Name: Saumya T Vira

Roll No: 2024102044

I. Introduction

A Carry Look-Ahead Adder (CLA) is an enhanced alternative to the conventional Ripple Carry Adder (RPA). In an RPA, each bit must wait for the carry to propagate from all preceding stages, creating significant delay—especially for larger bit-widths. To overcome this limitation, the 5-bit CLA design computes carry signals in parallel rather than allowing them to ripple sequentially.

In this architecture, the input operands are denoted as Ai and Bi for i = 0 to 4, with C0 as the initial carry input. The CLA minimizes delay by generating two key signals for each bit independently: the Propagate signal (Pi) and the Generate signal (Gi). These are defined as:

$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

Using these, each sum output (Si) is determined by:

$$S_i = P_i \oplus C_i$$

Positive edge-triggered flip-flops are incorporated to stabilize inputs and capture outputs. When the input bits are present before the rising clock edge, the corresponding outputs are computed and stored at the next positive edge.

II. CLA Concepts

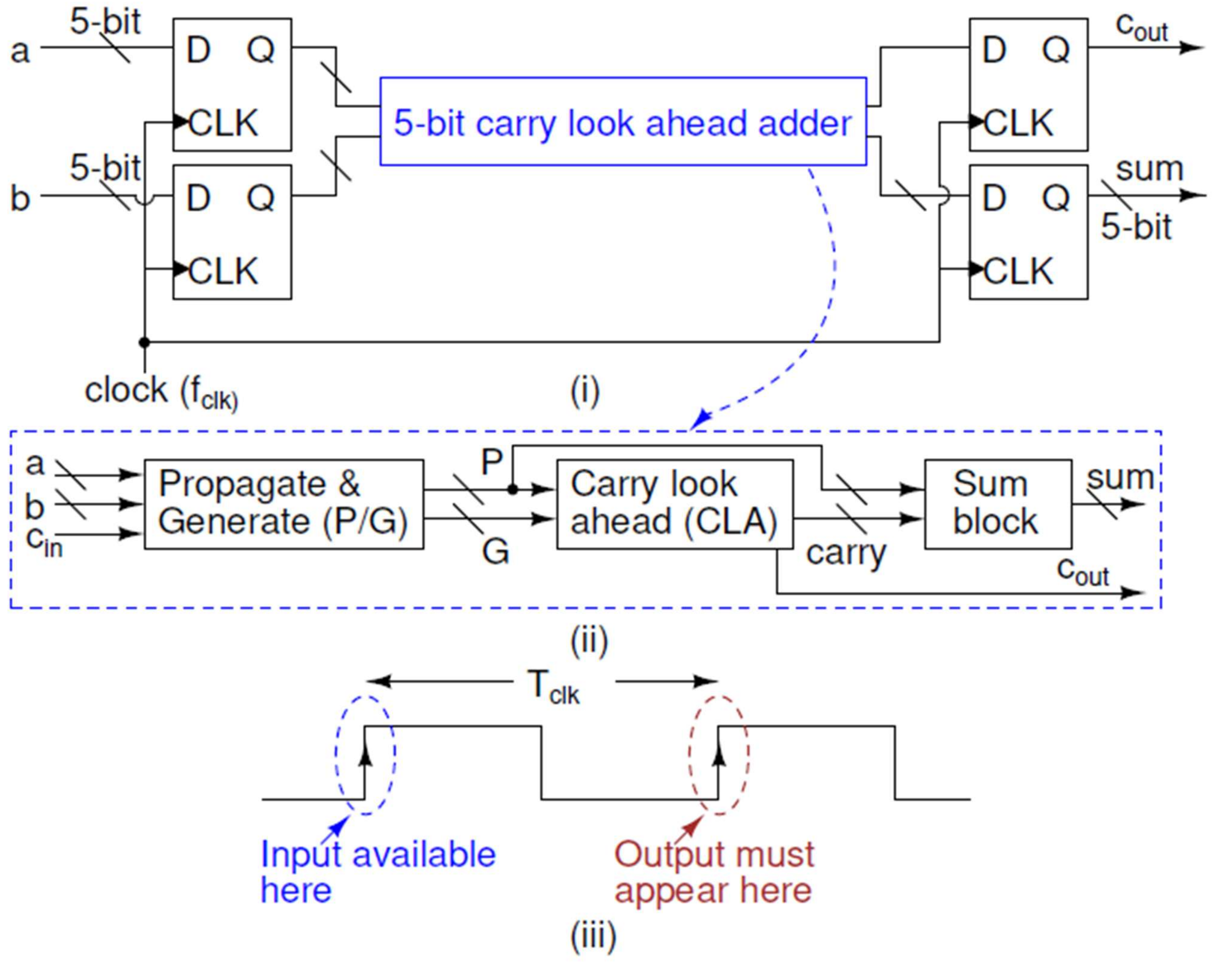
In a Carry Look-Ahead Adder, the carry for each subsequent stage (i + 1) is determined using the generate and propagate signals from the previous bit. This relationship is given by:

$C_{i+1} = G_i + (P_i \cdot C_i)$

For a 5-bit CLA, these expressions are fully expanded so that carries C1 through C5 are calculated simultaneously, eliminating the sequential delay found in ripple carry adders. The expanded Boolean expressions are:

$$
\begin{aligned}
C1 &= G0 + P0 \cdot C0 \\
C2 &= G1 + P1 \cdot G0 + P1 \cdot P0 \cdot C0 \\
C3 &= G2 + P2 \cdot G1 + P2 \cdot P1 \cdot G0 + P2 \cdot P1 \cdot P0 \cdot C0 \\
C4 &= G3 + P3 \cdot G2 + P3 \cdot P2 \cdot G1 + P3 \cdot P2 \cdot P1 \cdot G0 \\
&\quad + P3 \cdot P2 \cdot P1 \cdot P0 \cdot C0 \\
C5 &= G4 + P4 \cdot G3 + P4 \cdot P3 \cdot G2 + P4 \cdot P3 \cdot P2 \cdot G1 \\
&\quad + P4 \cdot P3 \cdot P2 \cdot P1 \cdot G0 + P4 \cdot P3 \cdot P2 \cdot P1 \cdot P0 \cdot C0
\end{aligned}
$$

As seen in the C5 expression, the logic becomes more complex as the bit-width increases. The final product term involves a 6-input AND gate, and combining all terms requires a 6-input OR gate, reflecting the rise in hardware complexity and fan-in.



III. Logic Gate Requirements

To implement the 5-bit Carry Look-Ahead (CLA) logic derived from the expanded carry equations, the following logic gates are required:

- **NOT gate**
- **XOR gate**: 2-input (used for Sum generation)
- **AND gates**: 2, 3, 4, 5, and 6 input versions
- **OR gates**: 2, 3, 4, 5, and 6 input versions

The need for 6-input AND and OR gates arises specifically from the computation of the **C5** carry term, which must account for the propagation chain across all five preceding stages (0 through 4).

All logic gates are implemented using **Complementary CMOS logic**.
The input and output D flip-flops are designed using **True-Single-Phase-Clock (TSPC)** topology.
The complete design follows a **3-stage structure**, where:

- Stage 1: TSPC flip-flop
- Stage 2: Combinational CLA block
- Stage 3: TSPC flip-flop
  This ensures signal stabilization across stages.

IV. Topology and Sizing

The circuit is implemented in **180 nm technology**, with a minimum feature size of **λ = 0.09 µm**.
Transistor sizing is selected to balance rise and fall times, compensating for the lower mobility of PMOS devices compared to NMOS devices.

IV.i Logic Gate Sizing

- **NOT Gate:** Uses standard sizing with
  - $WN=10\lambda$
  - $WP=20\lambda$
- **AND Gates:** Realized using a **NAND gate followed by an inverter**.
  To maintain consistent pull-down strength, the NMOS devices in the NAND stack are scaled by the number of inputs N.
  For example, a 6-input NAND uses:
  - $WN=6\times Wref$

- **OR Gates:** Implemented using a **NOR gate followed by an inverter**.
  To preserve pull-up strength, the PMOS devices in the NOR stack are scaled by N.
  For a 6-input NOR:
  - WP=6×Wref

IV.ii Flip-Flop Topology

The D flip-flops employ **TSPC dynamic logic**, chosen for its reduced transistor count and lower clock loading compared to traditional static CMOS flip-flops.

Timing Analysis Overview

The performance of the 5-bit Carry Look-Ahead Adder (CLA) is assessed using several key timing parameters:

- **Setup Time (Tsetup)**
  The minimum interval before the active clock edge during which the D input must stay stable.
  In simulation, this is found by gradually moving the D input transition closer to the rising clock edge until the flip-flop begins to latch incorrect data.

- **Hold Time (Thold)**
  The minimum duration after the active clock edge during which the input must remain unchanged.
  For the TSPC flip-flop used in this project, the theoretical hold time is nearly 0 ps, meaning the input may change immediately after the rising edge.

- **Clock-to-Q Delay (Tcq)**
  The time from the rising clock edge to when the Q output becomes valid.

- **Propagation Delay (Tpd)**
  The delay caused by the CLA's combinational logic.
  It represents the time required for the sum and carry signals to stabilize after input transitions.

The worst-case scenario occurs when the carry must propagate all the way from bit 0 to bit 5.

