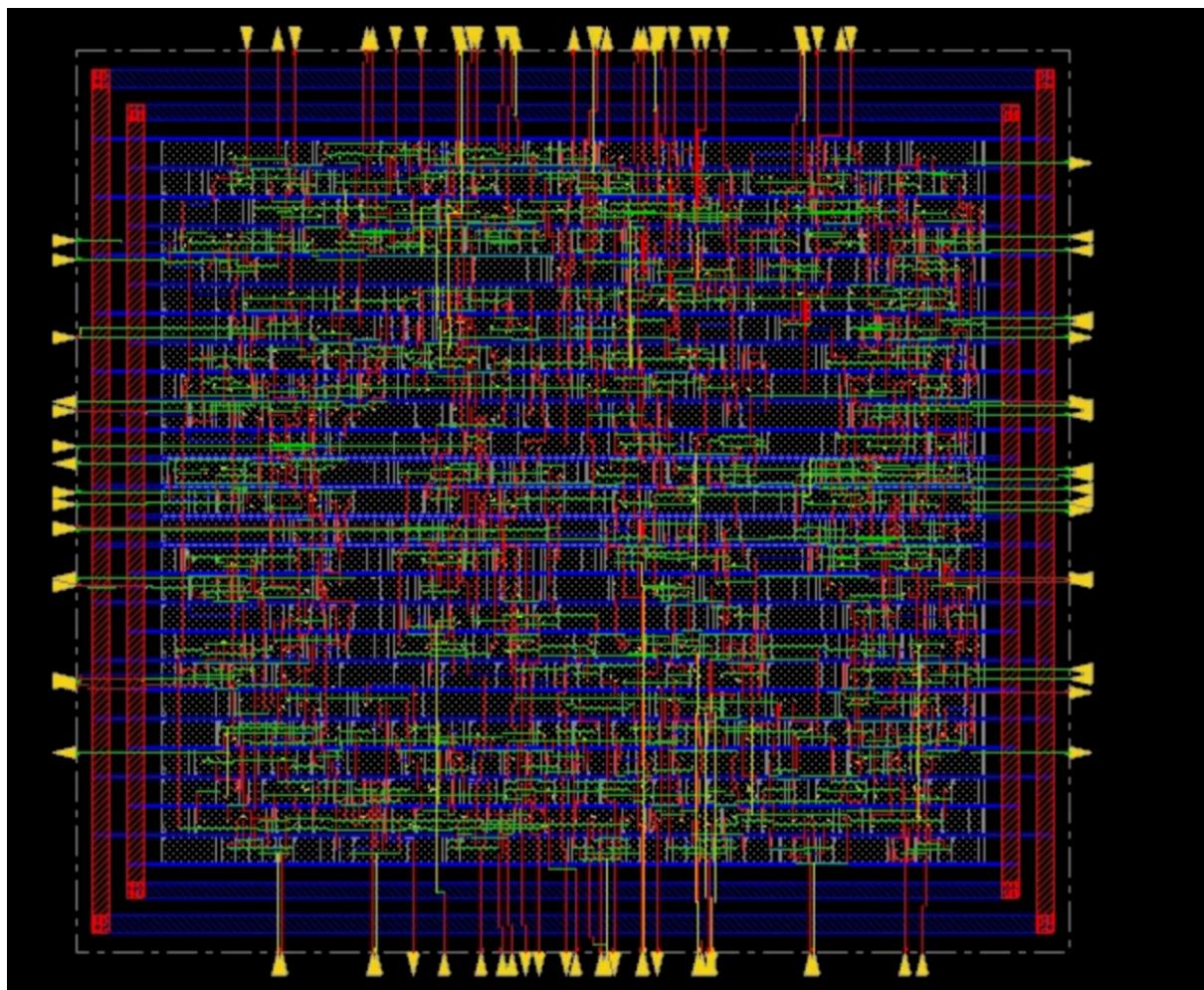


Figure 10: Modular Version ALU

DRC Verification:

*** Starting Verify DRC (MEM: 1975.1) ***

```

VERIFY DRC ..... Starting Verification
VERIFY DRC ..... Initializing
VERIFY DRC ..... Deleting Existing Violations
VERIFY DRC ..... Creating Sub-Areas

```

VERIFY DRC Using new threading
VERIFY DRC Sub-Area: {0.000 0.000 16.400 13.680} 1 of 1
VERIFY DRC Sub-Area : 1 complete 0 Viols.

