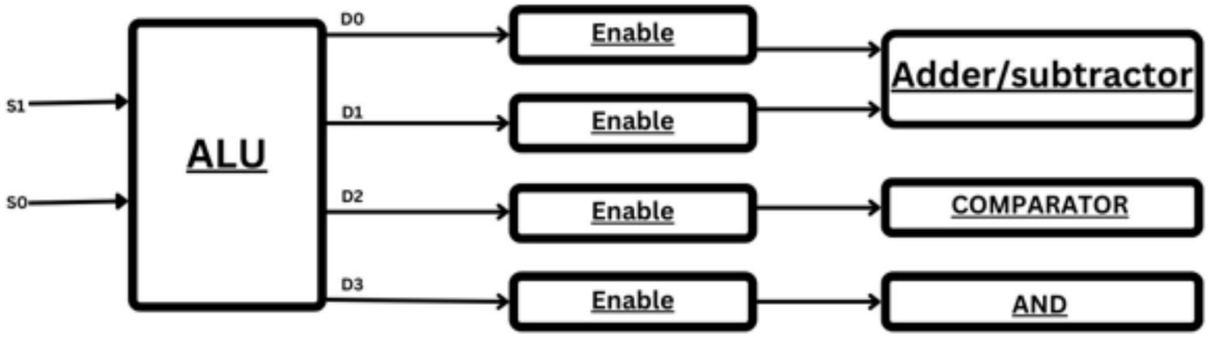
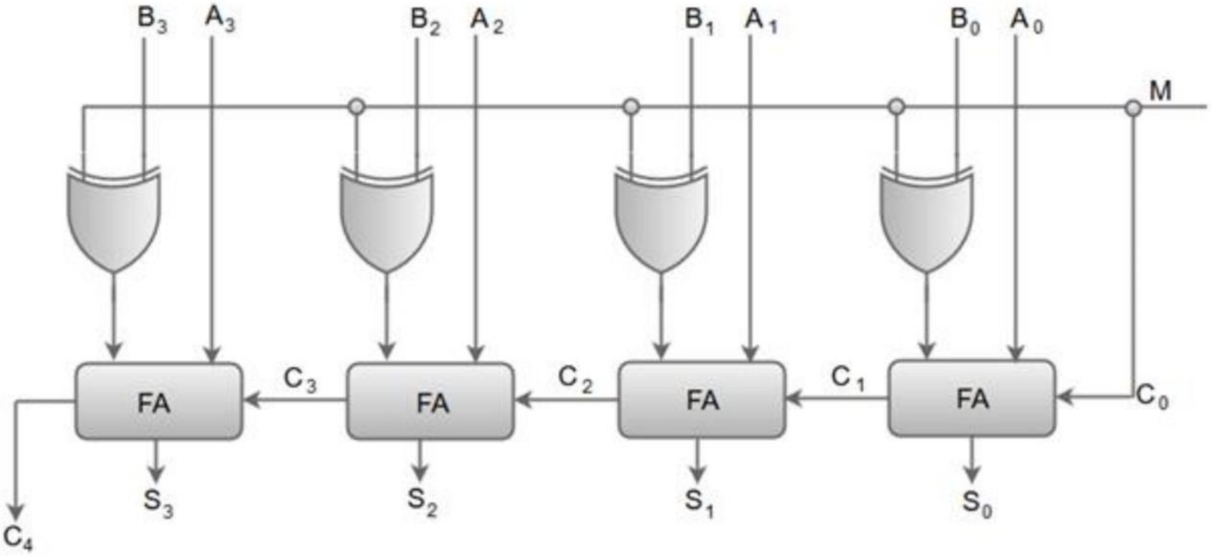
Circuit



Implemented ALU circuit includes logic gates and multiplexers for various operations controlled by select inputs. A 2:4 decoder manages operation selection. The 4-bit inputs are directed by the enable line, determined by select inputs. Modules (adder, subtractor, comparator, AND gate) operate based on enable signal and select inputs. Output pins connect to modules for accurate results in 4-bit addition, subtraction, comparison, and ANDing. Successful design and implementation enable efficient operation.

Adder-Subtractor



We made a fast adder and subtractor in our ALU using a special kind of adder called a carry-lookahead adder. It helps the ALU work quicker by reducing the time it takes to calculate certain things. For subtraction, we use a simple operation with an XOR gate and assume the numbers are in a specific format called 2's complement.

We also added a feature to our ALU that lets it add or subtract four numbers at once. This makes the ALU more versatile, and we used the same fast adder design to speed up the process. The subtraction is done using an XOR gate and a "Mode" input. This improvement allows our ALU to do a wider range of calculations efficiently.