Procedure for Extracting Tsetup, Thold, and Tcq

A sweep-based (iterative) simulation method is used to derive the timing characteristics of the D flip-flop:

- **Setup Time Extraction**
  - The D input transition is moved gradually closer to the rising clock edge.
  - When the flip-flop begins producing incorrect outputs, it indicates the input changed too late.
  - The last timing point at which the output is still correct is taken as the setup time.
- **Hold Time Extraction**
  - The D input transition is shifted progressively after the rising clock edge.
  - Incorrect latching indicates that the input changed too early.
  - The latest timing point at which valid output is maintained gives the hold time.
- **Loop-Based Precision**
  - Both setup and hold times are determined by sweeping the input timing and identifying the exact point where proper operation fails.
  - This method ensures accurate boundary detection for valid operation.
  - The same simulation window also provides a precise measurement of Tcq, based on the delay between the clock edge and the point when Q stabilizes.

VI. Prelayout Analysis

Tpcq Delay (Clock to Q Delay)

Tpcq rise = 5.05848e-11 s   T pcq fall = 1.07135e-10s   Avg Tpcq = 78.75 ps

Tsetup = 47.4 ps   Thold = 0 ps

CLA Adder

Total CLA Delay = Tpcq + Tcomb,max + Tsetup + Tskew

Fmax = 1 / Tclk min = 1.736 GHZ

VII.i Magic Layout

- In the magic-layout implementation of a 5-bit Carry Lookahead Adder (CLA), the design is constructed from modular propagate and generate (P/G) cells that are reused across CLAs of different sizes (2, 3, 4, 5, and 6bit). Each bit-cell produces the propagate signal (Pi = Ai XOR Bi) and the generate signal (Gi = Ai.Bi). These signals are then combined to implement the standard carry-lookahead relation:

  $$C_{i+1} = G_i + P_i \cdot C_i,$$

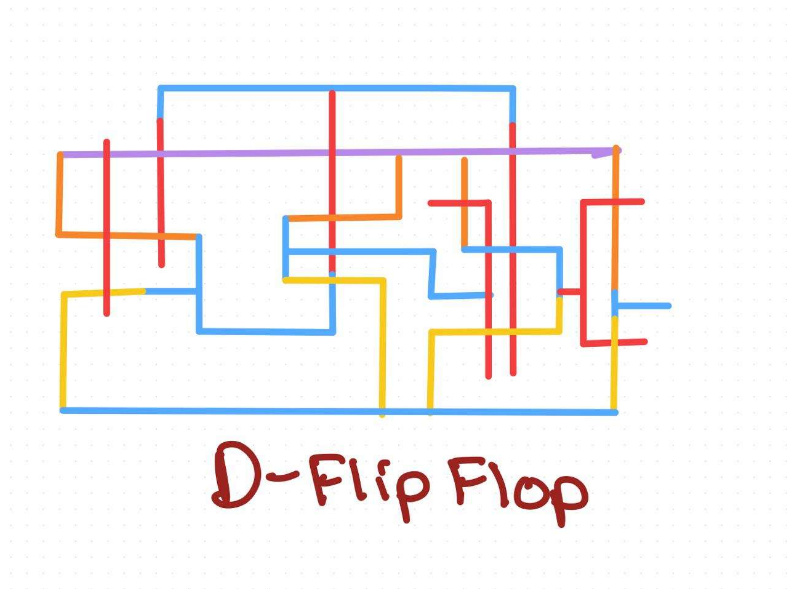
  with higher-order carries expanded hierarchically—for example,

  C2 = G1 + P1.G0 + P1.P0.C0.

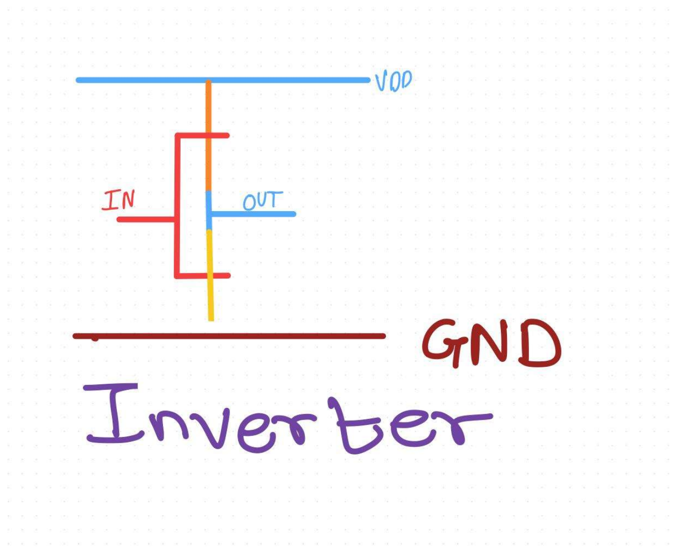
- Because the base P/G unit is verified only once, the same cell layout can be repeatedly used and tiled to assemble larger CLAs without redesigning individual logic blocks. In the 5-bit CLA, five identical P/G cells are arranged in sequence, with precise alignment of diffusion areas and metal routing to reduce parasitic capacitance and interconnect delay. This modular structure ensures scalability and uniformity: the same propagate and generate templates are consistently reused for different bit-width adders, producing an optimized and reliable carry path for implementation.