Verification Complete : 0 Viols.

*** End Verify DRC (CPU: 0:00:00.0 ELAPSED TIME: 0.00 MEM: 256.1M) ***

Connectivity Verification:

***** Start: VERIFY CONNECTIVITY *****

Start Time: Fri Dec 12 13:36:44 2025

Design Name: alu_32bit_modular
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (16.4000, 13.6800)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary

Found no problems or warnings.

End Summary

End Time: Fri Dec 12 13:36:44 2025
Time Elapsed: 0:00:00.0

***** End: VERIFY CONNECTIVITY *****

Verification Complete : 0 Viols. 0 Wrngs.

(CPU Time: 0:00:00.0 MEM: 0.000M)

2.2: Behavioral Version

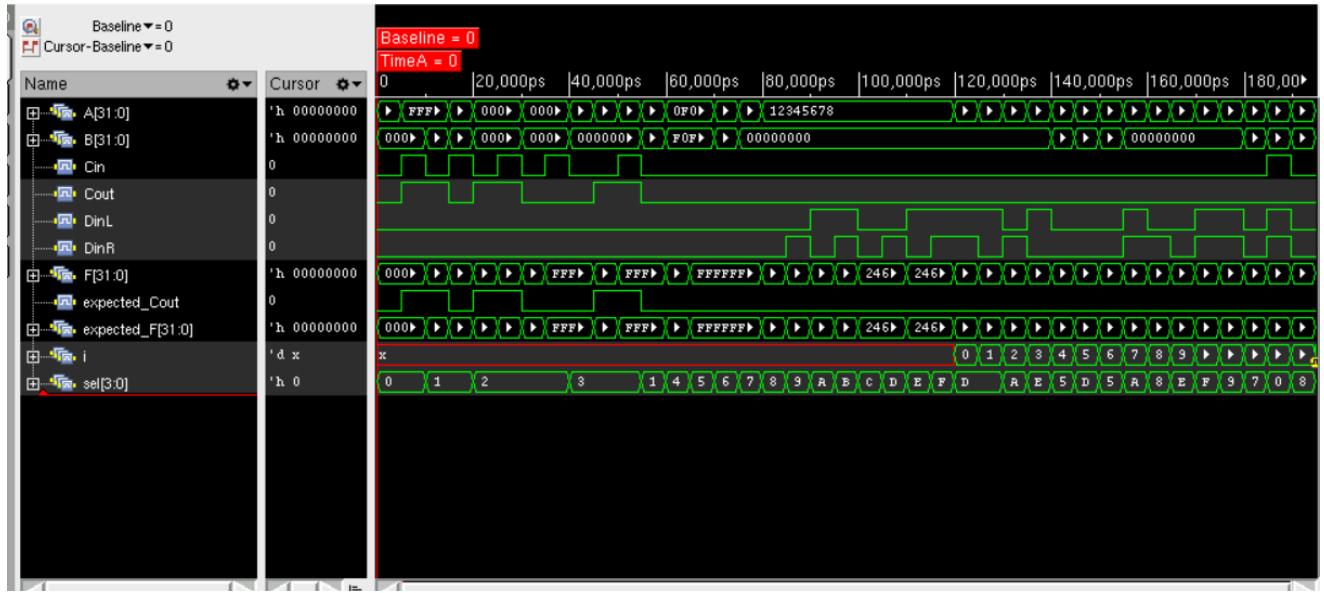


Figure 11: Testbench for figure 11

For the behavioral version, similar tests to the modular version were used to ensure the smooth operation of the 32 bit ALU.

