5-Bit Carry Look-Ahead Adder: Design, Implementation and Analysis

Siddhant Gudwani
Electronics and Communication Engineering
IIIT Hyderabad
Hyderabad, India
siddhant.gudwani@students.iiit.ac.in

***Abstract*—Addition is a fundamental operation in digital systems, significantly impacting the performance of arithmetic and logic units (ALUs), processors, and other digital circuits. The Carry Look-Ahead Adder (CLA) addresses the high propagation delay suffered by traditional ripple carry adders by employing a parallel approach to carry generation. This paper presents the complete design, implementation, and analysis of a 5-bit CLA adder using 180nm CMOS technology, utilizing D flip-flops for synchronous operation.**

***Index Terms*—Carry Look-Ahead Adder, 5-bit Adder, TSPC D Flip-Flop, CMOS VLSI, Timing Analysis**

I. Introduction

II. Proposed Structure and Design Details

A. Proposed Structure for the Adder

The 5-bit CLA adder consists of three main blocks as shown in Fig. 1:

- Propagate and Generate (P/G) Block
- Carry Look-Ahead Logic (CLA) Block
- Sum Generation Block

Inputs are captured at one clock edge and outputs appear at the **next clock edge**.

For two 5-bit numbers $a_5a_4a_3a_2a_1$ and $b_5b_4b_3b_2b_1$, the propagate (p_i) and generate (g_i) signals for each bit position are defined as (for $i = 1, 2, 3, 4, 5$):

Propagate and Generate Signals: For each bit position, the propagate and generate signals are defined as:

$$p_i = a_i \oplus b_i, \quad g_i = a_i \cdot b_i$$

Using these definitions, the carry-out for each bit can be written as:

$$c_{i+1} = g_i + (p_i \cdot c_i)$$

with the initial carry-in set to $c_0 = 0$.

Applying this relationship iteratively, the carry signals for all bit positions become:

$c_1 = g_0$

$c_2 = g_1 + (p_1 \cdot g_0) + (p_1 \cdot c_1)$

$c_2 = g_1 + (p_1 \cdot c_1) = g_1 + (p_1 \cdot g_0)$

$c_3 = g_2 + (p_2 \cdot c_2) = g_2 + (p_2 \cdot g_1) + (p_2p_1 \cdot g_0)$

$c_4 = g_3 + (p_3 \cdot c_3) = g_3 + (p_3 \cdot g_2) + (p_3p_2 \cdot g_1) + (p_3p_2p_1 \cdot g_0)$

$c_5 = g_4 + (p_4 \cdot c_4) = g_4 + (p_4 \cdot g_3) + (p_4p_3 \cdot g_2) + (p_4p_3p_2 \cdot g_1) + (p_4p_3p_2p_1 \cdot g_0)$

(Note: The original image had c_2 defined incorrectly. The correct expansion is shown above based on the logic $c_{i+1} = g_i + p_ic_i$.)

Finally, each sum bit is computed as:

$$S_i = p_i \oplus c_i$$

III. III. Design Details

A. A. Static CMOS Realization Using NAND Universal Gates

The complete 5-bit CLA architecture in this project is implemented using **static CMOS logic**, specifically utilizing **NAND-gate-based universal logic**. NAND is preferred in CMOS design because it provides:

- high noise margins,
- low static power consumption,
- highly regular layout structure,
- minimum number of transistors for universal logic implementation.

All major combinational blocks—propagate/generate (P/G) logic, carry look-ahead logic, and the sum generation logic—are built from NAND networks. This provides better transistor count efficiency, ease of layout design, and predictable timing behavior.

Since the design is fully static CMOS, every gate possesses strong pull-up and pull-down networks, ensuring full logic swings, glitch-free operation, and robust behavior across process, voltage, and temperature variations. Thereby, this logic proves to be more robust than Pass Transistor Logic.