Stick Diagrams

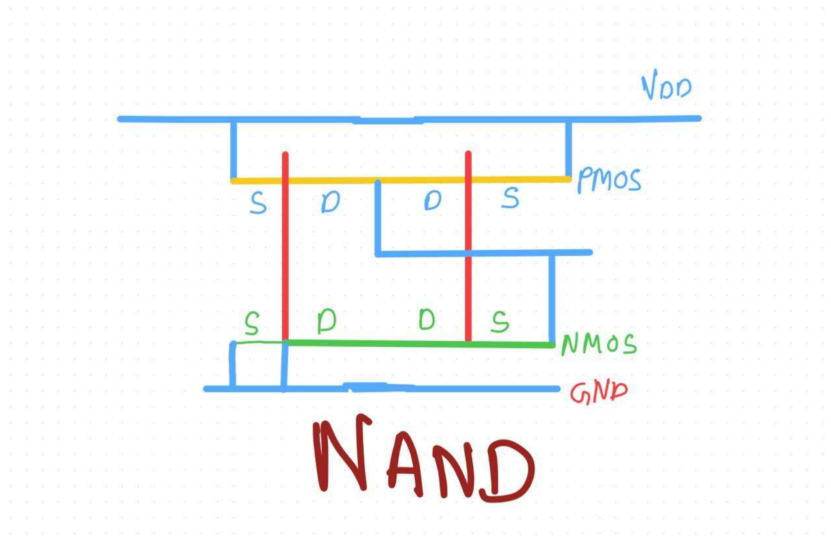
1. D Flip Flop



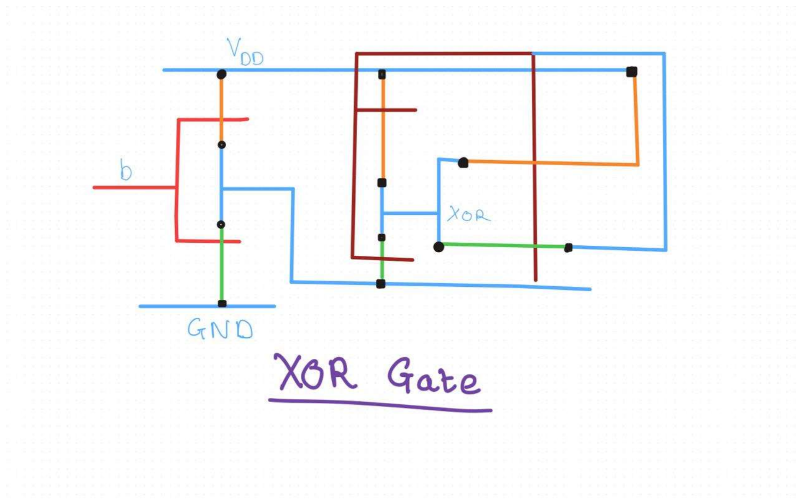
2. Inverter



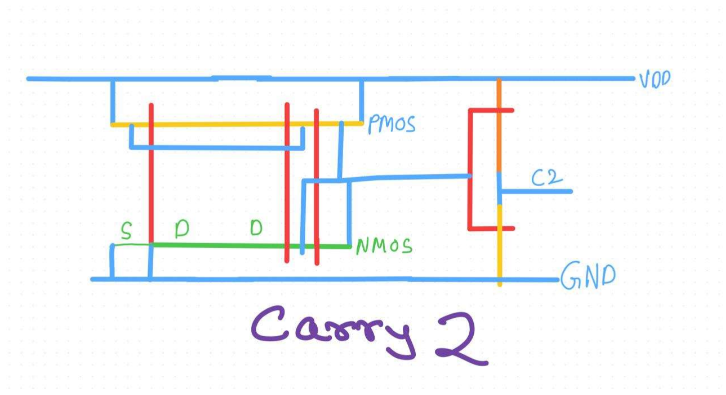
3. NAND



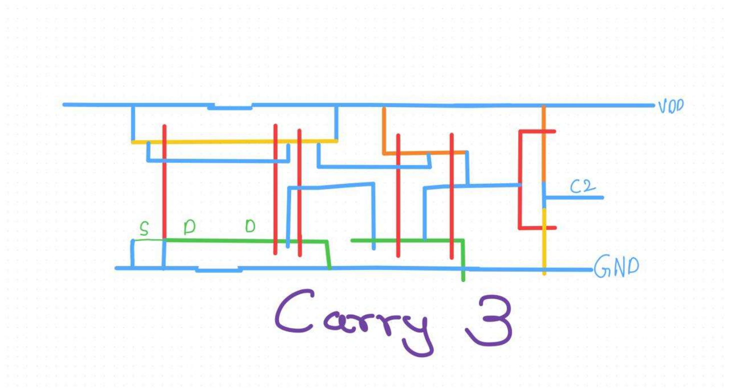
4. XOR



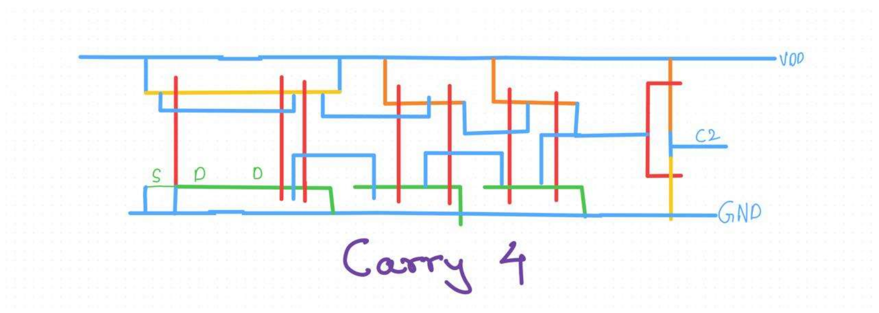
5. CARRRY 2



6. CARRY 3



7. CARRY 4



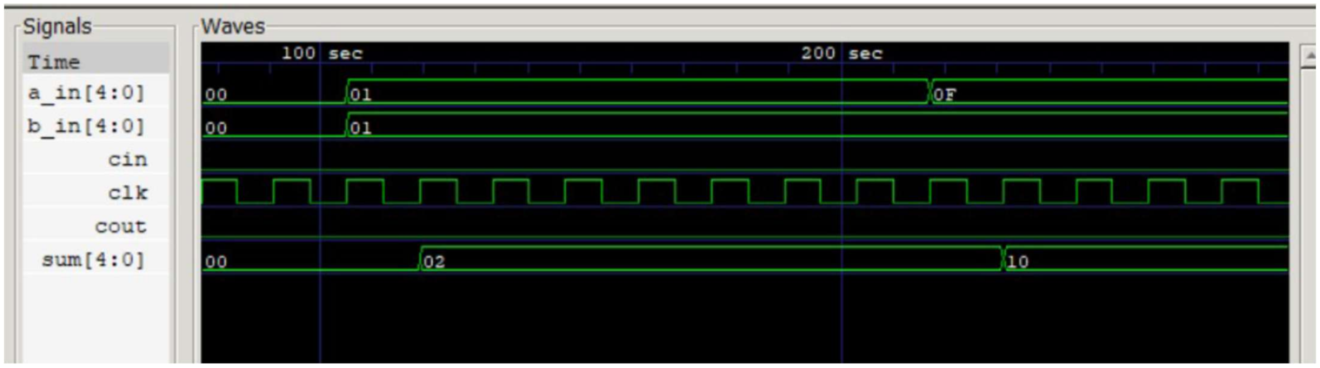
Hold Time of the TSPC Flip-Flop

In a TSPC (True Single Phase Clock) flip-flop, the effective hold time is theoretically close to 0 ps. This is because of its dynamic storage behavior and the way the clock isolates the input. When the clock rises, the transistor responsible for sampling the D input turns off almost instantly, cutting off the internal storage node from the input. Once this isolation occurs, any change at the D input after the clock edge can no longer affect the stored value.

Therefore, the input does not need to stay stable after the active clock edge, giving a theoretical **Thold ≈ 0 ps**.

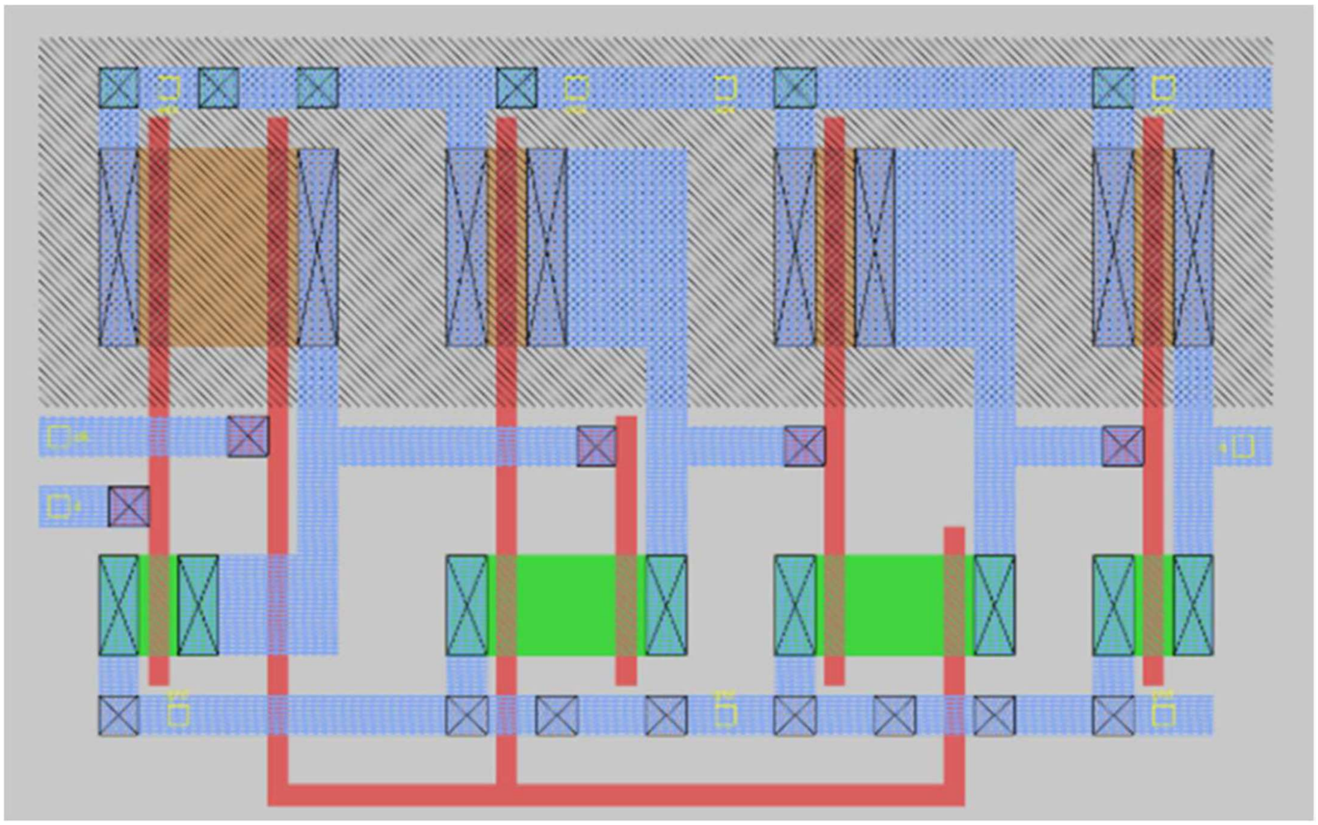
In actual hardware, small non-ideal effects—such as finite switching delays, charge redistribution, and clock skew—may introduce a very small hold time, but it usually remains negligible.