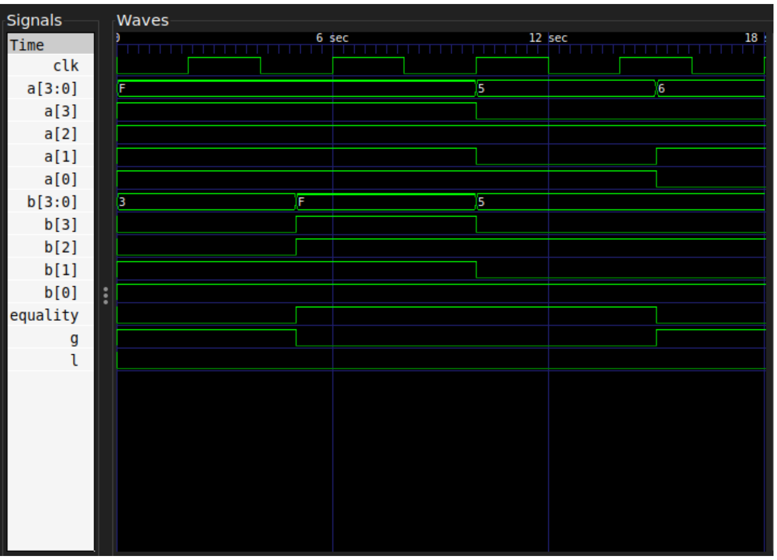
Comparator

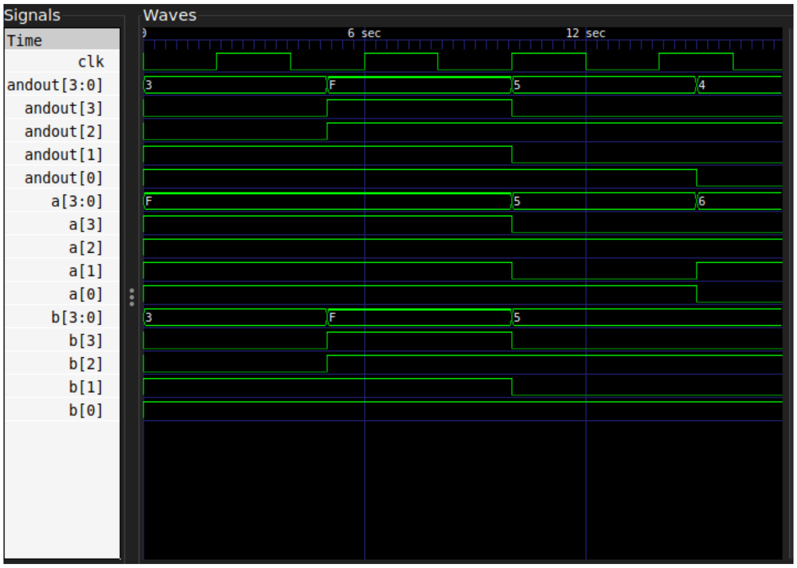
The comparator module in our ALU analyzes four-bit inputs, determining if they are equal, greater, or less than each other by comparing individual bits. Its output provides valuable information for decision-making in diverse applications.

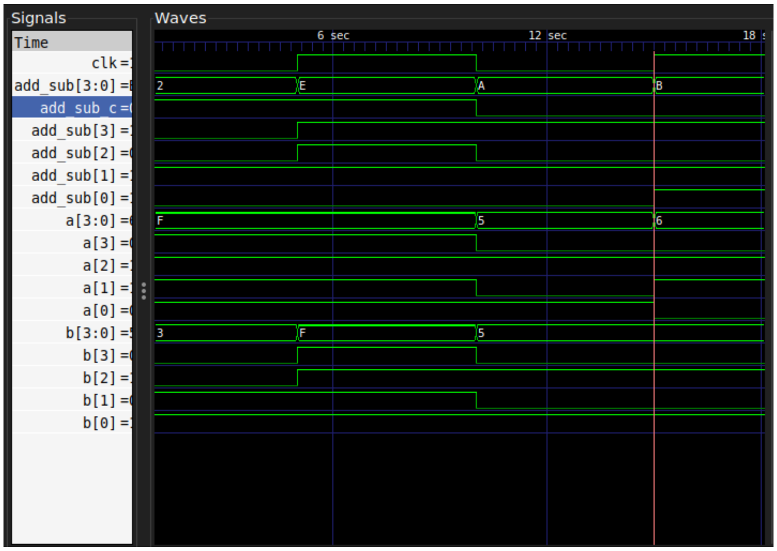
And Block

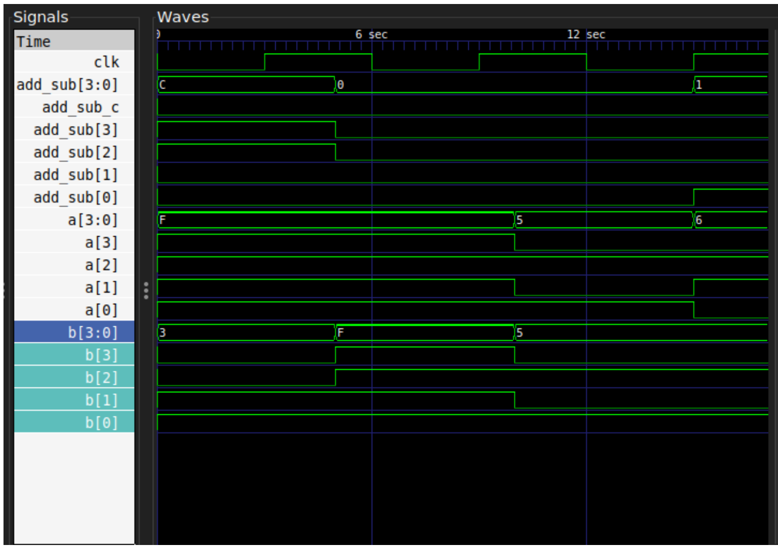
This module performs the AND operation with each bit in the 2 four input bits.