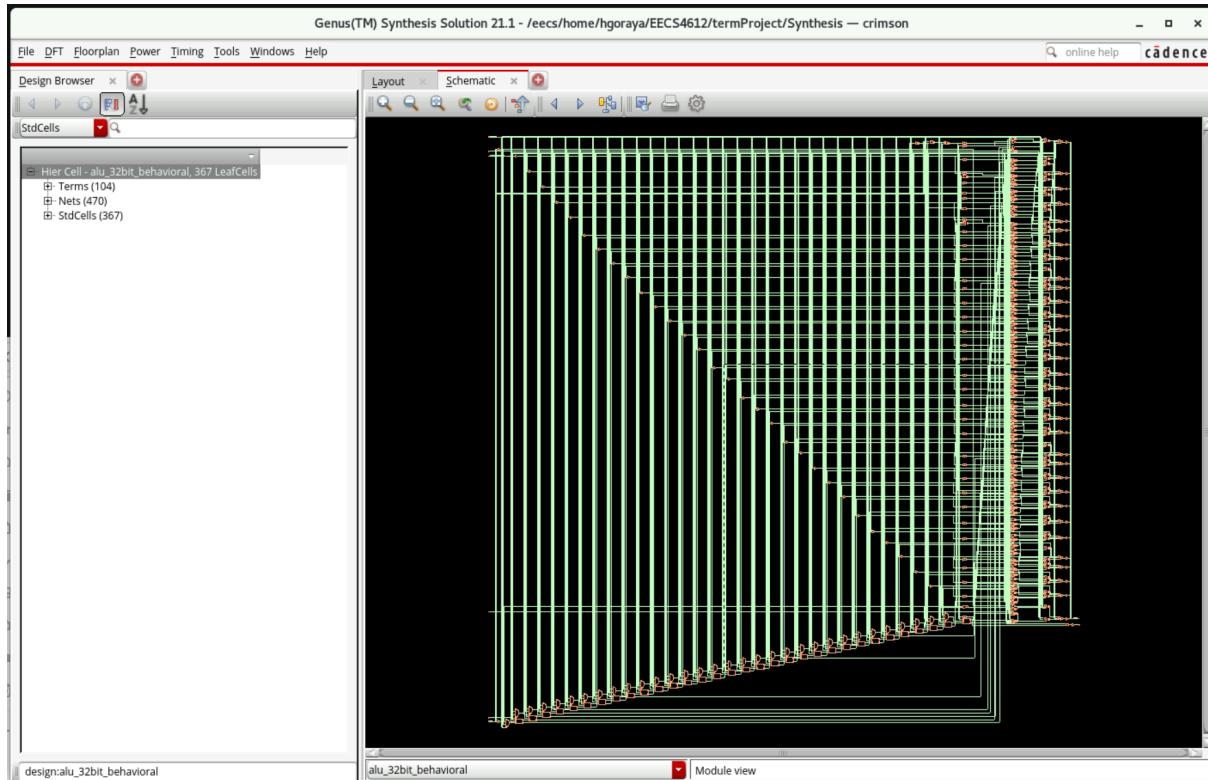


Figure 12: Import for Synthesis

Area Report:

The following report is cut just for relevant information. Full reports are available in the zip file.

Instances	Area

total 367 975.726

We can see that the area is lower than the modular version. It has at least a 30% overall improvement in area.

Power report:

Category	Leakage	Internal	Switching	Total
Subtotal	3.61007e-08	3.28942e-07	2.49149e-05	2.52799e-05
Percentage	0.14%	1.30%	98.56%	100.00%

DRC Check:

*** Starting Verify DRC (MEM: 2678.7) ***

```
VERIFY DRC ..... Starting Verification
VERIFY DRC ..... Initializing
VERIFY DRC ..... Deleting Existing Violations
VERIFY DRC ..... Creating Sub-Areas
VERIFY DRC ..... Using new threading
VERIFY DRC ..... Sub-Area: {0.000 0.000 50.800 44.460} 1 of 1
VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.
```

Verification Complete : 0 Viols.