8. Verilog

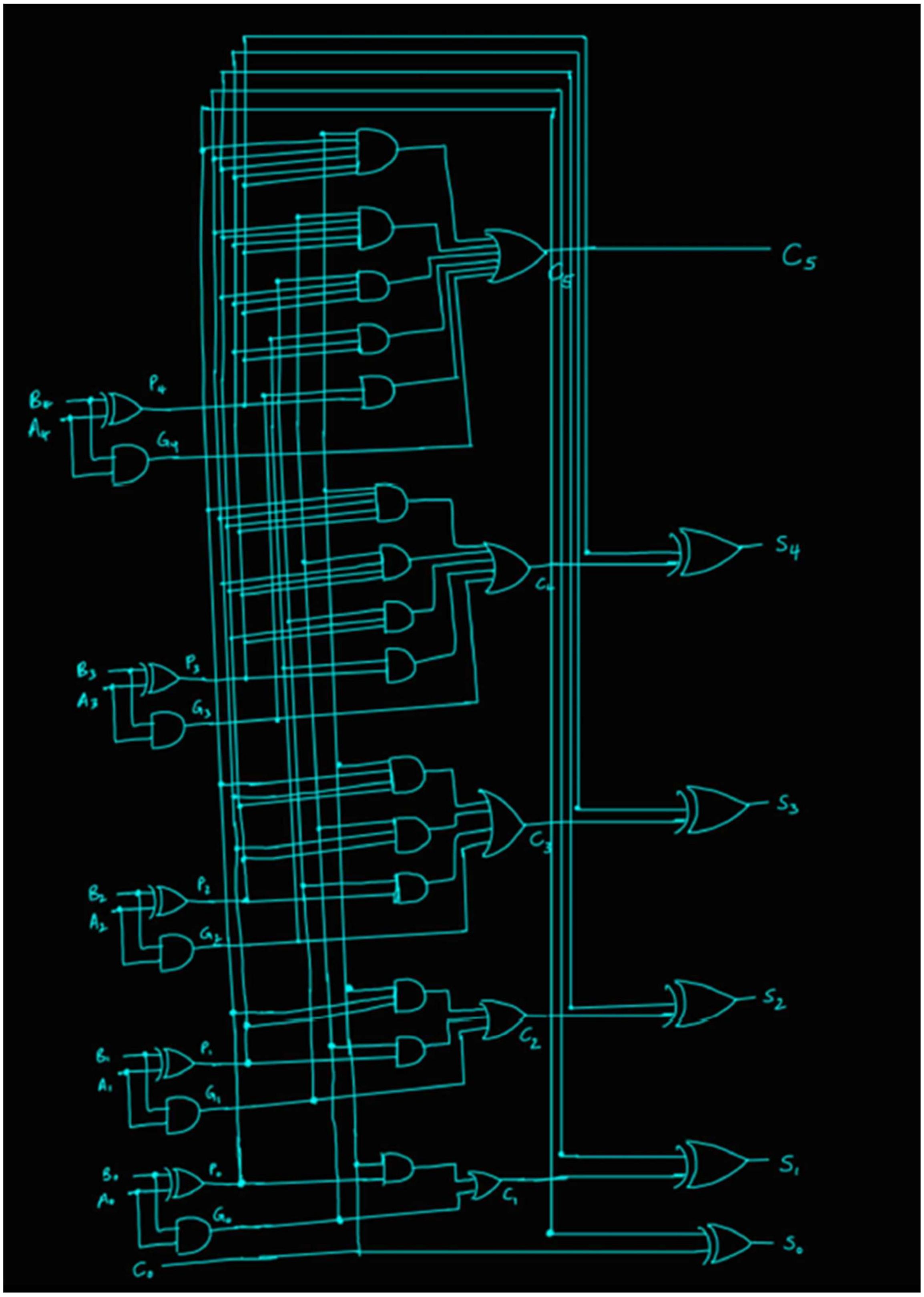


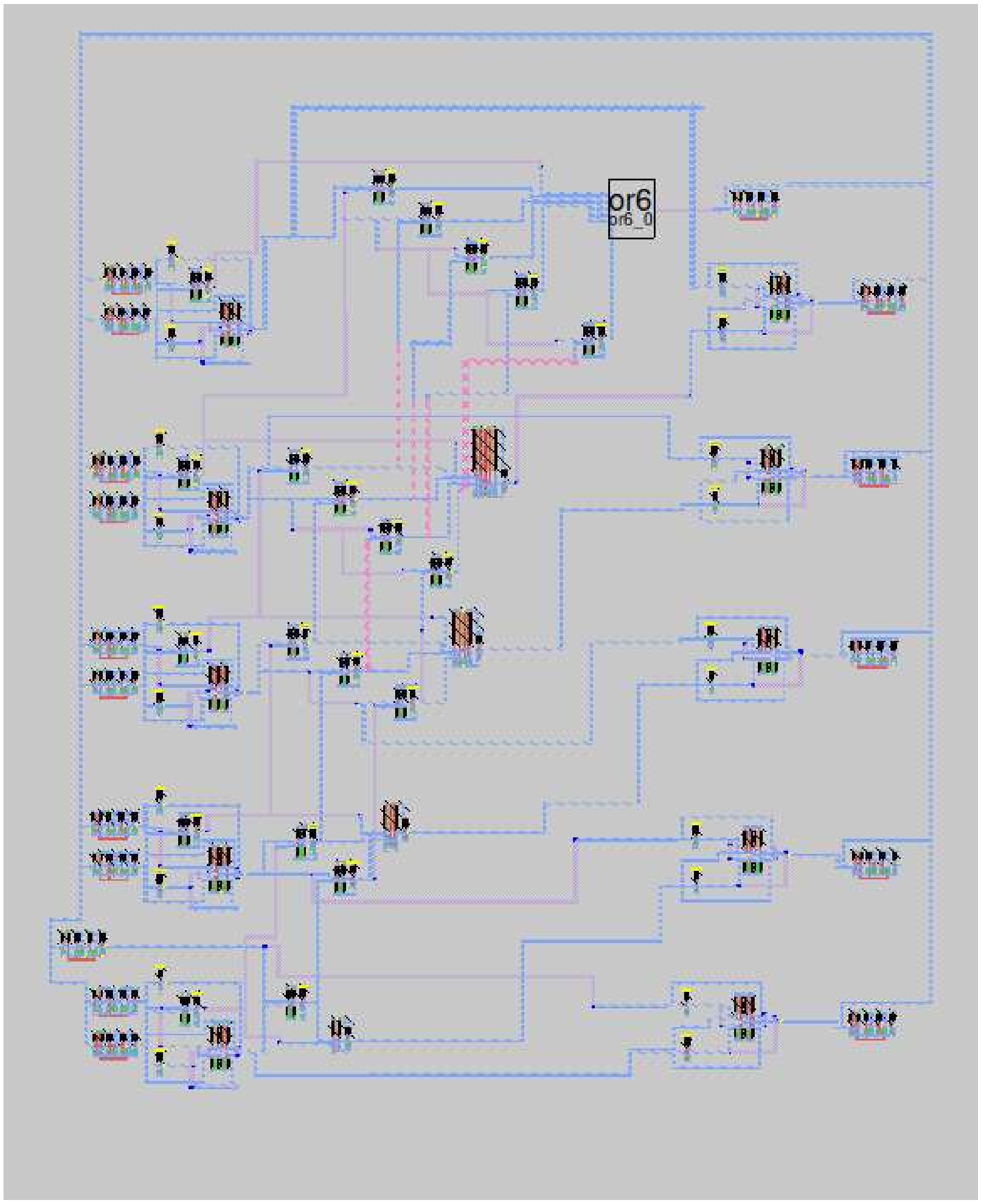
MAGIC

D FLIP FLOP



5 Bit CLA Adder

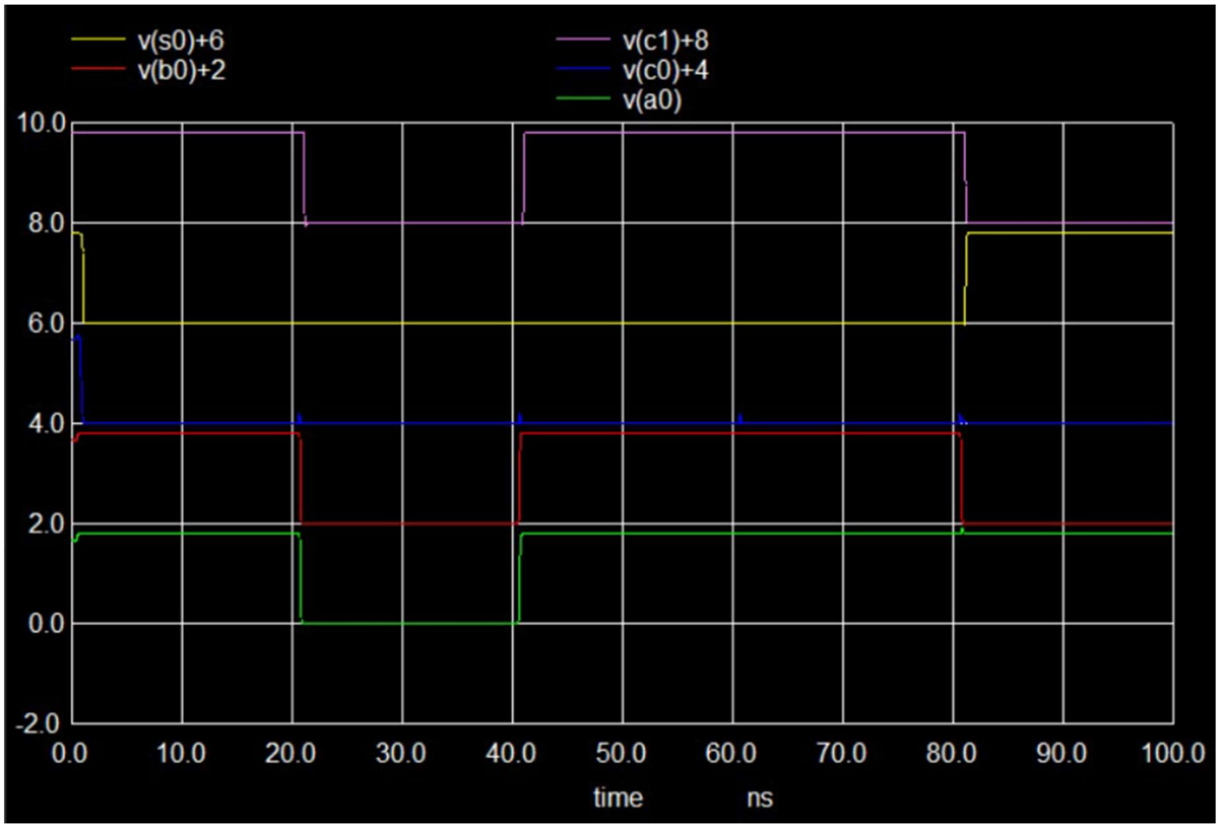