Verilog







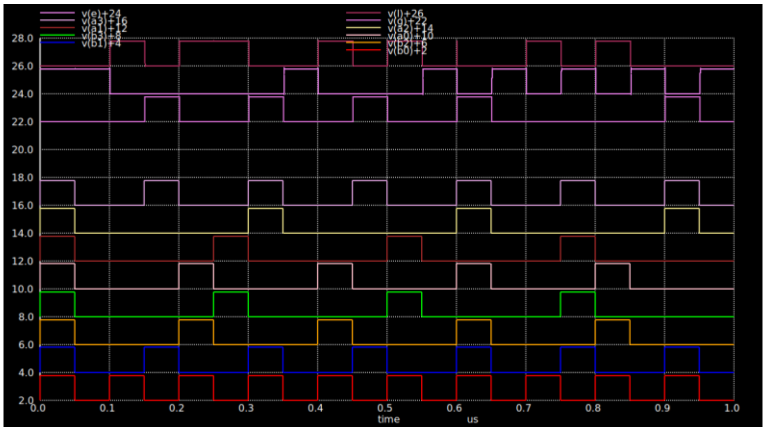


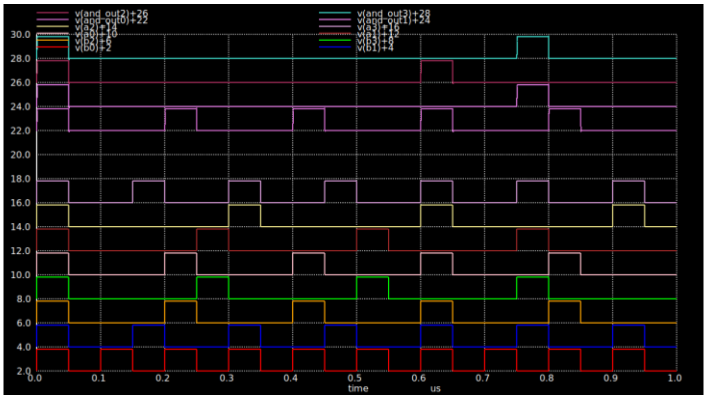
Implemented ALU using modular design, creating individual modules for each circuit block and connecting them to complete the circuit. Shared an enable block for the Adder and Subtractor, leveraging their common operation within the same module. Called all functions (modules) in the testbench and saved the output in a vcd file for viewing in gtkwave.

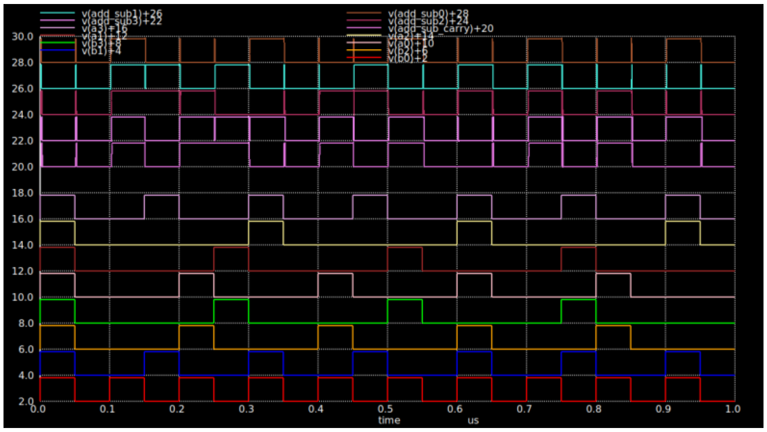
Mistakes

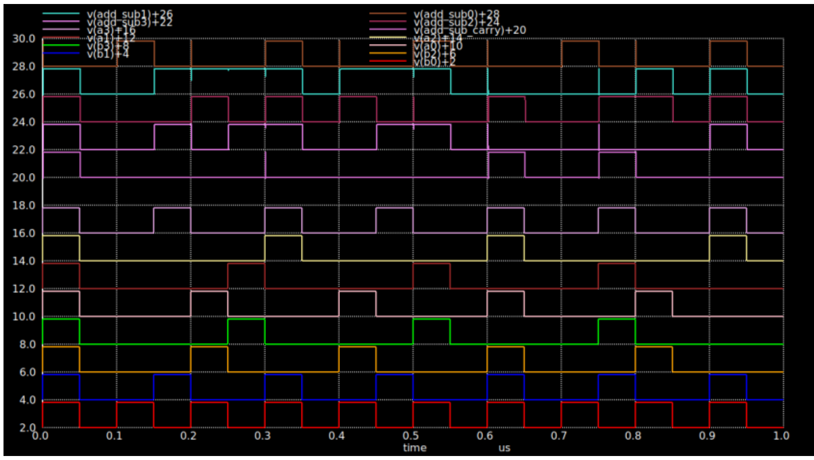
When my enable for comparator is not on, my output for equal to was always 1 which I changed by anding with the enable.