The proposed 5-bit CLA adder uses a CMOS-based implementation with the following elements:

1) **Inverter (INV):** The inverter is the fundamental building block with:
   - NMOS: $W_N = 1.8\mu$m, $L = 2\lambda = 0.18\mu$m
   - PMOS: $W_P = 3.6\mu$m, $L = 2\lambda = 0.18\mu$m
2) **2-Input NAND Gate:** The 2-input NAND gate uses:
   - Series NMOS: $W = 2 \times W_N = 3.6\mu$m
   - Parallel PMOS: $W = W_P = 3.6\mu$m

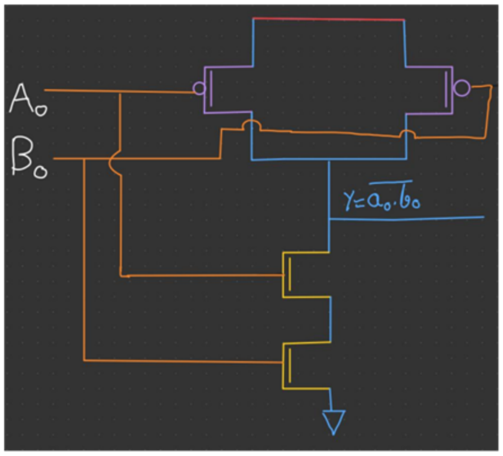


Fig. 1: Fig. 3: NAND Gate

3) **3-Input, 4-Input, and 5-Input NAND Gates:** For n-input NAND gates:
   - Series NMOS: $W = n \times W_N$ (compensate for series resistance)
   - Parallel PMOS: $W = W_P$ (no series connection)
4) **2-Input XOR Gate:** Implemented using 4 NAND gates in NAND-NAND configuration:

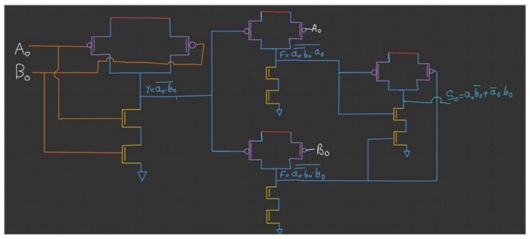


Fig. 2: Fig. 4: XOR using NAND gates

B. B. Propagate and Generate (P/G) Block

The P/G block generates the propagate and generate signals for each bit position:

- Propagate: XOR gate (4 NAND gates)
- Generate: NAND gate followed by inverter (active-low output $\overline{g_n}$)

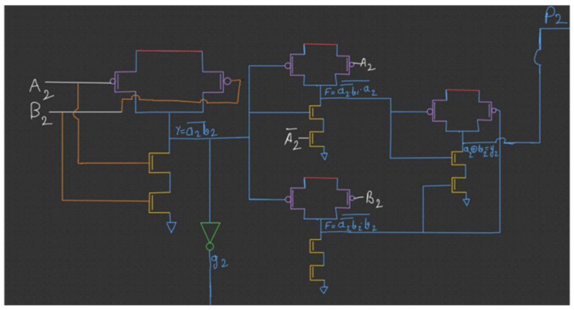


Fig. 3: P/G Block

C. C. Carry Look-Ahead Block

The carry logic block implements equations (4)-(8) using:

- 2-input NAND gates for two-term products
- 3-input NAND gates for three-term products
- 4-input NAND gates for four-term products
- 5-input NAND gates for five-term products
- Multi-input NAND gates for final OR operations (De Morgan's theorem)

D. D. Sum Generation Block

Sum bits are generated using XOR gates:

$$\text{sum}_i = p_i \oplus c_{i-1}$$

where c_0 is connected to ground (0).

E. E. TSPC Master-Slave D Flip-Flops for Input and Output Registers

To ensure that the 5-bit CLA operates synchronously with the system clock, **True Single-Phase Clock (TSPC) master-slave D flip-flops** are used at both the input and output of the adder. These flip-flops serve two essential roles:

1) **Input Registering:** The input operands A and B are sampled at the rising edge of the clock, ensuring that the combinational CLA logic receives stable data throughout the evaluation interval.
2) **Output Registering:** The computed sum and carry-out are stored in output flip-flops and become available at the next rising edge, ensuring predictable timing and clean interfacing with downstream logic.