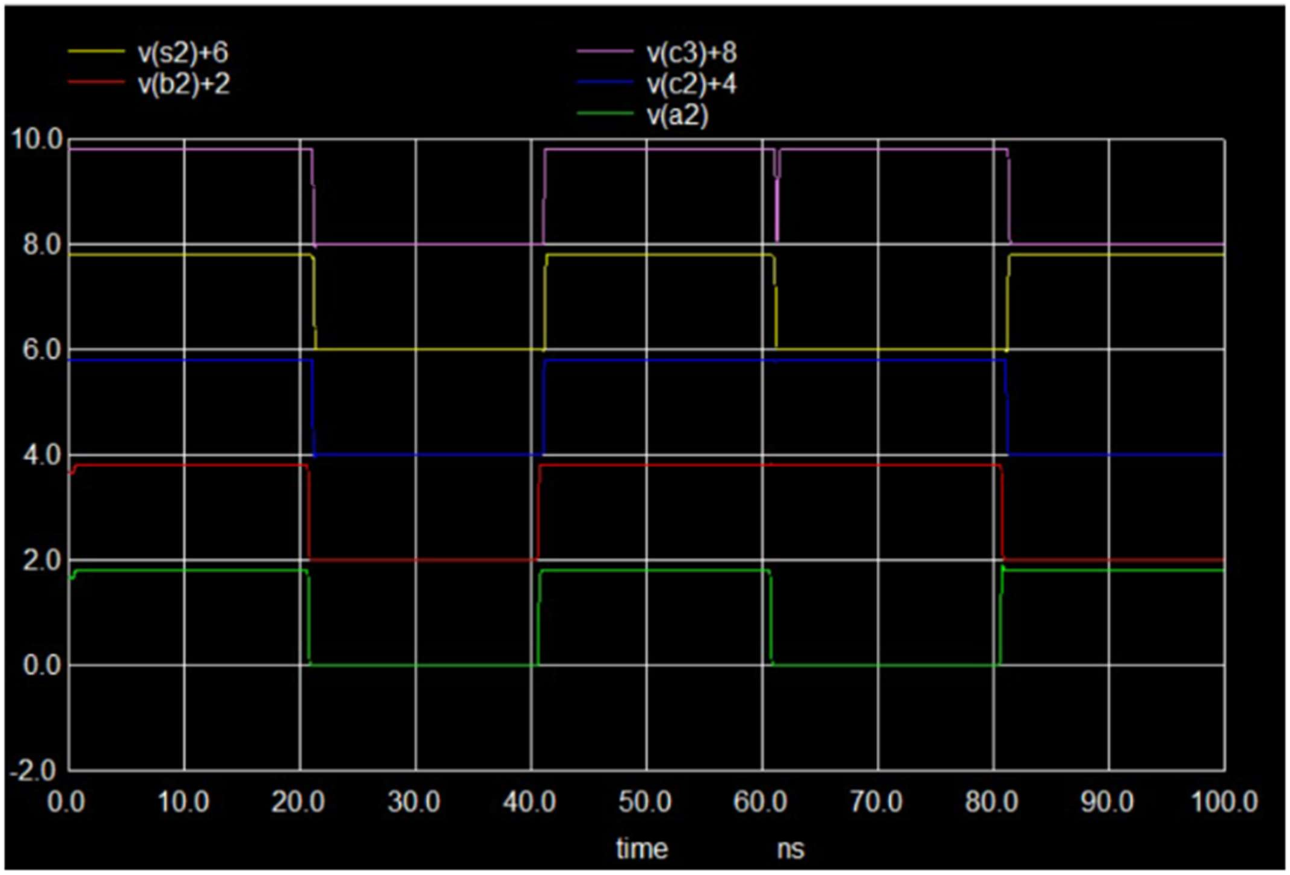


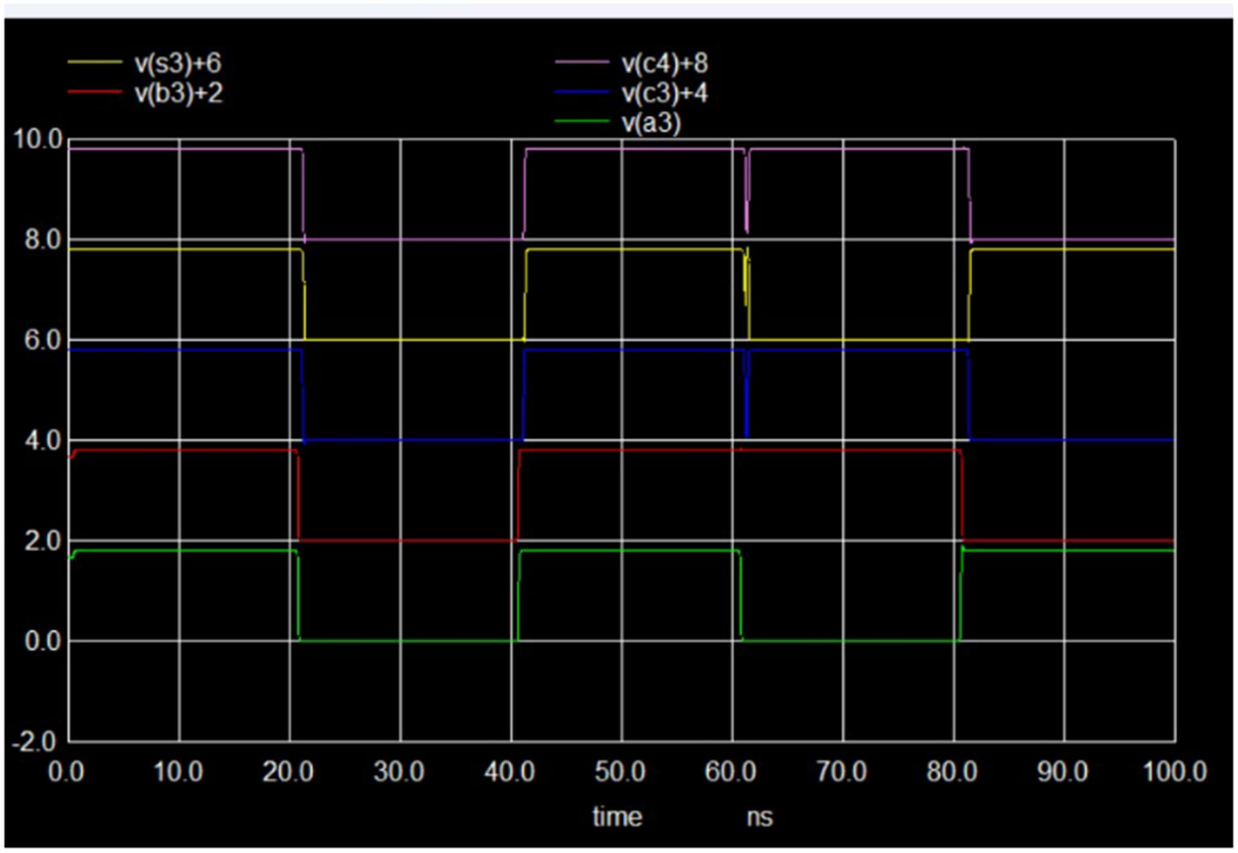
The size of the final circuit comes out to be 103.5 µm × 63.18 µm.