Ngspice







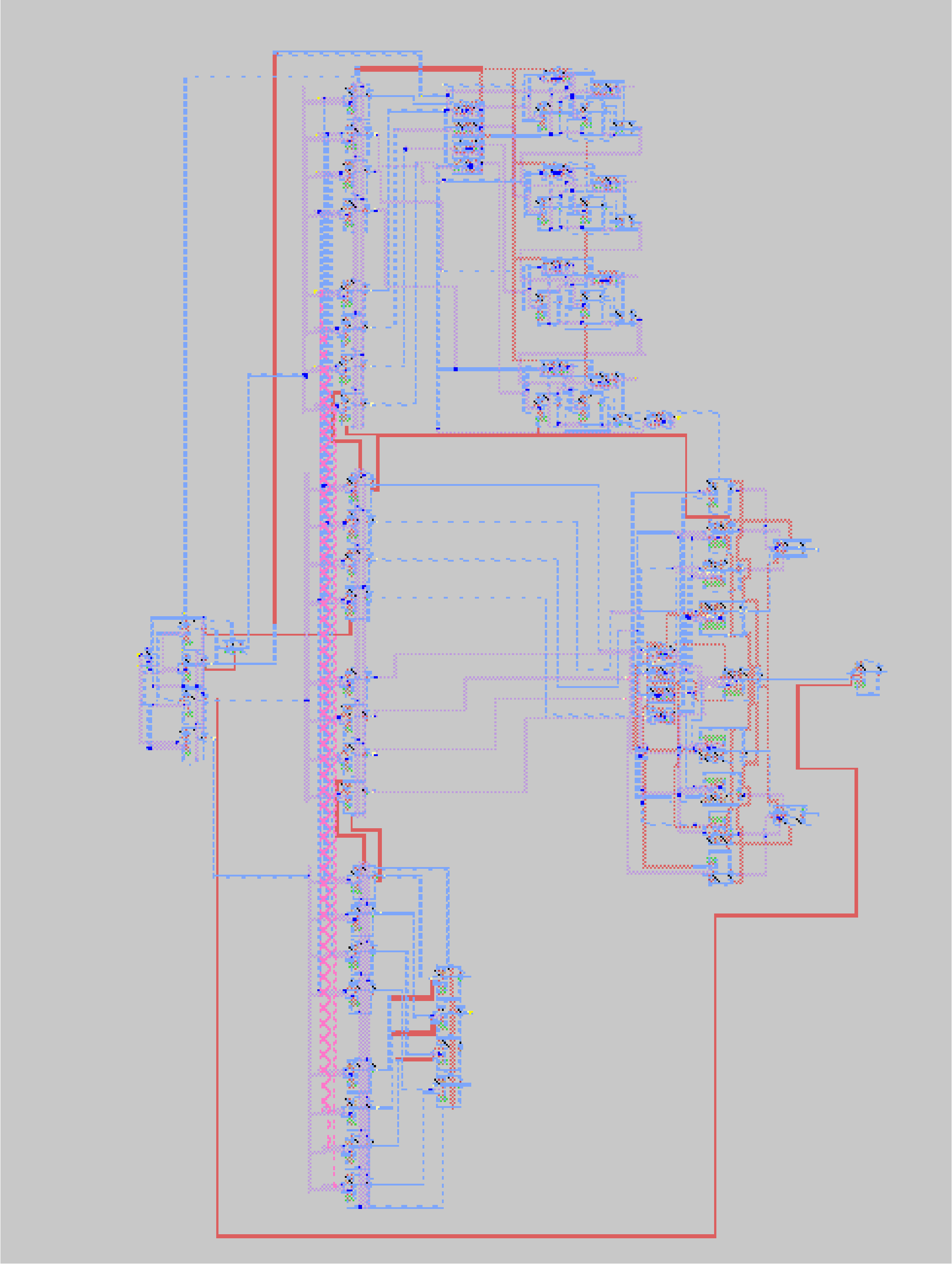


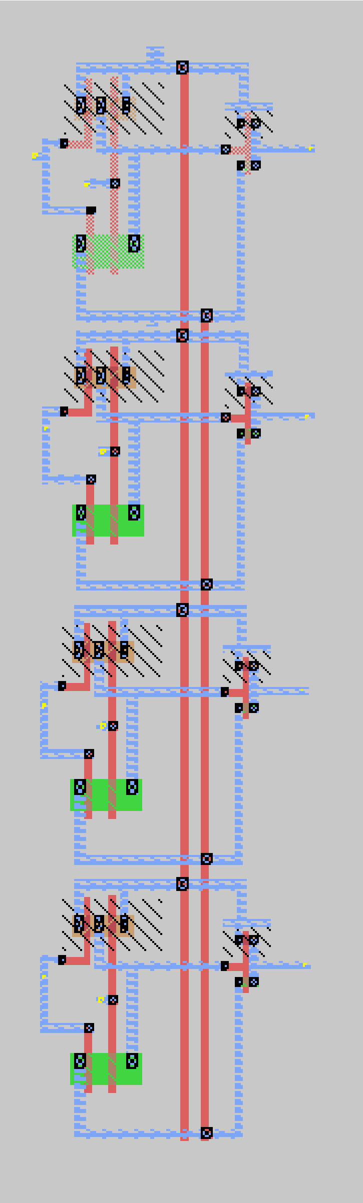
Mistakes

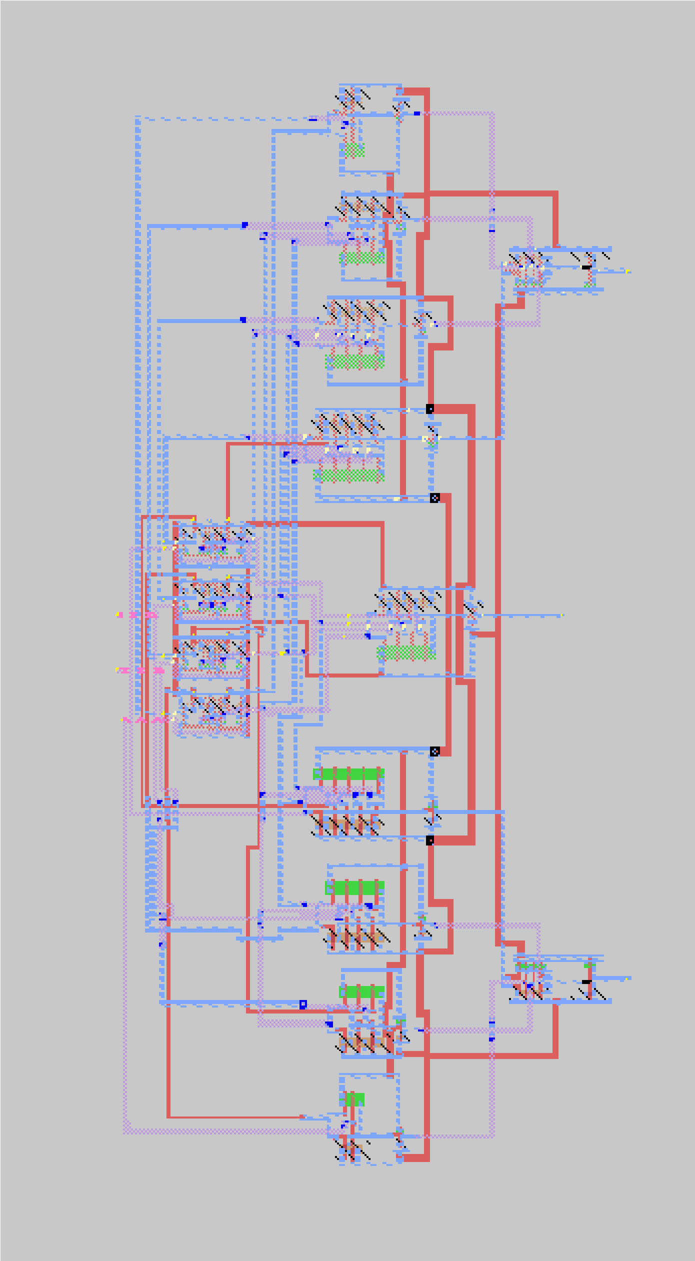
→ Irregular outputs caused confusion due to pulse inputs.

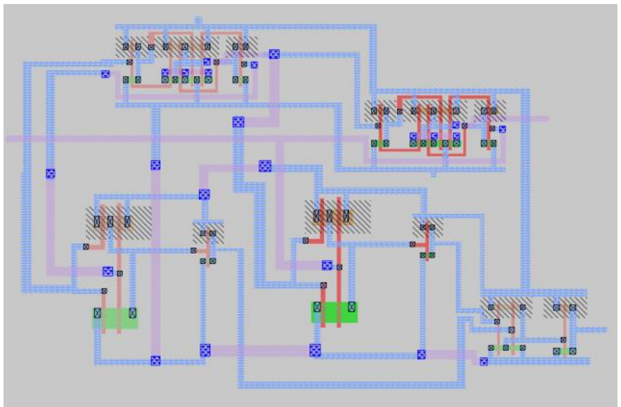
→ Forgot to ground some modules, resulting in incorrect outputs.

