Designing a 4 Bits Carry Lookahead Adder

Task 1: Half Adder as a P & G Generator

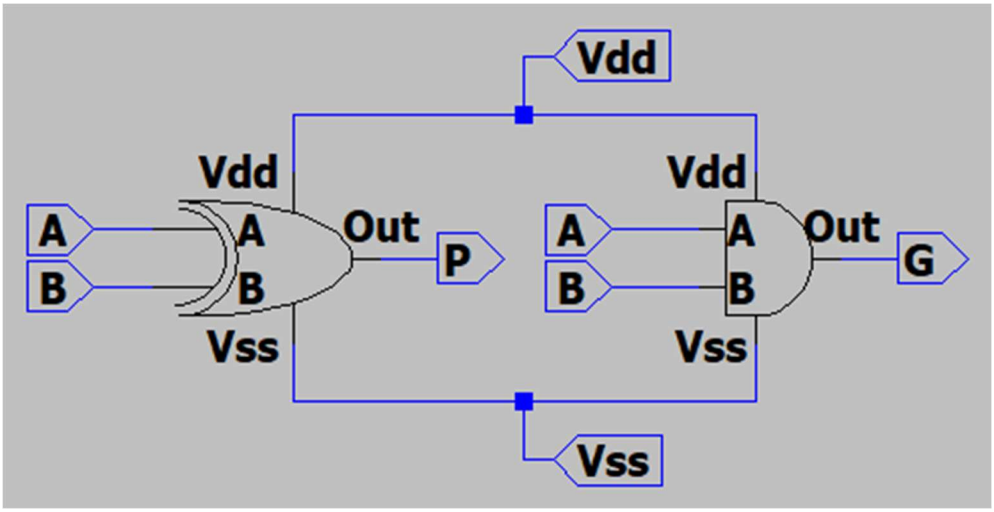


Figure 1: Half Adder Circuit as a PG Generator

A half adder was designed using one XOR gate and one AND gate. Since the Sum (S) and Carry (C) outputs will be renamed as P and G when forming the PG generator symbol, they are labeled accordingly in the PG Generator schematic.

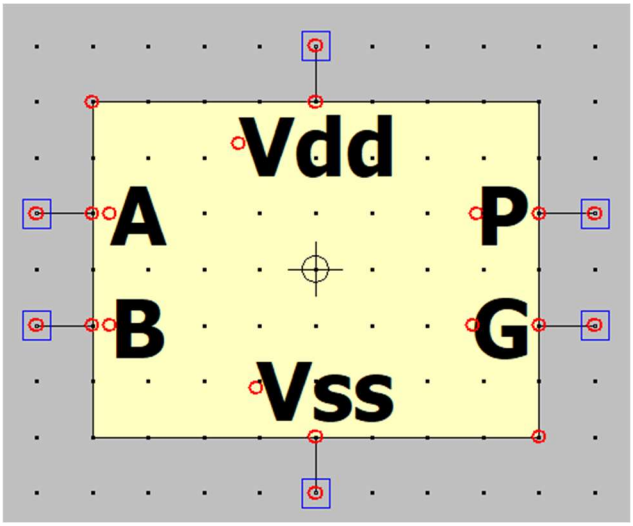
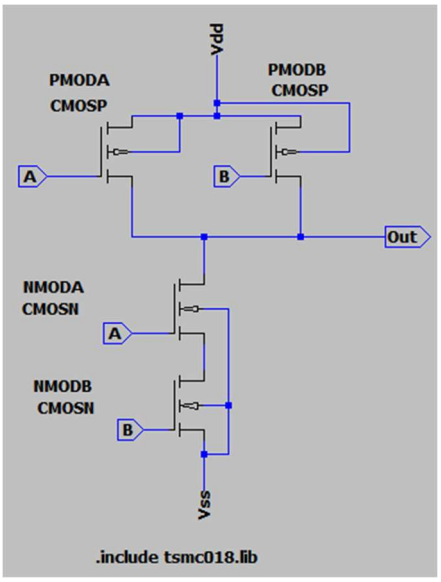


Figure 2: PG Generator symbol

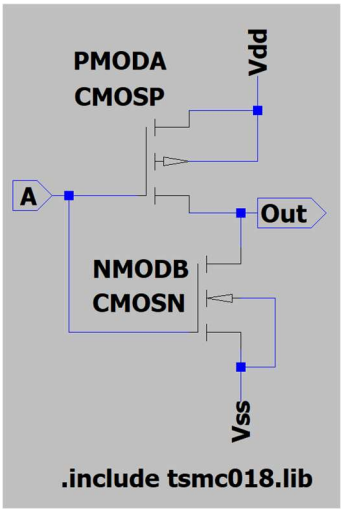


NAND1.sub - Notepad

File Edit Format View Help

```
.subckt NAND1 A B Out Vss Vdd

.inc tsmc018.lib
.lib standard.mos

M§PMODA Vdd A Out Vdd CMOSP l=1u w=2u
M§PMODB Vdd B Out Vdd CMOSP l=1u w=2u
M§NMODA Out A N001 Vss CMOSN l=1u w=1u
M§NMODB N001 B Vss Vss CMOSN l=1u w=1u

.model NMOS NMOS
.model PMOS PMOS

.ends NAND1
```

Figure 3: NAND Gate Schematic and Subcircuit File Adjustment



NOT1.sub - Notepad

File Edit Format View Help

```
.subckt NOT1 Vdd Vss A Out

.inc tsmc018.lib
.lib standard.mos

M§PMODA Vdd A Out Vdd CMOSP l=1u w=2u
M§NMODB Out A Vss Vss CMOSN l=1u w=1u

.model NMOS NMOS
.model PMOS PMOS

.ends NOT1
```

Figure 4: NOT Gate Schematic and Subcircuit File Adjustment

Since the AND gate consists of one NAND gate and one NOT gate, and the XOR gate consists of four NAND gates, the model directives for the NAND and NOT gates were removed. Then, the TSMC 180nm library was added to these two gates, and the names of the NMOS and PMOS transistors were changed to CMOSN and CMOSP. As specified in the assignment requirements, the lengths were set to 1 µm for both transistors. The widths were adjusted such that the PMOS width is twice that of the NMOS, resulting in 2 µm for CMOSP and 1 µm for CMOSN. The width and length values for the NAND and NOT gates are defined within their corresponding subcircuit files.