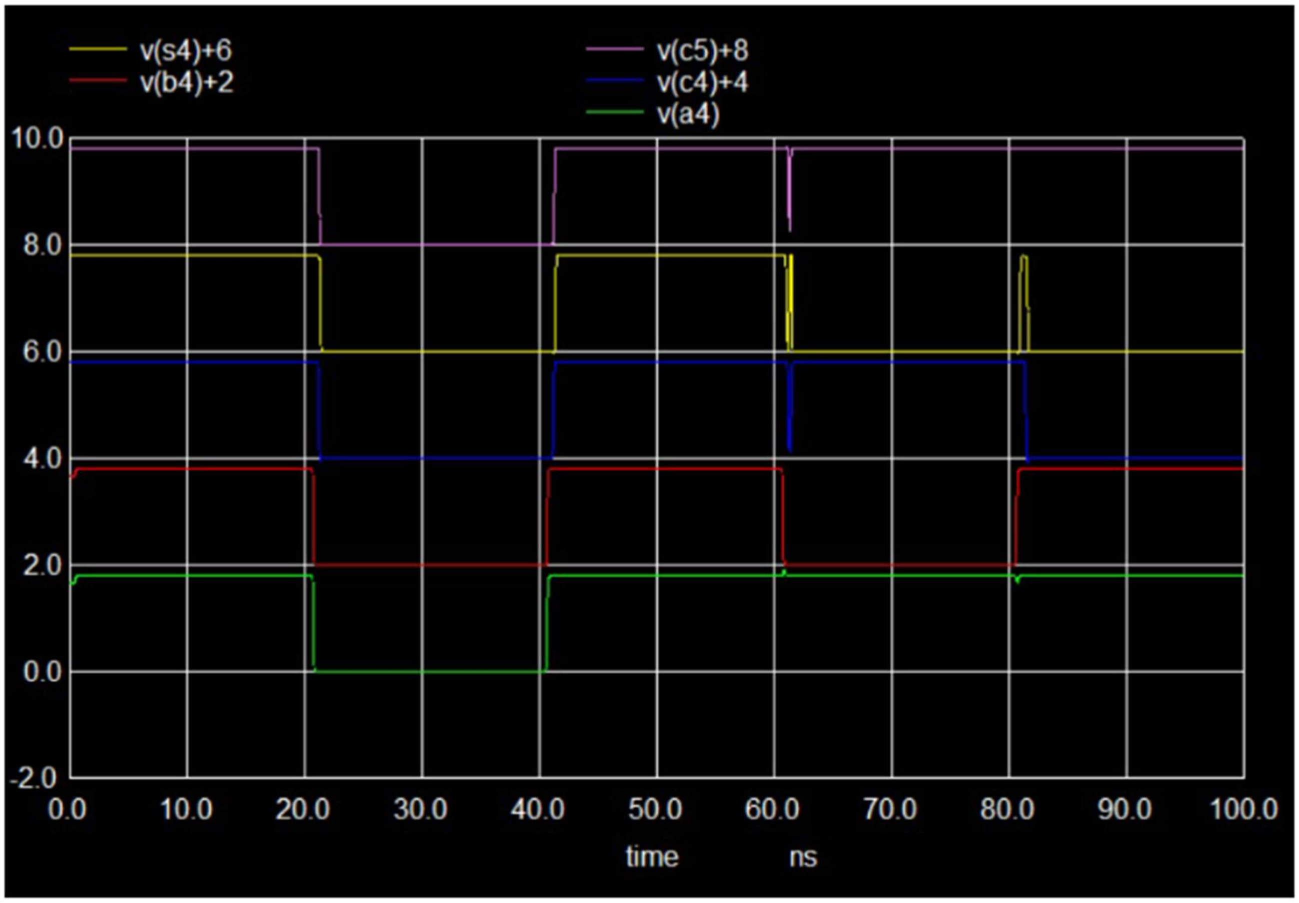
NGSPICE

Adder Ckt

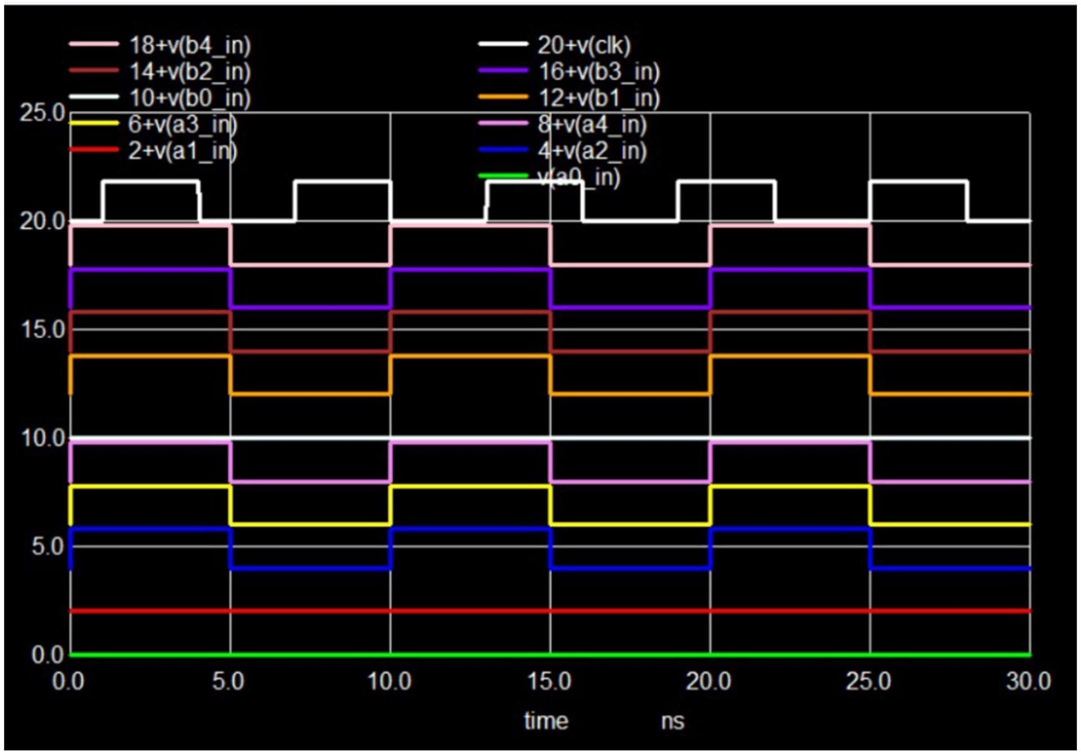


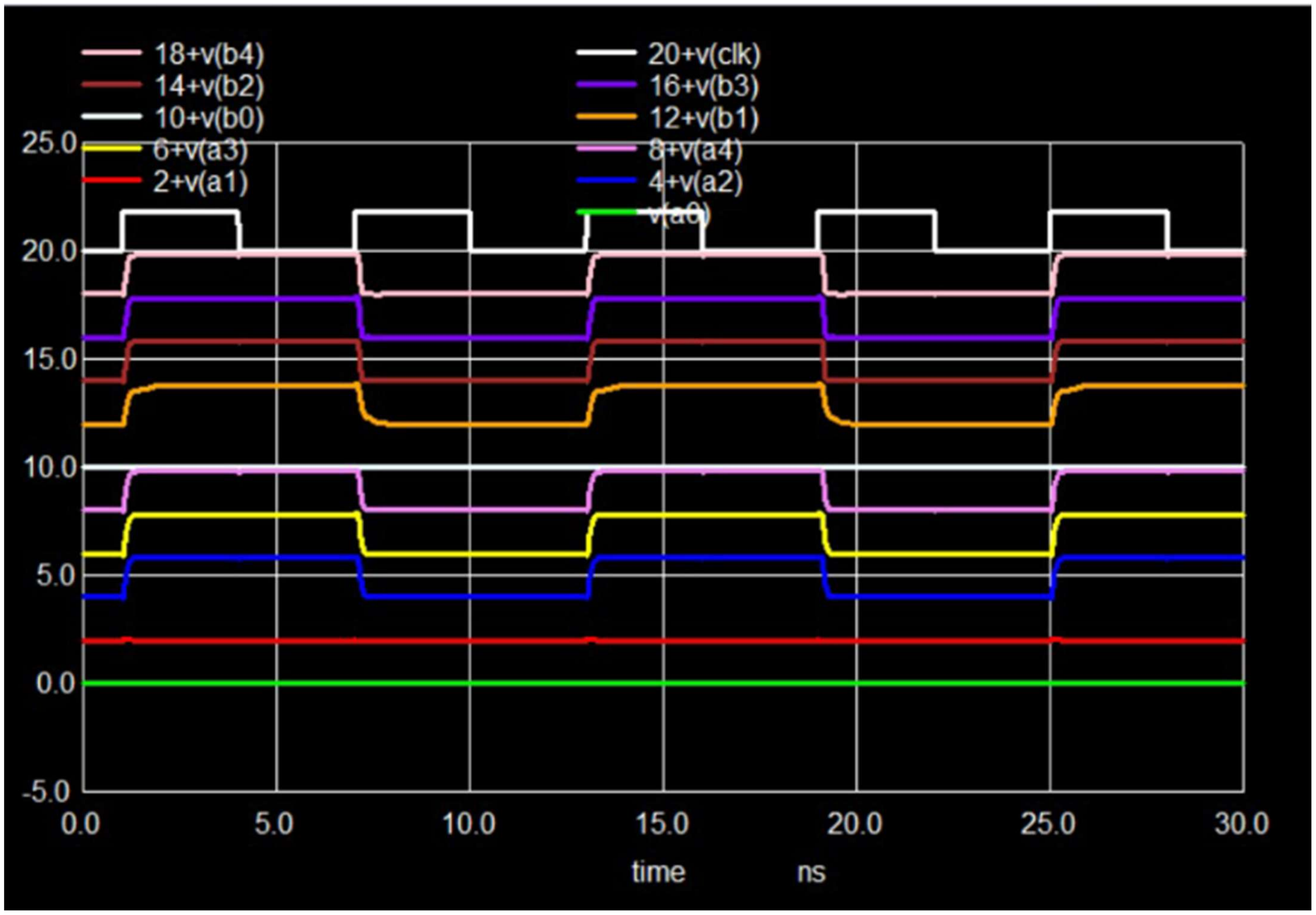


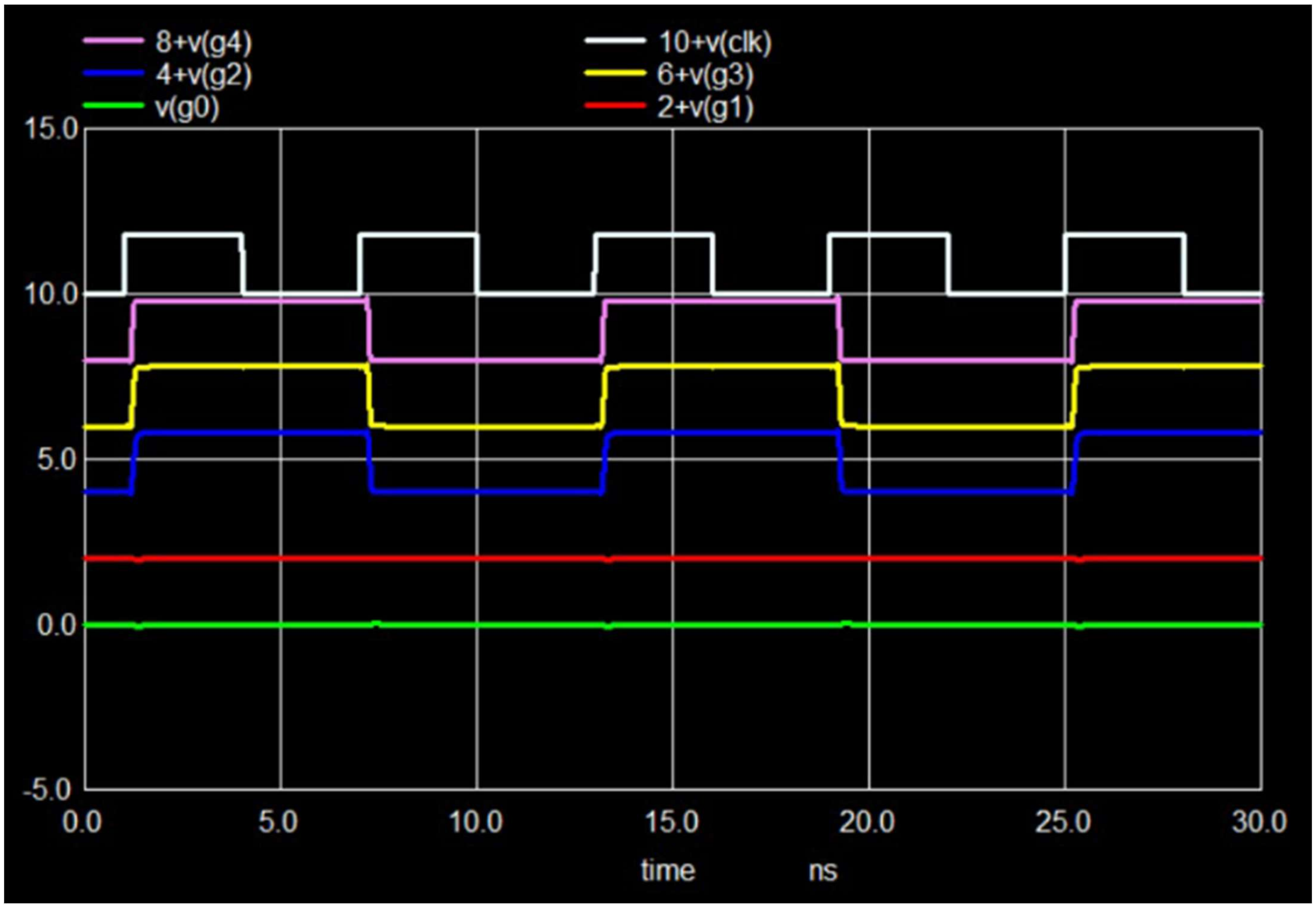

v(s3)+6
v(b3)+2
v(c4)+8
v(c3)+4
v(a3)
10.0
8.0
6.0
4.0