MAGIC









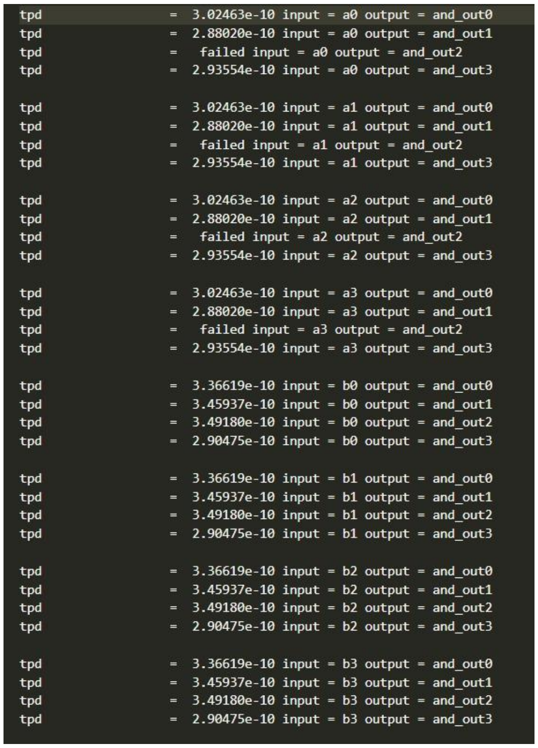
Mistakes

→Connected multiple wires of different labels which gave segmentation fault

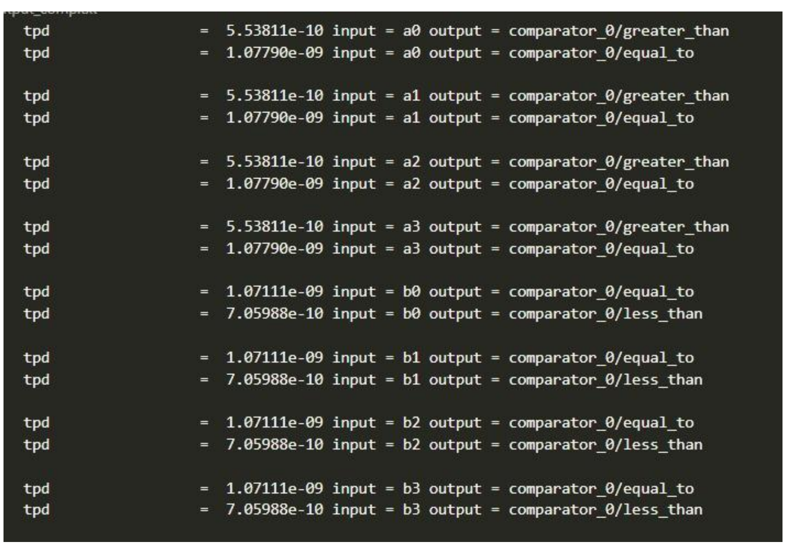
→Lot of overlapping of wires which gave errors and a series of wrong outputs

Delay Analysis

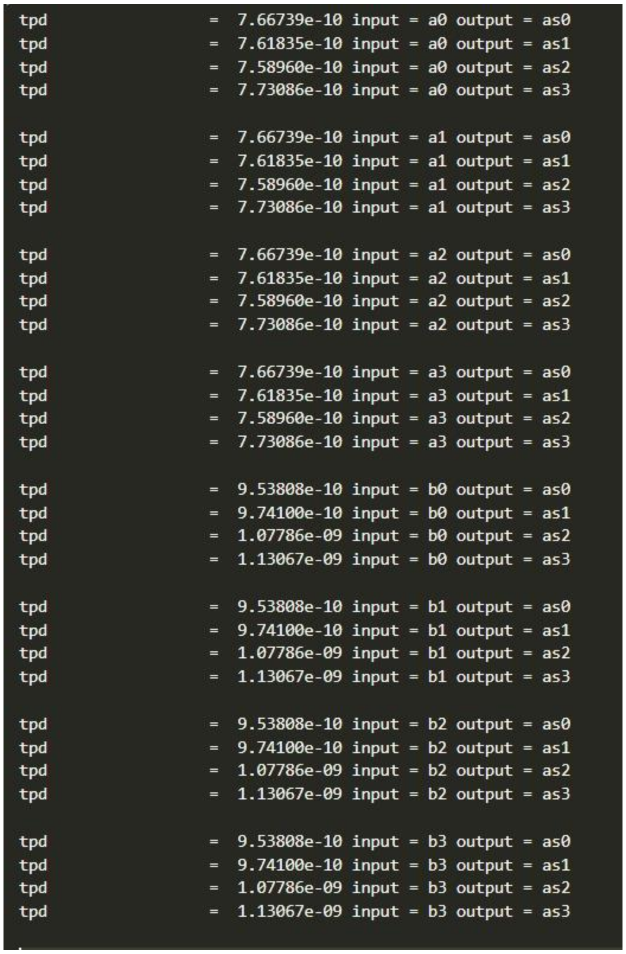
Used a Python script to iteratively run and analyze our scripts, determining delays for all inputs and outputs in the circuits. Delays in circuits can stem from factors like propagation delay, influenced by component speed, and capacitance/inductance effects, slowing down electrical signal flow.