*** End Verify DRC (CPU: 0:00:00.0 ELAPSED TIME: 0.00 MEM: 256.1M) ***

Connectivity Verification:

***** Start: VERIFY CONNECTIVITY *****

Start Time: Fri Dec 12 14:08:56 2025

Design Name: alu_32bit_behavioral
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (50.8000, 44.4600)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary

Found no problems or warnings.

End Summary

End Time: Fri Dec 12 14:08:56 2025

Time Elapsed: 0:00:00.0

***** End: VERIFY CONNECTIVITY *****

Verification Complete : 0 Viols. 0 Wrngs.

(CPU Time: 0:00:00.0 MEM: 0.000M)

Comparison of Modular vs Behavioral

We can see that the behavioral version takes up less area (about $\frac{1}{3}$ less area) than the modular version. It also consumes less power as well and seems to be slower. The pad frame will be used with the Behavioral version.

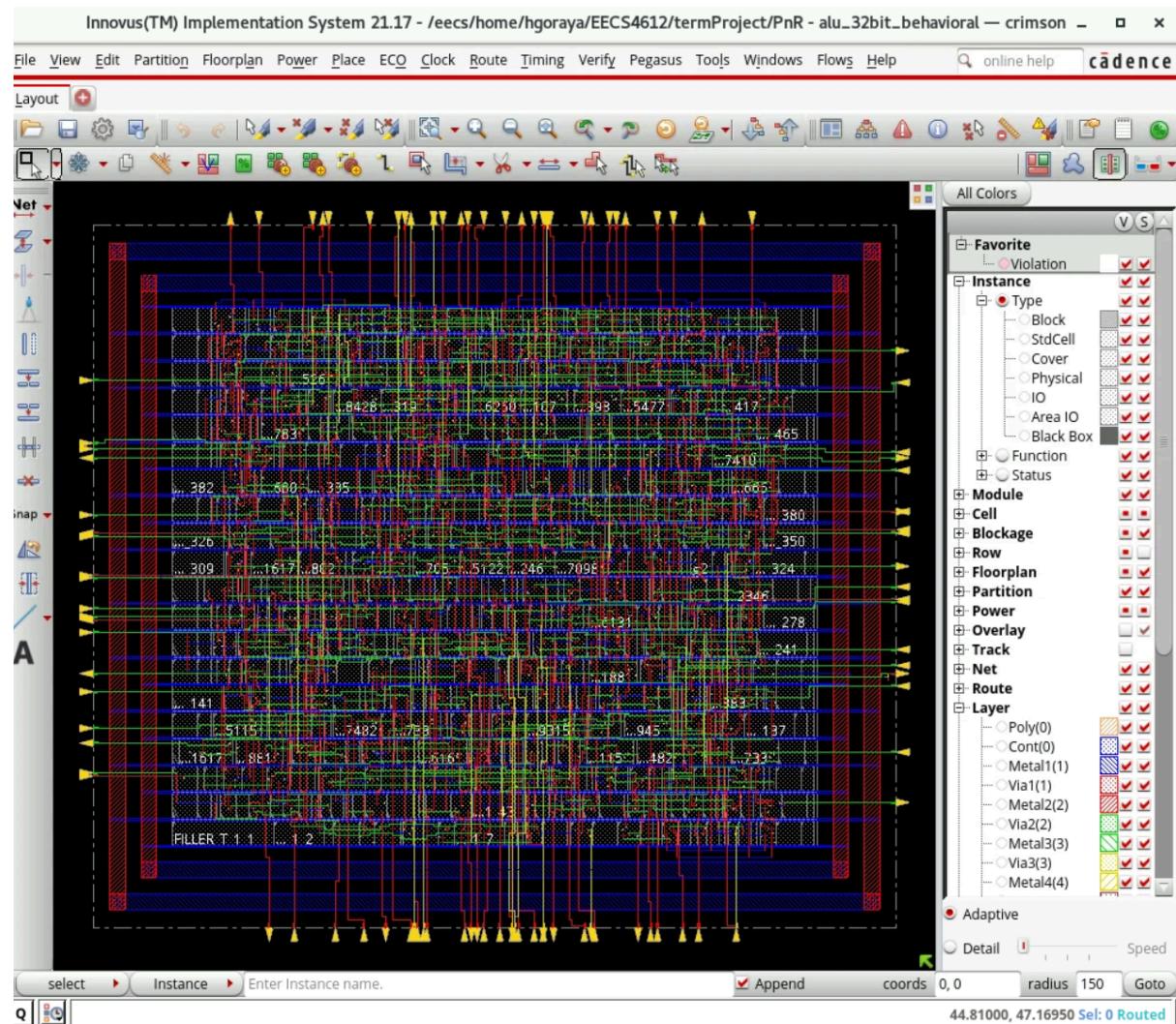


Figure 12: Innovus final design

This is the PnR design for the behavioral unit with all inputs and outputs.