2.0
0.0
-2.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
time
ns

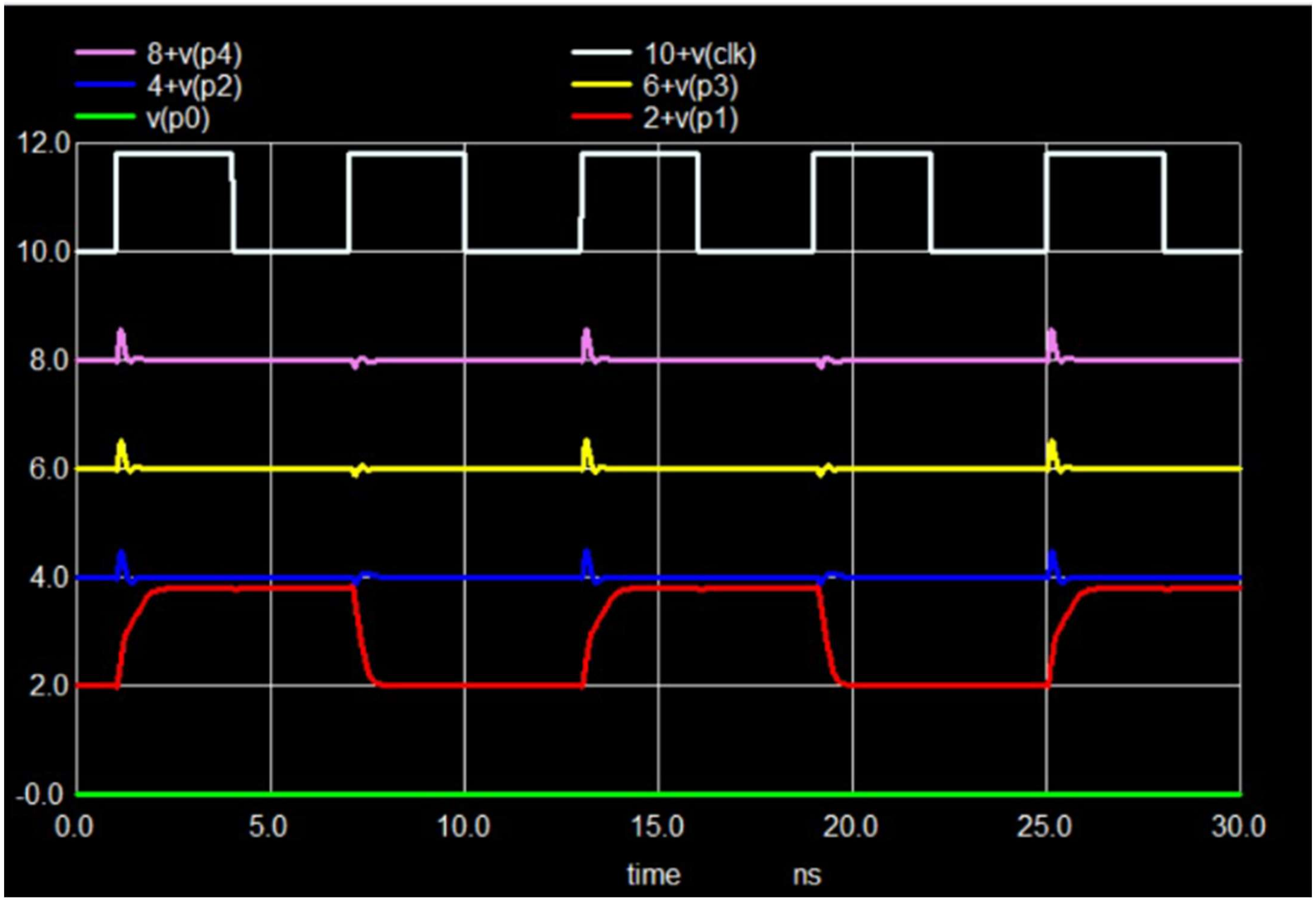

v(s4)+6
v(b4)+2
v(c5)+8
v(c4)+4
v(a4)
10.0
8.0
6.0
4.0
2.0
0.0
-2.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
time
ns