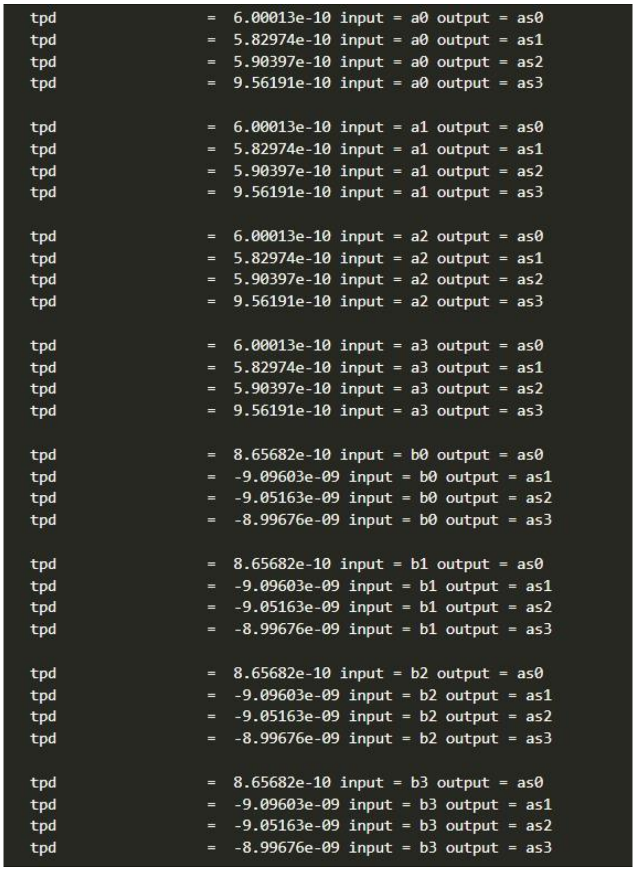
```
tpd                 =  3.02463e-10 input = a0 output = and_out0
tpd                 =  2.88020e-10 input = a0 output = and_out1
tpd                 =   failed input = a0 output = and_out2
tpd                 =  2.93554e-10 input = a0 output = and_out3

tpd                 =  3.02463e-10 input = a1 output = and_out0
tpd                 =  2.88020e-10 input = a1 output = and_out1
tpd                 =   failed input = a1 output = and_out2
tpd                 =  2.93554e-10 input = a1 output = and_out3

tpd                 =  3.02463e-10 input = a2 output = and_out0
tpd                 =  2.88020e-10 input = a2 output = and_out1
tpd                 =   failed input = a2 output = and_out2
tpd                 =  2.93554e-10 input = a2 output = and_out3

tpd                 =  3.02463e-10 input = a3 output = and_out0
tpd                 =  2.88020e-10 input = a3 output = and_out1
tpd                 =   failed input = a3 output = and_out2
tpd                 =  2.93554e-10 input = a3 output = and_out3

tpd                 =  3.36619e-10 input = b0 output = and_out0
tpd                 =  3.45937e-10 input = b0 output = and_out1
tpd                 =  3.49180e-10 input = b0 output = and_out2
tpd                 =  2.90475e-10 input = b0 output = and_out3

tpd                 =  3.36619e-10 input = b1 output = and_out0
tpd                 =  3.45937e-10 input = b1 output = and_out1
tpd                 =  3.49180e-10 input = b1 output = and_out2
tpd                 =  2.90475e-10 input = b1 output = and_out3

tpd                 =  3.36619e-10 input = b2 output = and_out0
tpd                 =  3.45937e-10 input = b2 output = and_out1
tpd                 =  3.49180e-10 input = b2 output = and_out2
tpd                 =  2.90475e-10 input = b2 output = and_out3

tpd                 =  3.36619e-10 input = b3 output = and_out0
tpd                 =  3.45937e-10 input = b3 output = and_out1
tpd                 =  3.49180e-10 input = b3 output = and_out2
tpd                 =  2.90475e-10 input = b3 output = and_out3
```



```
tpd                 =  5.53811e-10 input = a0 output = comparator_0/greater_than
tpd                 =  1.07790e-09 input = a0 output = comparator_0/equal_to

tpd                 =  5.53811e-10 input = a1 output = comparator_0/greater_than
tpd                 =  1.07790e-09 input = a1 output = comparator_0/equal_to

tpd                 =  5.53811e-10 input = a2 output = comparator_0/greater_than
tpd                 =  1.07790e-09 input = a2 output = comparator_0/equal_to

tpd                 =  5.53811e-10 input = a3 output = comparator_0/greater_than
tpd                 =  1.07790e-09 input = a3 output = comparator_0/equal_to

tpd                 =  1.07111e-09 input = b0 output = comparator_0/equal_to
tpd                 =  7.05988e-10 input = b0 output = comparator_0/less_than

tpd                 =  1.07111e-09 input = b1 output = comparator_0/equal_to
tpd                 =  7.05988e-10 input = b1 output = comparator_0/less_than

tpd                 =  1.07111e-09 input = b2 output = comparator_0/equal_to
tpd                 =  7.05988e-10 input = b2 output = comparator_0/less_than

tpd                 =  1.07111e-09 input = b3 output = comparator_0/equal_to
tpd                 =  7.05988e-10 input = b3 output = comparator_0/less_than
```



```
tpd                  =  7.66739e-10 input = a0 output = as0
tpd                  =  7.61835e-10 input = a0 output = as1
tpd                  =  7.58960e-10 input = a0 output = as2
tpd                  =  7.73086e-10 input = a0 output = as3

tpd                  =  7.66739e-10 input = a1 output = as0
tpd                  =  7.61835e-10 input = a1 output = as1
tpd                  =  7.58960e-10 input = a1 output = as2
tpd                  =  7.73086e-10 input = a1 output = as3

tpd                  =  7.66739e-10 input = a2 output = as0
tpd                  =  7.61835e-10 input = a2 output = as1
tpd                  =  7.58960e-10 input = a2 output = as2
tpd                  =  7.73086e-10 input = a2 output = as3

tpd                  =  7.66739e-10 input = a3 output = as0