A True Single-Phase Clock (TSPC) D flip-flop is used for input and output registers. The TSPC topology offers:

- Single clock signal (no complementary clock needed)
- Low power consumption
- Reduced clock skew
- Fast operation

F. Design Details (Topology and Sizing) of Each Block (Q2)

The core combinational logic is implemented using **static CMOS logic** within the 180nm technology node.

IV. Block-Level Simulation and Verification (Q3)

A. Complete 5-Bit CLA Adder Verification

The combinational CLA adder was verified with multiple test cases.

Crucially, the adder circuit was designed and partitioned into three core blocks (Propagate/Generate, Carry Look-Ahead, and Sum Block). Simulation confirmed that when these three blocks were integrated, the resulting 5-bit CLA adder circuit worked perfectly for all test cases, including the worst-case scenario.

Test Case: 10011 + 10001 = 100100

- Expected: $s_4s_3s_2s_1s_0 = 00100$, $c_{out} = 1$.

Simulation Results:

TABLE I: Complete Adder Test Results

Output	Expected	Measured	Status
s_0	0	0.02V	✓
s_1	0	0.03V	✓
s_2	1	1.76V	✓
s_3	0	0.02V	✓
s_4	0	0.03V	✓
c_{out}	1	1.74V	✓

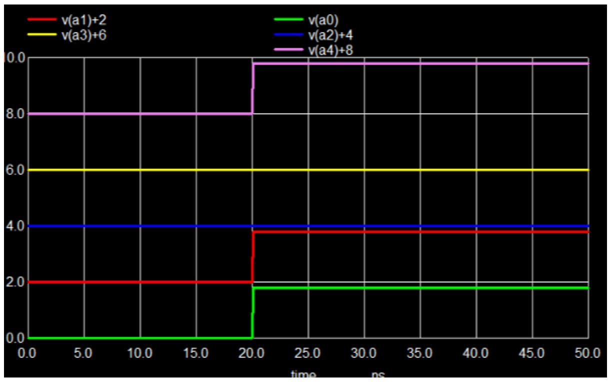


Fig. 4: Input bits of a

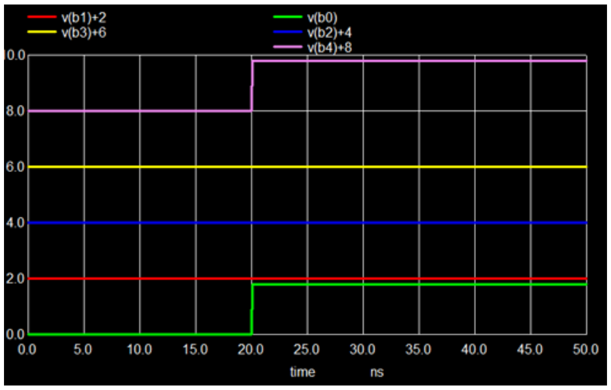


Fig. 5: Input bits of b

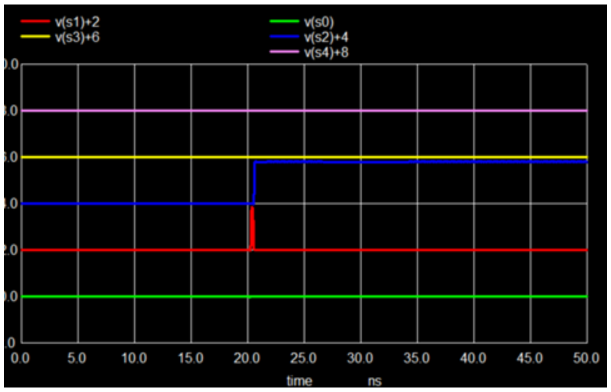


Fig. 6: output sum

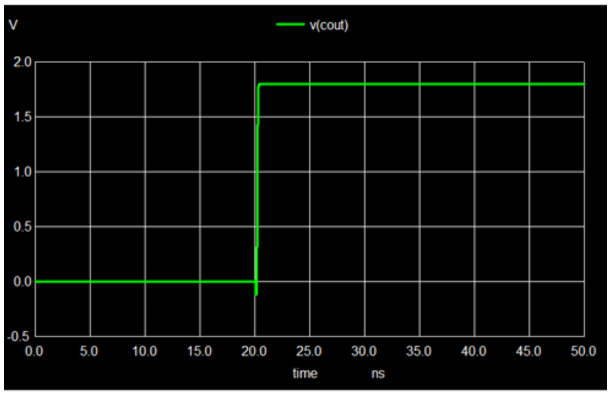


Fig. 7: Cout

All test cases passed successfully, confirming correct functionality of the complete 5-bit CLA adder.