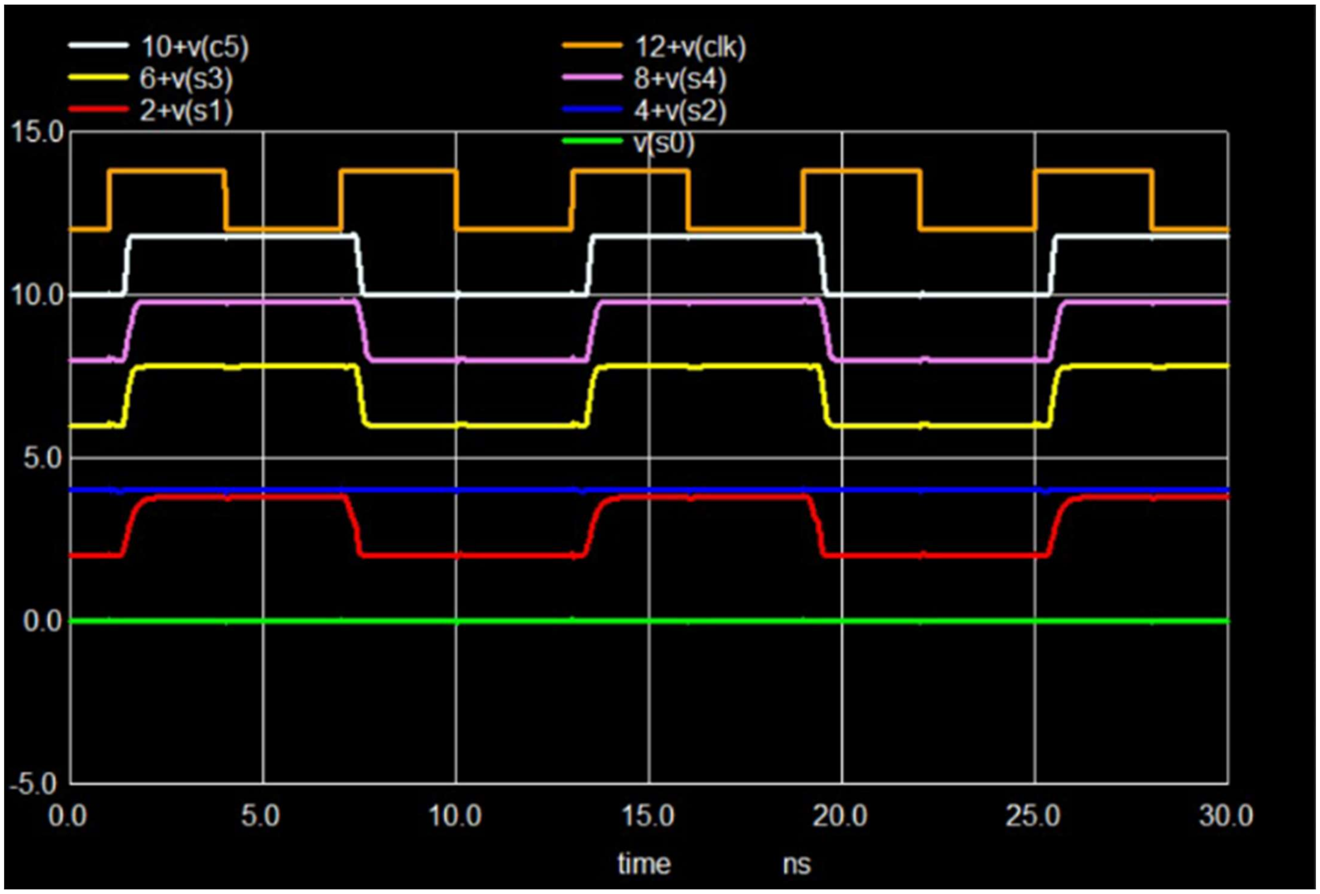
CLA ADDER


18+v(b4_in)
14+v(b2_in)
10+v(b0_in)
6+v(a3_in)
2+v(a1_in)
20+v(clk)
16+v(b3_in)
12+v(b1_in)
8+v(a4_in)
4+v(a2_in)
v(a0_in)
25.0
20.0
15.0
10.0
5.0
0.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
time
ns


18+v(b4)
14+v(b2)
10+v(b0)
6+v(a3)
2+v(a1)
20+v(clk)
16+v(b3)
12+v(b1)
8+v(a4)
4+v(a2)
v(a0)
25.0
20.0
15.0
10.0
5.0
0.0
-5.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
time
ns

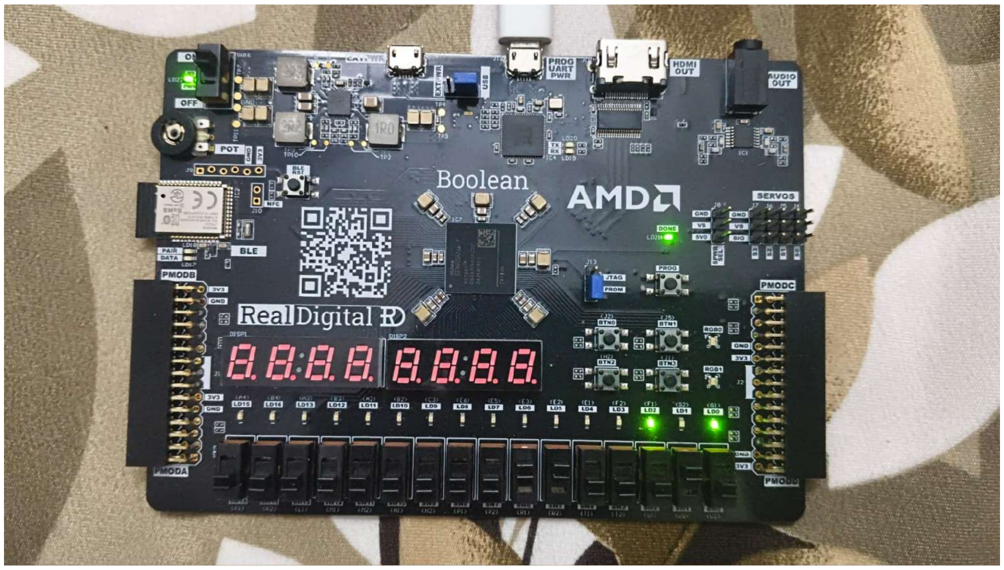

8+v(g4)
10+v(clk)
4+v(g2)
6+v(g3)
v(g0)
2+v(g1)
15.0
10.0
5.0
0.0
-5.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
time
ns


8+v(p4)
10+v(clk)
4+v(p2)
6+v(p3)
v(p0)
2+v(p1)
12.0
10.0
8.0
6.0
4.0
2.0
-0.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
time
ns

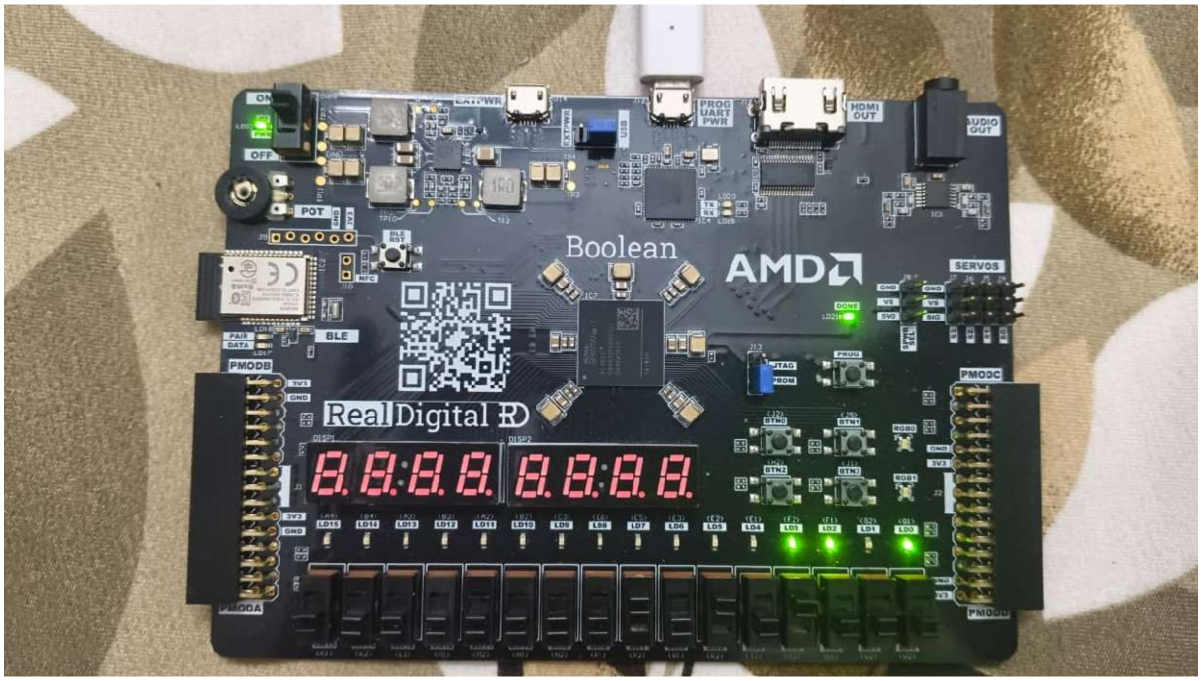


10+v(c5)
12+v(clk)
6+v(s3)
8+v(s4)
2+v(s1)
4+v(s2)
v(s0)
15.0
10.0
5.0
0.0
-5.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
time
ns

FPGA Implementation



A=10   B=11 C0=0 S=101 (Verified)



A = 1000 B =101 C0=0 S= 1011

The implemented Verilog design was programmed onto the AMD Boolean FPGA development board shown . The board provides onboard switches, push buttons, LEDs, and seven-segment displays that were used to apply the input vectors A and B, as well as the carry input Cin, and to observe the resulting sum outputs and carry-out signals in real time. This physical implementation verifies that the functional behavior observed during simulation is preserved in actual hardware operation.

Conclusion

This project demonstrates the successful design and verification of a 5-bit Carry Look-Ahead (CLA) adder using CMOS logic and clocked D flip-flops. The complete system was first verified at the transistor level through pre-layout NGSPICE simulations and then validated at the behavioral level using Verilog simulations and FPGA implementation. The inclusion of D flip-flops at both the input and output stages ensures that the data is captured on a positive edge of the clock and the final results are made available on the following clock edge. This timing arrangement provides sufficient time for the CLA block to compute the sum and carry outputs accurately while maintaining proper synchronization across all stages of the circuit. The measured timing parameters obtained from NGSPICE (setup time, hold time, and clock-to-Q propagation delay) confirm correct flip-flop operation. Further verification using Verilog simulation and FPGA hardware testing showed complete agreement between theoretical, simulated, and experimental results, validating the correctness of the overall design.