tpd                  =  7.61835e-10 input = a3 output = as1
tpd                  =  7.58960e-10 input = a3 output = as2
tpd                  =  7.73086e-10 input = a3 output = as3

tpd                  =  9.53808e-10 input = b0 output = as0
tpd                  =  9.74100e-10 input = b0 output = as1
tpd                  =  1.07786e-09 input = b0 output = as2
tpd                  =  1.13067e-09 input = b0 output = as3

tpd                  =  9.53808e-10 input = b1 output = as0
tpd                  =  9.74100e-10 input = b1 output = as1
tpd                  =  1.07786e-09 input = b1 output = as2
tpd                  =  1.13067e-09 input = b1 output = as3

tpd                  =  9.53808e-10 input = b2 output = as0
tpd                  =  9.74100e-10 input = b2 output = as1
tpd                  =  1.07786e-09 input = b2 output = as2
tpd                  =  1.13067e-09 input = b2 output = as3

tpd                  =  9.53808e-10 input = b3 output = as0
tpd                  =  9.74100e-10 input = b3 output = as1
tpd                  =  1.07786e-09 input = b3 output = as2
tpd                  =  1.13067e-09 input = b3 output = as3
```



```
tpd                  =  6.00013e-10 input = a0 output = as0
tpd                  =  5.82974e-10 input = a0 output = as1
tpd                  =  5.90397e-10 input = a0 output = as2
tpd                  =  9.56191e-10 input = a0 output = as3

tpd                  =  6.00013e-10 input = a1 output = as0
tpd                  =  5.82974e-10 input = a1 output = as1
tpd                  =  5.90397e-10 input = a1 output = as2
tpd                  =  9.56191e-10 input = a1 output = as3

tpd                  =  6.00013e-10 input = a2 output = as0
tpd                  =  5.82974e-10 input = a2 output = as1
tpd                  =  5.90397e-10 input = a2 output = as2
tpd                  =  9.56191e-10 input = a2 output = as3

tpd                  =  6.00013e-10 input = a3 output = as0
tpd                  =  5.82974e-10 input = a3 output = as1
tpd                  =  5.90397e-10 input = a3 output = as2
tpd                  =  9.56191e-10 input = a3 output = as3

tpd                  =  8.65682e-10 input = b0 output = as0
tpd                  = -9.09603e-09 input = b0 output = as1
tpd                  = -9.05163e-09 input = b0 output = as2
tpd                  = -8.99676e-09 input = b0 output = as3

tpd                  =  8.65682e-10 input = b1 output = as0
tpd                  = -9.09603e-09 input = b1 output = as1
tpd                  = -9.05163e-09 input = b1 output = as2
tpd                  = -8.99676e-09 input = b1 output = as3

tpd                  =  8.65682e-10 input = b2 output = as0
tpd                  = -9.09603e-09 input = b2 output = as1
tpd                  = -9.05163e-09 input = b2 output = as2
tpd                  = -8.99676e-09 input = b2 output = as3

tpd                  =  8.65682e-10 input = b3 output = as0
tpd                  = -9.09603e-09 input = b3 output = as1
tpd                  = -9.05163e-09 input = b3 output = as2
tpd                  = -8.99676e-09 input = b3 output = as3
```

Critical Path

Identified the critical path in our ALU by calculating the maximum delay using a Python script. The critical path, from the inputs to the output of the adder-subtractor module, represents the most complex calculations. Addressing delays in this path optimizes the overall ALU performance.

Conclusion

In conclusion, our VLSI Final Project Report underscores the successful design and implementation of the ALU. Rigorous simulations with Verilog HDL and NgSpice validated its capacity for accurate 4-bit addition, subtraction, comparison, and ANDing operations. Further, we simulated the physical layout using MAGIC. A thorough delay analysis identified the critical path in the ALU design. Addressing delays along this critical path promises to optimize the overall performance of our ALU.Overall, the successful design and implementation of the ALU demonstrate our ability tocreate a functional and efficient circuit. The ALU's ability to perform various operations accurately makes it a valuable component in digital systems.