B. D Flip-Flop (TSPC) Functional Verification

The TSPC DFF was functionally verified to capture the input (D) on the rising clock edge.

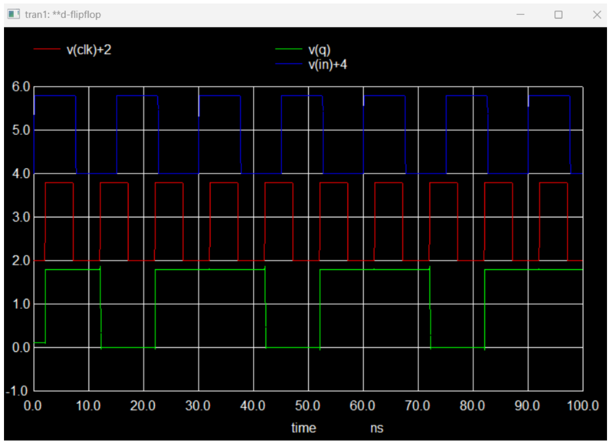


Fig. 8: D flip-flop functional waveform: CLK, D input, Q output

V. Calculation of Setup Time, Hold Time and Clock to Q Delay (Q4)

A. Setup Time (t_{su}) Measurement

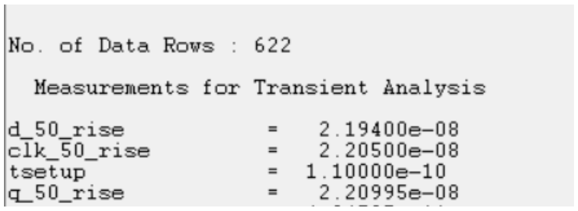
A. Setup Time Setup time is the minimum amount of time that the data input (D) must be stable and valid before the arrival of the clock edge to ensure proper operation of the flip-flop. If the data changes within this time window, it may lead to incorrect behavior or undefined states. In a D flip-flop, the setup time constraint is critical to avoid metastability. It is a key parameter in timing analysis and affects the overall performance of synchronous digital systems.

Rising Edge of Clock Comes at 5ns and our input is stabilised at 4.89ns.

SetupTime = 5ns 4.89ns = 0.11ns

Measured Setup Time:

- **Setup time:** $t_{su} = 110$ **ps**.



```
No. of Data Rows : 622

  Measurements for Transient Analysis

d_50_rise              =   2.19400e-08
clk_50_rise            =   2.20500e-08
tsetup                 =  1.10000e-10
q_50_rise              =   2.20995e-08
```

Fig. 9: Setup time measurement (110ps): Data must be stable before clock edge.

B. Hold Time (t_h) Measurement

Hold time is the minimum time the data must remain stable after the clock edge. The TSPC topology inherently has a zero hold time.

Hold Time:

- **Hold time:** $t_h = 0$ **ps**.

C. Clock-to-Q Delay (t_{c2q}) Measurement

Clock-to-Q delay (t_{c2q}) is the time for the output to change after a clock edge. We measure the delay from the clock reaching $V_{DD}/2$ (0.9V) to the output (Q) reaching $V_{DD}/2$.