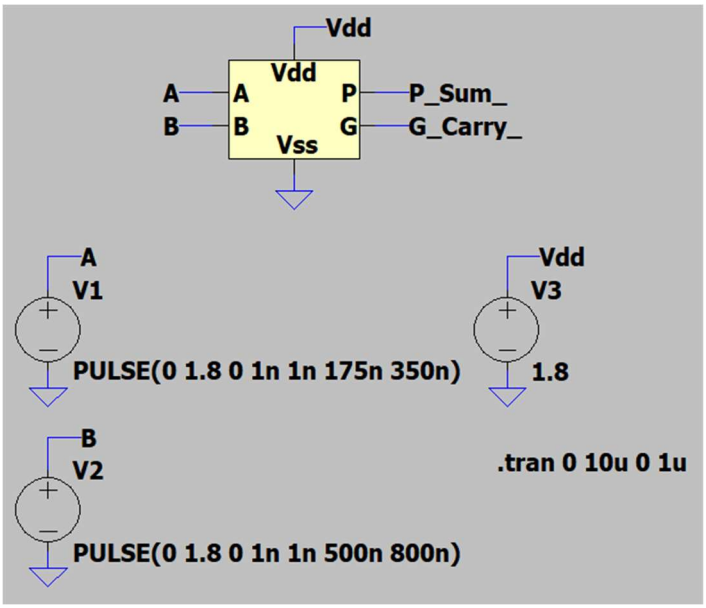


Figure 5: PG Generator Circuit for Simulation

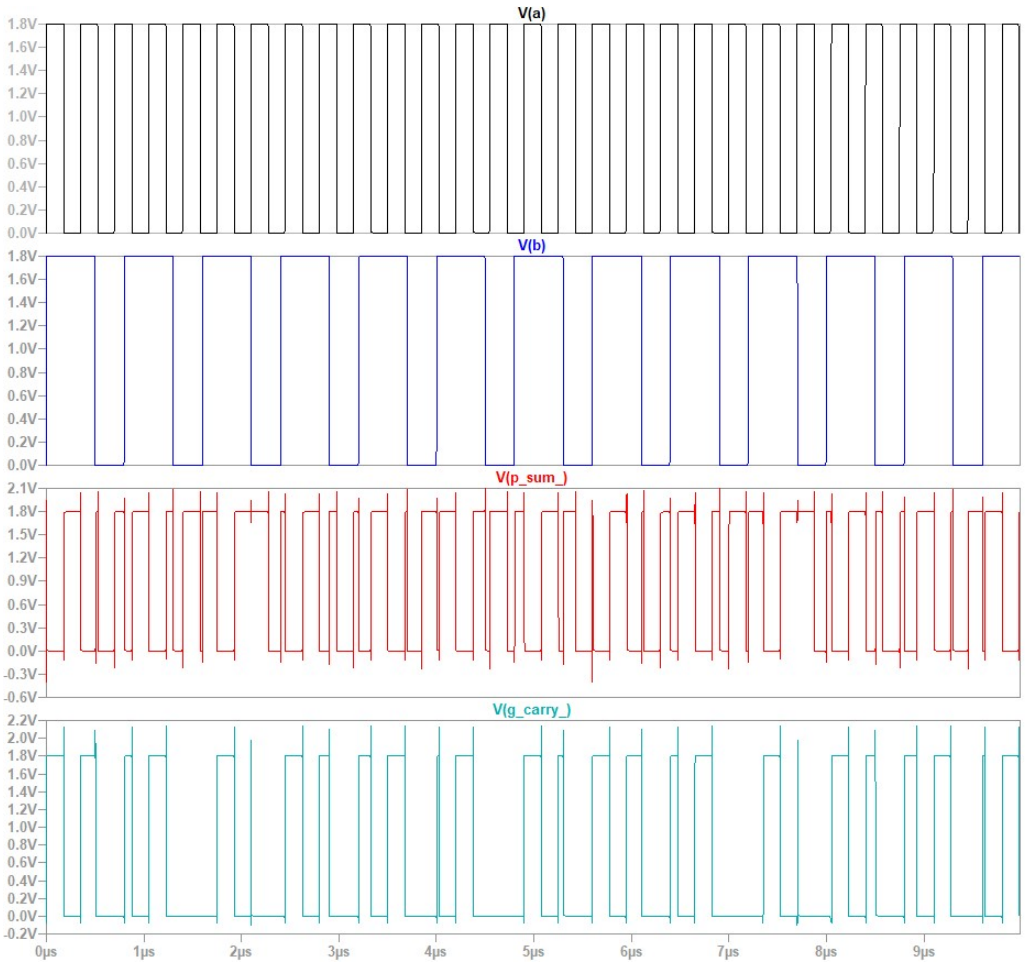


Figure 6: Simulation results of PG Generator circuit

Figure 5 and 6 shows the PG Generator (half adder) circuit along with its simulation results. The PG Generator design includes one XOR gate and one AND gate to generate the sum and carry outputs, as mentioned in Figure 1. Input signals A and B are applied to both gates. The output labeled P_Sum_ corresponds to the XOR of inputs A and B, while the output labeled G_Carry_ represents their AND result. The simulation was conducted using a supply voltage of 1.8V. Input A was defined using the pulse source PULSE(0 1.8 0 1n 1n 175n 350n), and input B was defined using PULSE(0 1.8 0 1n 1n 500n 800n), as specified in the assignment instructions. The transient analysis was run for 10 µs with a time step of 1 µs. As shown in the waveform results, the circuit behaves as expected: P_Sum_ is high when only one of the inputs is high, and G_Carry_ is high only when both inputs are high.

Task 2: 2-Bit Carry Lookahead Adder (CLA) Design

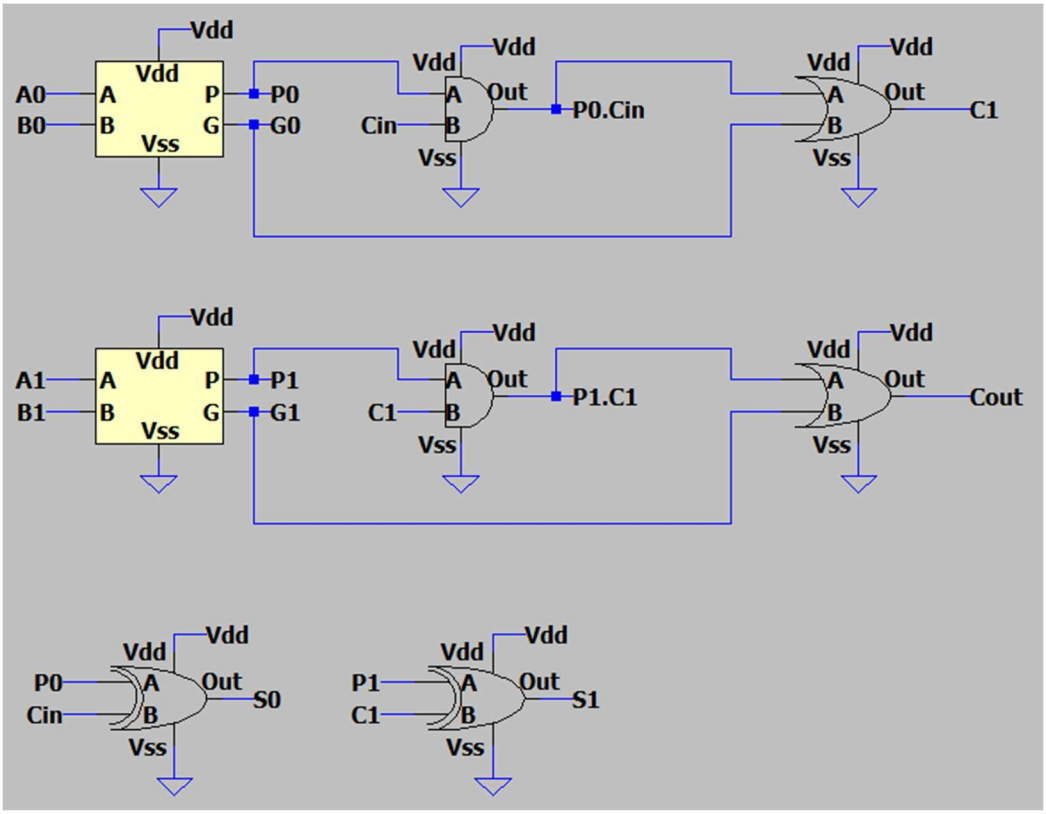


Figure 7: 2 Bit CLA Circuit Using PG Generators

The implementation of a 2-bit CLA circuit using two PG Generators is shown in Figure 7. The circuit schematic was constructed based on the formulas provided in the 'Generate and Propagate Logic', 'Carry Calculation', and 'Sum Calculation' sections of Background 6 in the assignment guidelines. The main formula used in the 'Carry Calculation' is 'Ci+1 = Gi + Pi·Ci', and the circuit is implemented accordingly to calculate C1 and C2 (Cout).