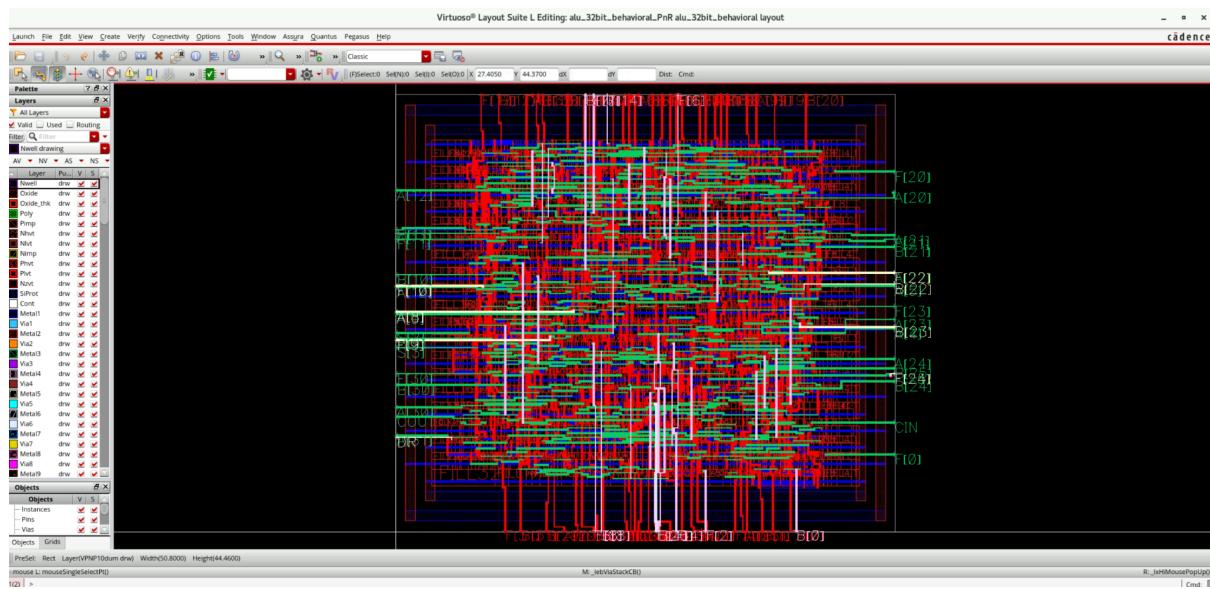


Figure 13: Virtuoso view

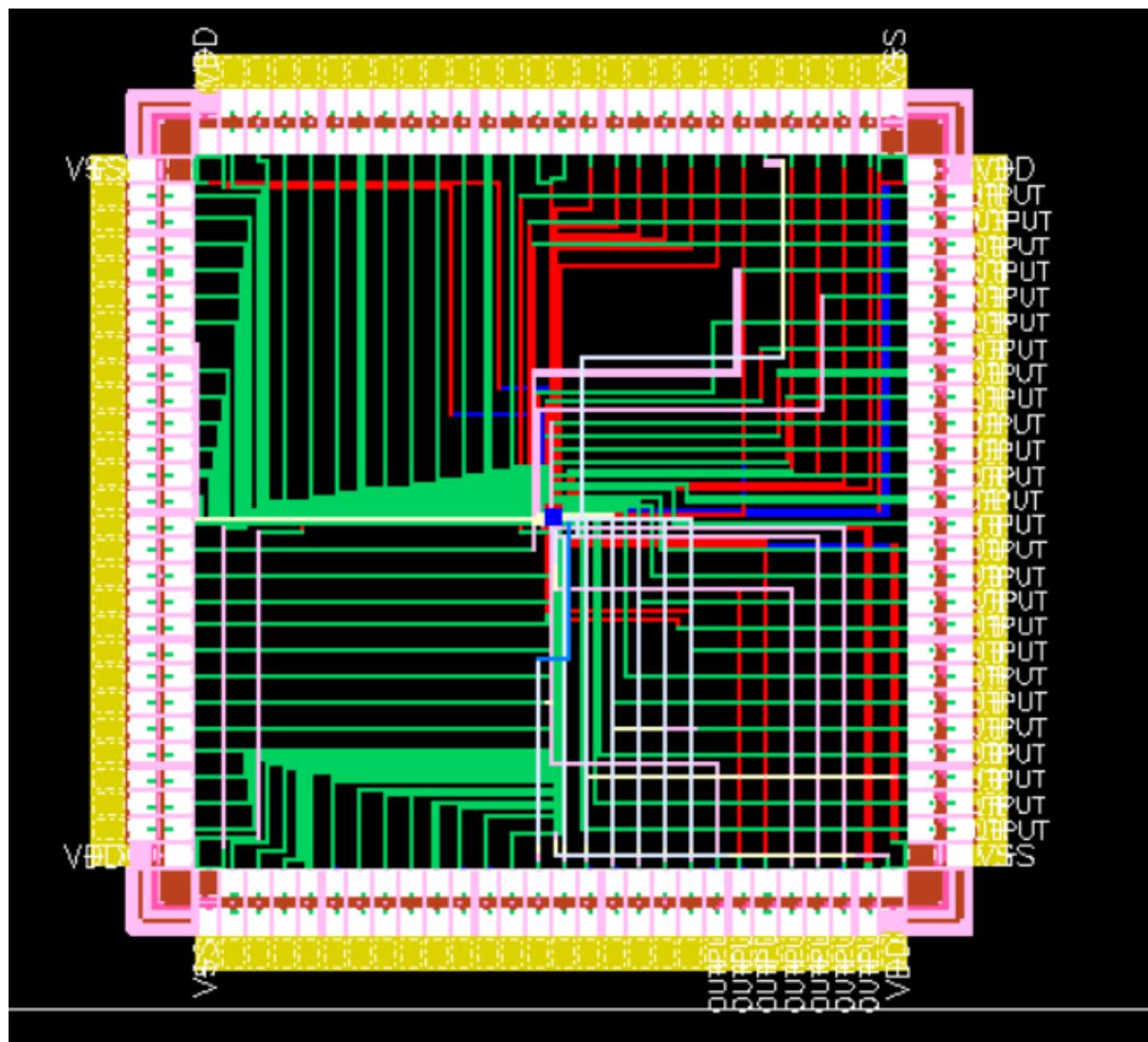
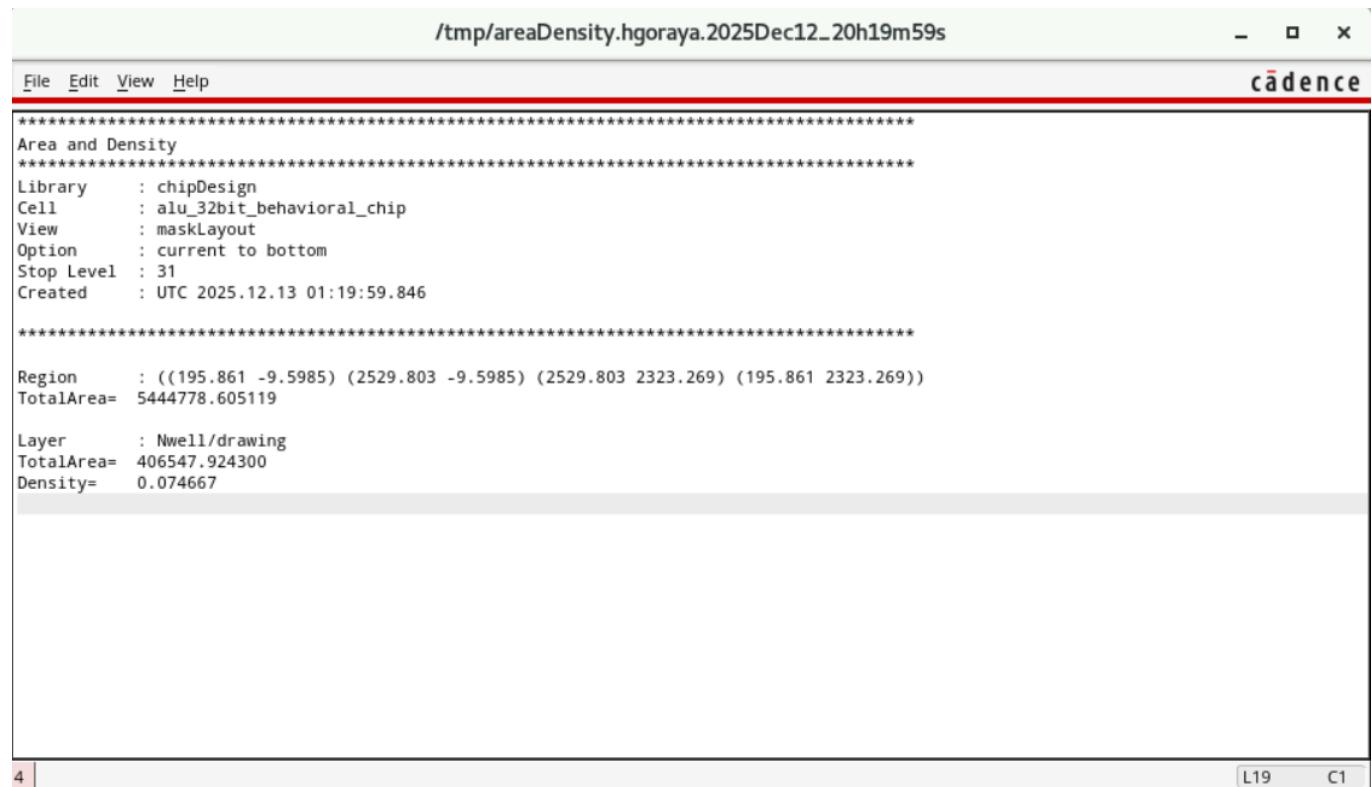


Figure 14: Final Chip Design

With the final chip design, there are some errors which seem to be from the pad frame design and the bondpad structures and not the connectivity or the core.



The screenshot shows a terminal window titled '/tmp/areaDensity.hgoraya.2025Dec12_20h19m59s' running on the Cadence software. The window displays the results of an area and density analysis for a specific chip design. The output includes details about the library, cell, view, options, stop level, and creation date. It also lists the region, total area, layer, and density.

```
*****  
Area and Density  
*****  
Library      : chipDesign  
Cell         : alu_32bit_behavioral_chip  
View         : maskLayout  
Option       : current to bottom  
Stop Level   : 31  
Created      : UTC 2025.12.13 01:19:59.846  
*****  
Region      : ((195.861 -9.5985) (2529.803 -9.5985) (2529.803 2323.269) (195.861 2323.269))  
TotalArea= 5444778.605119  
Layer        : Nwell/drawing  
TotalArea= 406547.924300  
Density=    0.074667  
*****
```

Figure 15: Area and Density

With Tools -> Area and Density calculator, we can see that the total area is 406,547.92.