Formula Table:

Generate and Propagate logic	Gi = Ai.Bi Pi=Ai⊕Bi
Carry Calculation	Ci+1=Gi+Pi·Ci
Sum Calculation	Si = Pi⊕Ci

Step 1:

A0	A1	B0	B1	Cin (C0)
0V	0V	0V	0V	1.8V

Expected simulation results using the formula table:

S0	S1	C2 (Cout)
1.8V	0V	0V

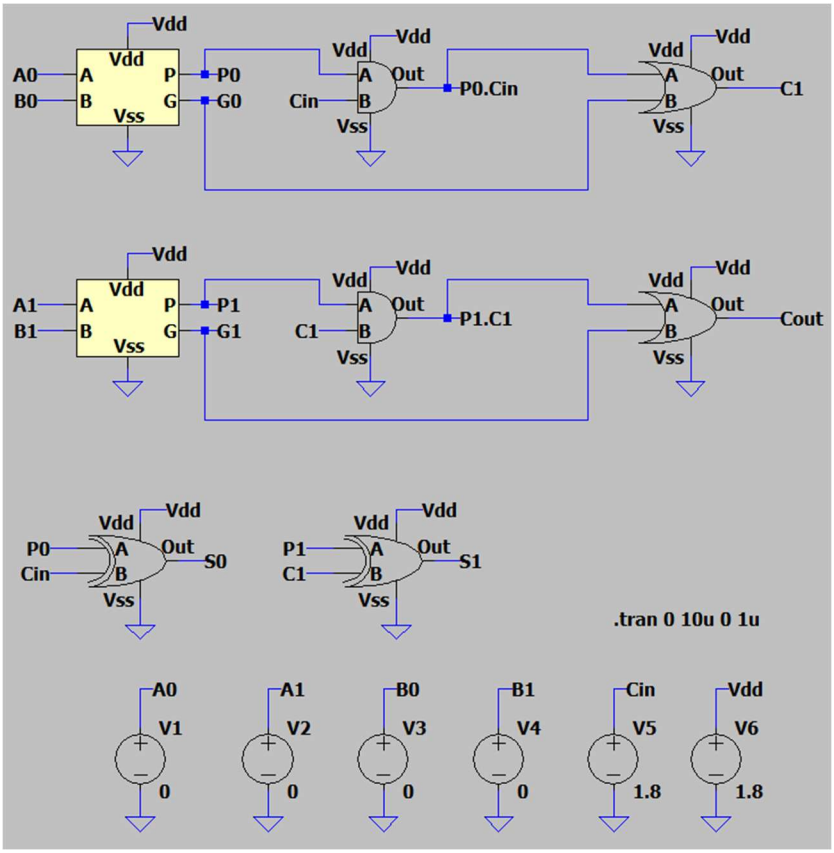


Figure 8: 2 Bit CLA circuit for Step 1 simulation

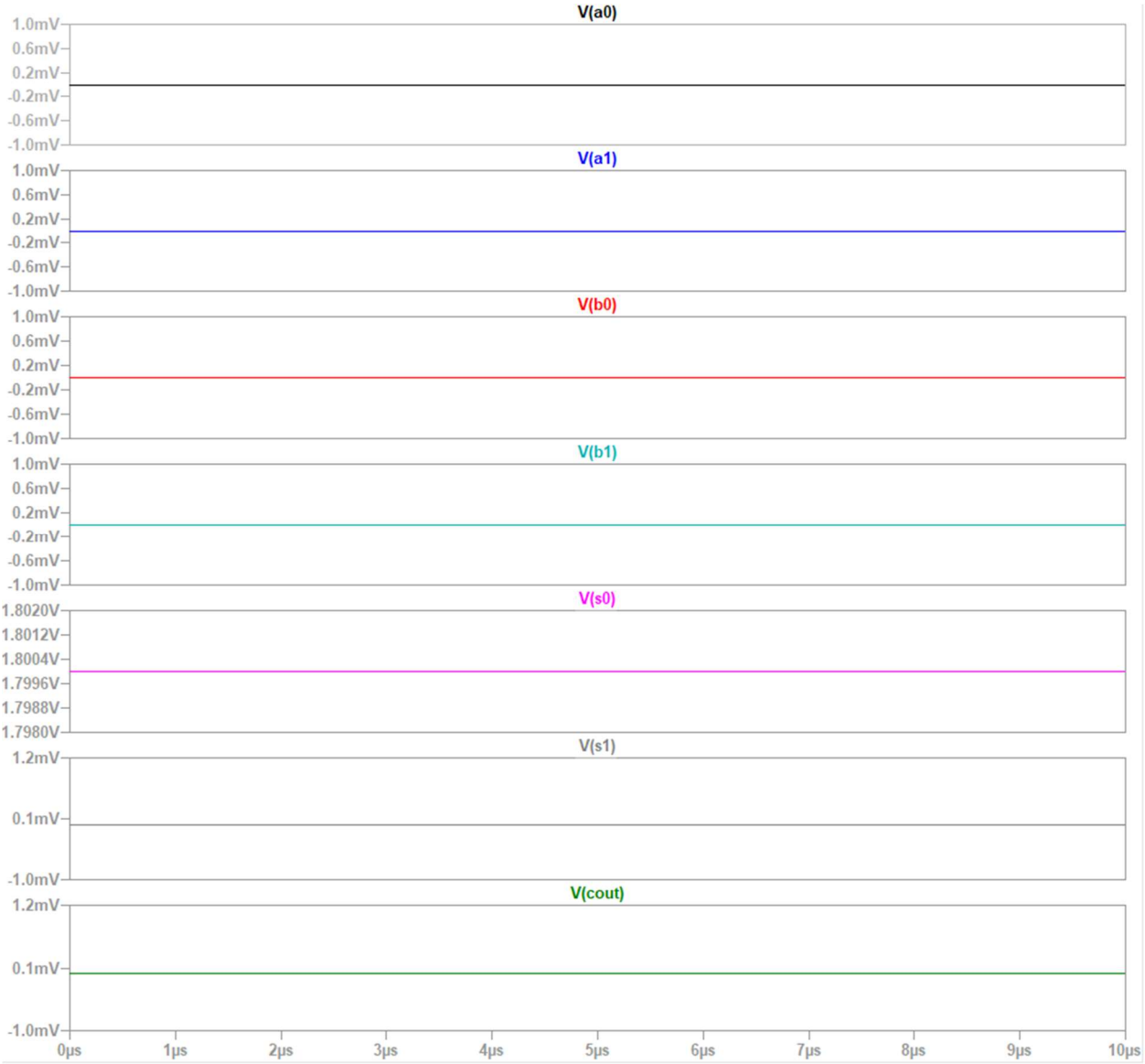


Figure 9: Simulation Results for Step 1

Step 2:

A0	A1	B0	B1	Cin (C0)
0V	1.8V	1.8V	0V	1.8V

Expected simulation results using the formula table:

S0	S1	C2 (Cout)
0V	0V	1.8V

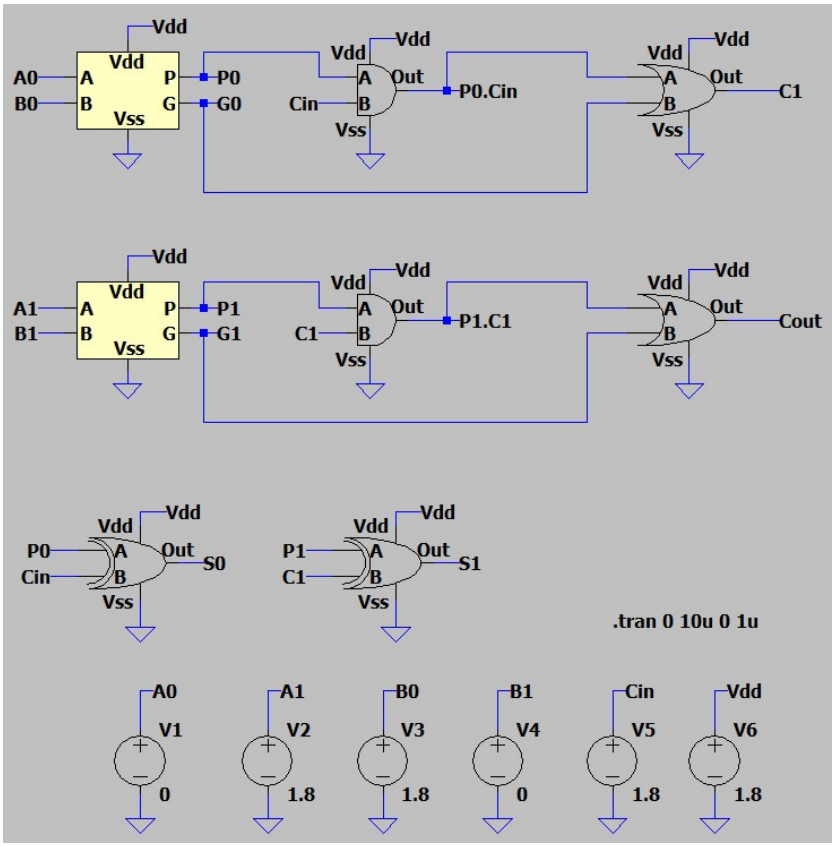


Figure 10: 2 Bit CLA circuit for Step 2 simulation

Figure 11: Simulation Results for Step 2

Step 3:

A0	A1	B0	B1	Cin (C0)
1.8V	1.8V	1.8V	0V	1.8V

Expected simulation results using the formula table:

S0	S1	C2 (Cout)
1.8V	0V	1.8V

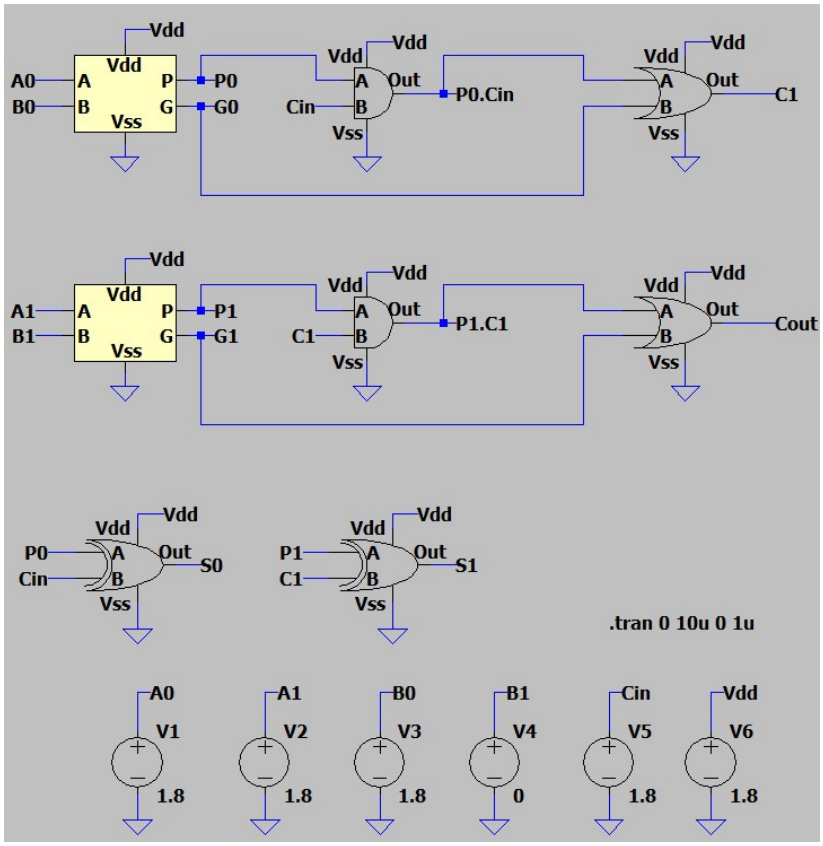


Figure 12: 2 Bit CLA circuit for Step 3 simulation

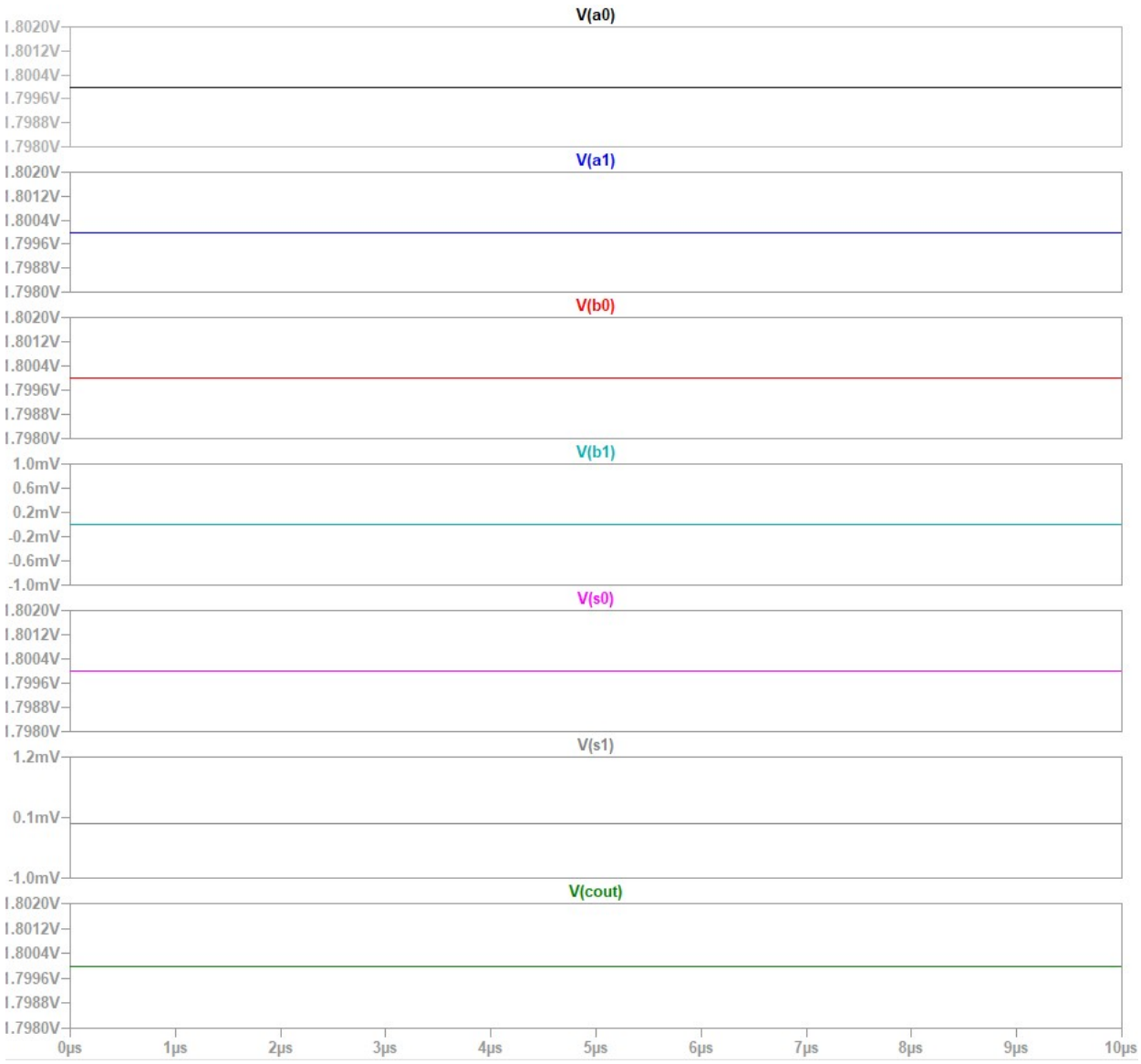


Figure 13: Simulation Results for Step 3

Step 4:

A0	A1	B0	B1	Cin (C0)
1.8V	1.8V	1.8V	1.8V	1.8V

Expected simulation results using the formula table:

S0	S1	C2 (Cout)
1.8V	1.8V	1.8V

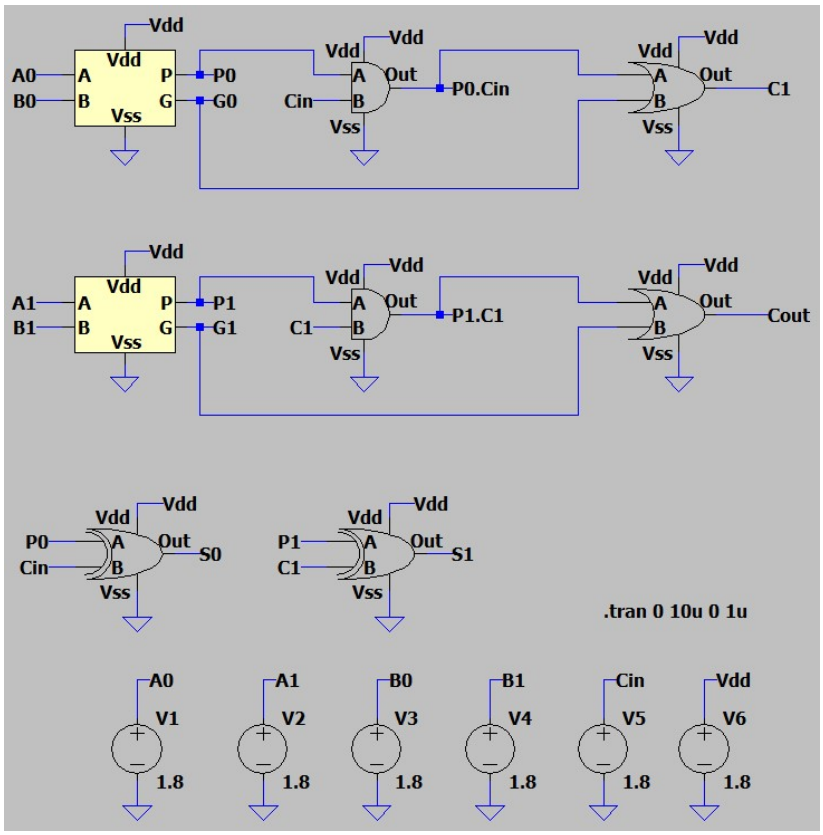


Figure 14: 2 Bit CLA circuit for Step 4 simulation

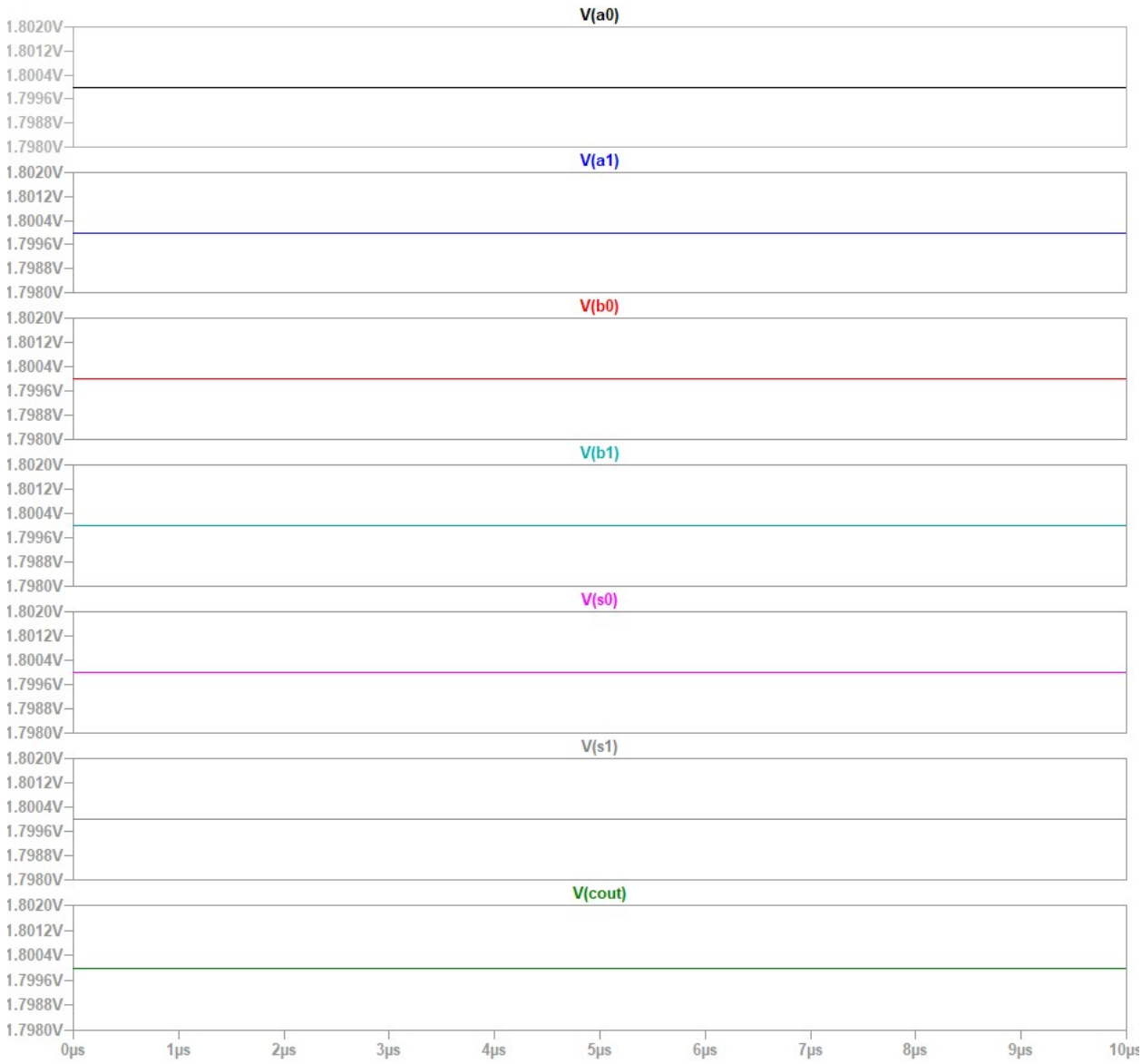


Figure 15: Simulation Results for Step 4

Final Result Table for Task 2:

	S0	S1	C2 (Cout)
Step 1	1.8V	0V	0V
Step 2	0V	0V	1.8V
Step 3	1.8V	0V	1.8V
Step 4	1.8V	1.8V	1.8V

The logic operations for the outputs S0, S1, and C2 (Cout), calculated using the formula table based on the voltage values of inputs A0, A1, B0, and B1 provided for the four steps in the assignment guidelines, were found to match the simulation results in all four steps. It was observed that the theoretical and experimental implementations of the 2-bit CLA circuit operated successfully.

Task 3: 4-Bit Carry Lookahead Adder (CLA) from 2-Bit CLA's combined with RCA logic

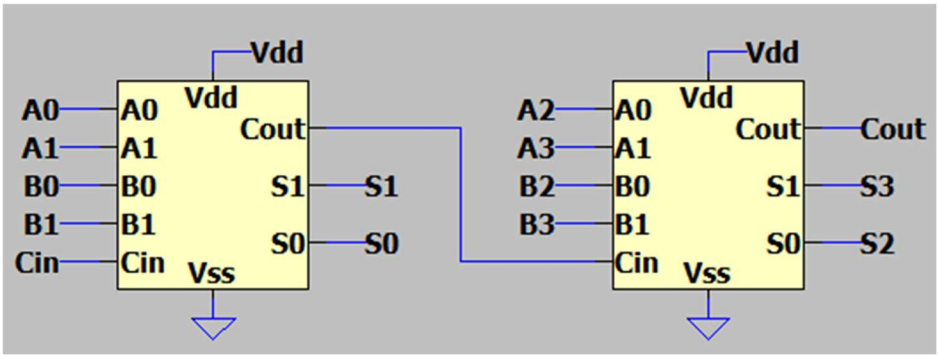


Figure 16: 4 Bit CLA circuit using combination of Two 2 Bit CLA's symbol

The 4-bit CLA circuit was designed by cascading two 2-bit CLA modules in a Ripple Carry Adder (RCA) format, as shown in the Figure 16. The first 2-bit CLA takes inputs A0–A1, B0–B1, and Cin, and produces outputs S0, S1, and an intermediate carry-out (Cout). This carry-out is then connected to the Cin input of the second 2-bit CLA module, which receives A2–A3 and B2–B3 as its additional inputs and produces outputs S2, S3, and the final carry-out. Both modules are supplied with Vdd and grounded through Vss. This structure effectively implements a 4-bit adder using two hierarchical 2-bit CLA blocks, preserving the propagate-generate logic for fast carry computation.

Formula Table:

Generate and Propagate logic	$G_i = A_i.B_i$ $P_i = A_i \oplus B_i$
Carry Calculation	$C_{i+1} = G_i + P_i \cdot C_i$
Sum Calculation	$S_i = P_i \oplus C_i$

The same calculation method used in the formula table for the 2-bit CLA design was applied to the 4-bit CLA to obtain the values of S0, S1, S2, S3, and Cout.

Step 1:

Cin	A0	A1	A2	A3	B0	B1	B2	B3
1.8V	0V	0V	1.8V	0V	0V	0V	1.8V	0V

Expected simulation results using the formula table:

S0	S1	S2	S3	Cout
1.8V	0V	0V	1.8V	0V

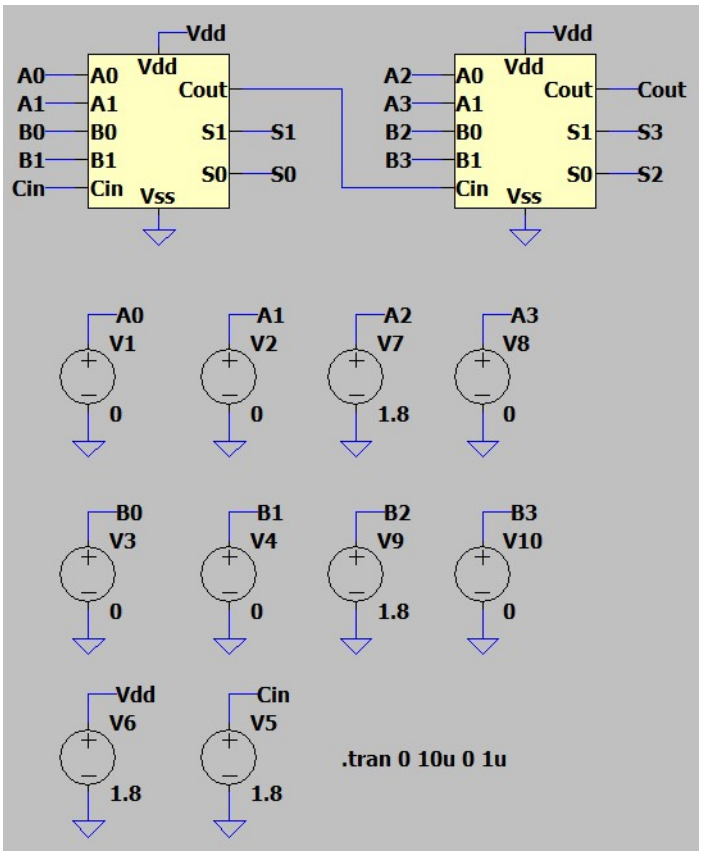


Figure 17: 4 Bit CLA circuit for Step 1 Simulation

Figure 18: Simulation Results for Step 1

Step 2:

Cin	A0	A1	A2	A3	B0	B1	B2	B3
1.8V	0V	1.8V	0V	1.8V	1.8V	0V	1.8V	0V

Expected simulation results using the formula table:

S0	S1	S2	S3	Cout
0V	0V	0V	0V	1.8V

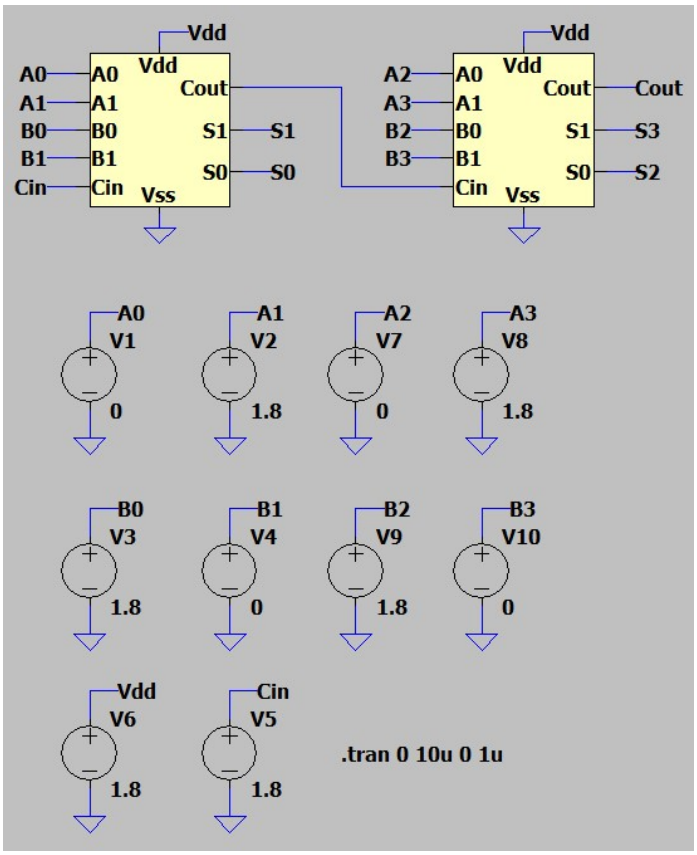


Figure 19: 4 Bit CLA circuit for Step 2 Simulation

Figure 20: Simulation Results for Step 2

Step 3:

Cin	A0	A1	A2	A3	B0	B1	B2	B3
1.8V	1.8V	1.8V	0V	1.8V	1.8V	0V	1.8V	0V

Expected simulation results using the formula table:

S0	S1	S2	S3	Cout
1.8V	0V	0V	0V	1.8V

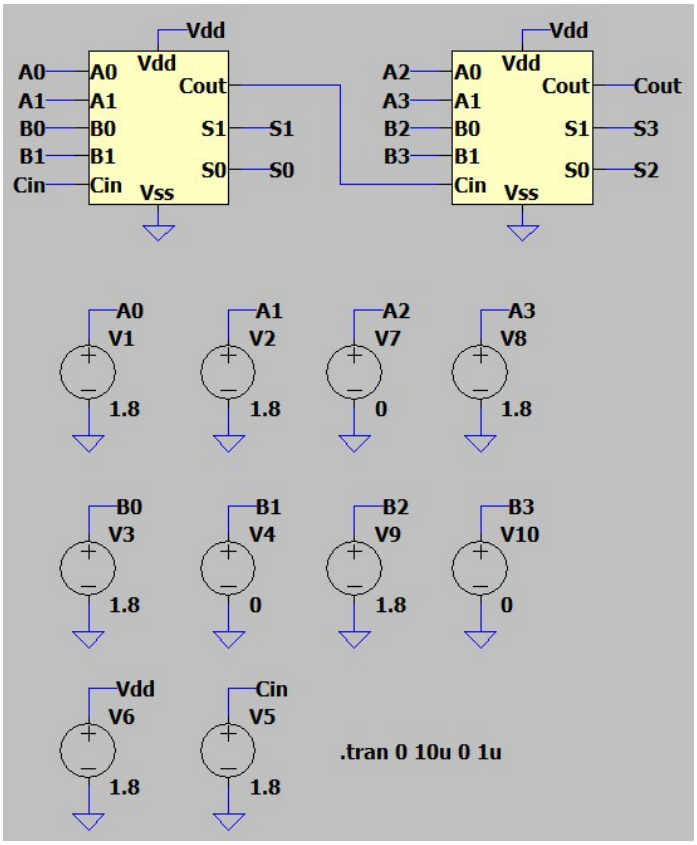


Figure 21: 4 Bit CLA circuit for Step 3 Simulation

Figure 22: Simulation Results for Step 3

Step 4:

Cin	A0	A1	A2	A3	B0	B1	B2	B3
1.8V	1.8V	1.8V	1.8V	1.8V	1.8V	1.8V	1.8V	1.8V

Expected simulation results using the formula table:

S0	S1	S2	S3	Cout
1.8V	1.8V	1.8V	1.8V	1.8V

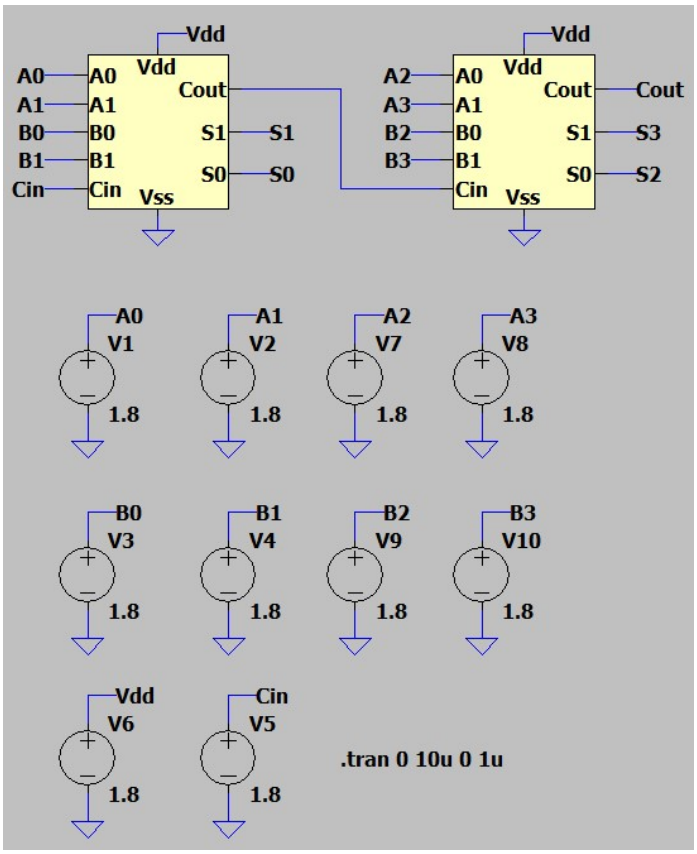


Figure 23: 4 Bit CLA circuit for Step 4 Simulation

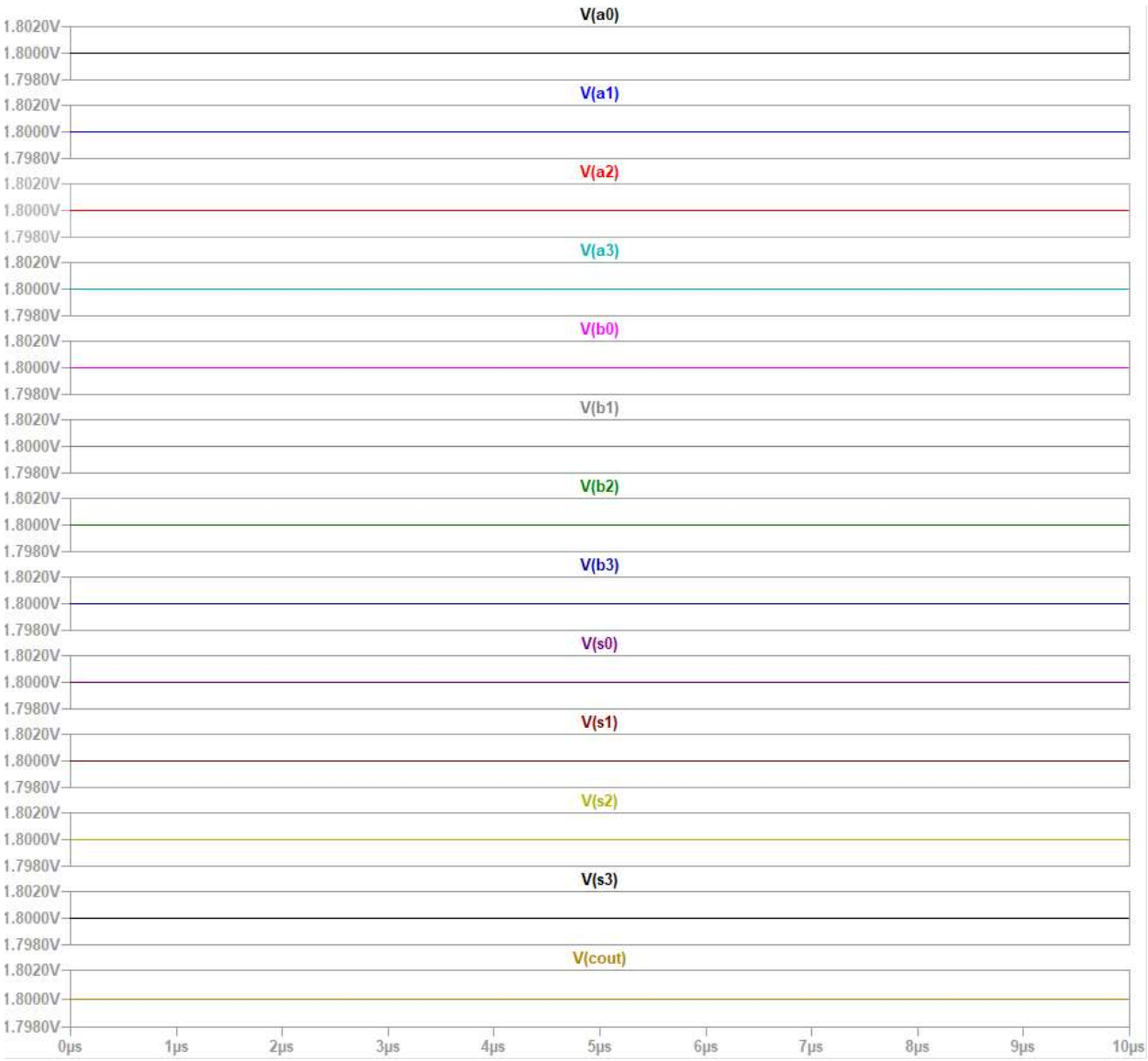


Figure 24: Simulation Results for Step 4

Final Result Table for Task 3:

	S0	S1	S2	S3	Cout
Step 1	1.8V	0V	0V	1.8V	0V
Step 2	0V	0V	0V	0V	1.8V
Step 3	1.8V	0V	0V	0V	1.8V
Step 4	1.8V	1.8V	1.8V	1.8V	1.8V

The logic operations for the outputs S0, S1, S2, S3, and Cout, calculated using the formula table based on the voltage values of inputs A0 to A3 and B0 to B3 provided for the four steps in the assignment guidelines, were found to match the simulation results in all steps. It was observed that the theoretical and experimental implementations of the 4-bit CLA circuit operated successfully.

Discussion of Design Area

For Task 1:

As the first task, a half adder was designed as a PG generator. Its construction involved one XOR gate and one AND gate. The XOR gate consists of four NAND gates, and the AND gate is made up of one NAND gate and one NOT gate. Each NAND gate is built using two PMOS and two NMOS transistors, while each NOT gate consists of one PMOS and one NMOS transistor.

NAND Gate → 2 x PMOS + 2 x NMOS

NOT Gate → 1 x PMOS + 1 x NMOS

PG Generator = [XOR → 4 x NAND] + [AND → 1 x NAND + 1 x NOT] = 22 transistors (11 x PMOS + 11 x NMOS)

Total transistor count of PG Generator = 22 transistors

11 x PMOS = 11 x 2µm

11 x NMOS = 11 x 1µm

Total width of PG Generator = 33µm

For Task 2:

As the second task, 2 Bit CLA was designed using 2 PG Generators, 2 AND Gates, 2 OR Gates and 2 XOR Gates.

OR Gate → 1 x NAND + 2 x NOT

2 Bit CLA = [2 x PG Generator] + [2 x AND] + [2 x OR] + [2 x XOR]

Total transistor count of 2 Bit CLA = 104 transistors (52 x PMOS + 52 x NMOS)

52 x PMOS = 52 x 2µm

52 x NMOS = 52 x 1µm

Total width of 2 Bit CLA = 156µm

For Task 3:

As the third task, 4 Bit CLA was designed using two 2 Bit CLAs.

4 Bit CLA = [2 x 2 Bit CLAs]

Total transistor count of 4 Bit CLA = 208 transistors (104 x PMOS + 104 x NMOS)

104 x PMOS = 104 x 2µm

104 x NMOS = 104 x 1µm

Total width of 4 Bit CLA = 312µm

	Transistor Count	PMOS Count	NMOS Count	Total Width	Total Length
Task 1	22	11	11	33 µm	22 µm
Task 2	104	52	52	156 µm	104 µm
Task 3	208	104	104	312 µm	208 µm

Discussion of Delay in Designs

In Task 1, the PG Generator was implemented using one XOR gate and one AND gate. The XOR gate was built from four NAND gates, while the AND gate was made using one NAND and one NOT gate. Since the XOR gate uses more logic gates, it causes more delay compared to the AND gate. Therefore, the critical path in this design is from the inputs A and B to the output P (Sum), which passes through the XOR logic. The Carry output (G) has a shorter path and less delay since it goes through fewer gates.

In Task 2, the 2-bit CLA circuit was designed using two PG Generators along with additional AND, OR, and XOR gates for carry and sum computation. The delay in this circuit is primarily determined by the carry propagation path, as each carry output depends on the previous carry value according to the formula $C_{i+1} = G_i + P_i \cdot C_i$. The critical path begins at the initial carry-in (Cin), propagates through the carry logic to compute C1 and C2(Cout). Since sum outputs S0 and S1 depend on the availability of the corresponding carry values, the delay through the carry generation logic directly impacts the speed of the entire circuit.

In Task 3, the 4-bit CLA was constructed by cascading two 2-bit CLA blocks in a Ripple Carry Adder (RCA) format. As a result, the carry-out from the first 2-bit CLA becomes the carry-in for the second. This structure makes the delay of the overall circuit dependent on the sequential propagation of the carry signal through both CLA blocks. The critical path starts from the initial Cin, passes through the first CLA's carry logic to generate C2, then continues through the second CLA to produce the final Cout. This means the carry signal needs to go through two stages one

after another, which makes the delay longer than in the 2-bit CLA. Because of this, the carry path is the part of the circuit that causes the most delay in the 4-bit design.