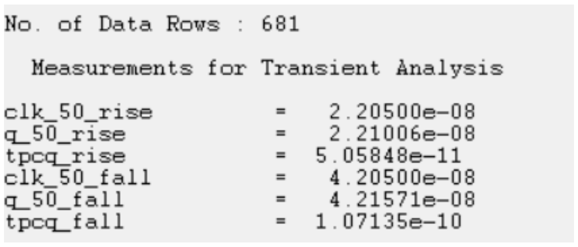
Measured Values from Transient Analysis:

- $t_{c2q,rising} = 50.58$ ps (5.05848e-11 s)
- $t_{c2q,falling} = 107.14$ ps (1.07135e-10 s)

Clock-to-Q Delay Summary:

TABLE II: D Flip-Flop Clock-to-Q Delay Characterization

Transition Type	Delay (ps)
Q: 0 $\rightarrow$ 1 (rising)	**50.58**
Q: 1 $\rightarrow$ 0 (falling)	107.14
Maximum (worst-case)	**107.14**
Average	78.86



```
No. of Data Rows : 681

  Measurements for Transient Analysis

clk_50_rise            =   2.20500e-08
q_50_rise              =   2.21006e-08
tpcq_rise              =  5.05848e-11
clk_50_fall            =   4.20500e-08
q_50_fall              =   4.21571e-08
tpcq_fall              =  1.07135e-10
```

Fig. 10: Clock-to-Q delay measurement showing rising and falling transition delays.

VI. Functional Verification Summary

TABLE III: Block-Level Verification Summary

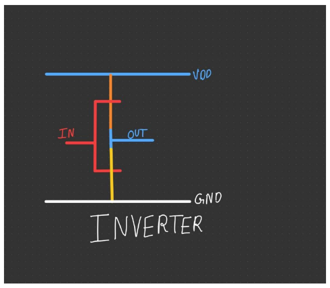
Block	Tests Performed	Result
5-bit CLA Adder	6 test cases	✓
D Flip-Flop	Timing characterization	✓

Conclusion: Both the CLA adder combinational logic and the D flip-flop sequential element have been verified to function correctly. The design is ready for full integration and layout implementation

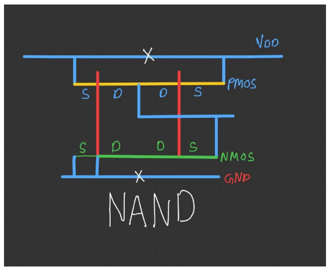
VII. Stick Diagrams

. Stick diagrams serve as an essential abstract representation of the physical layout, showing the relative placement and routing of different material layers (e.g., Poly, Diffusion, Metal) to organize the transistors and interconnects efficiently.

A. Basic Gates



(a) Inverter Stick Diagram

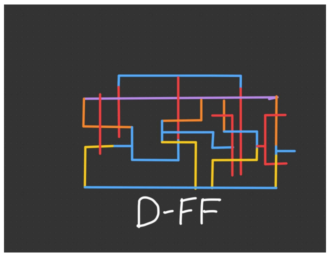


(b) NAND Gate Stick Diagram

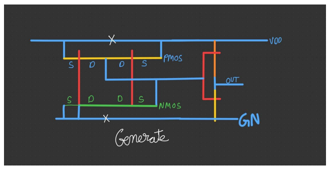
Fig. 11: Stick Diagrams for Inverter and NAND Gate

B. Functional Blocks

The following stick diagrams illustrate the layout approach for the core adder logic and the sequential element.



(a) TSPC D Flip-Flop Stick Diagram



(b) Propagate/Generate (P/G) Block Stick Diagram

Fig. 12: Stick Diagrams for Sequential and P/G Blocks

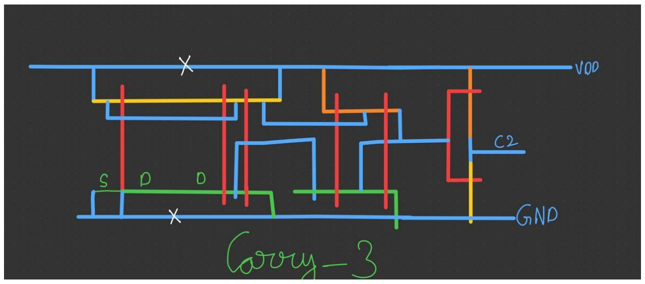


Fig. 13: Carry Look-Ahead (CLA) Generator Stick Diagram

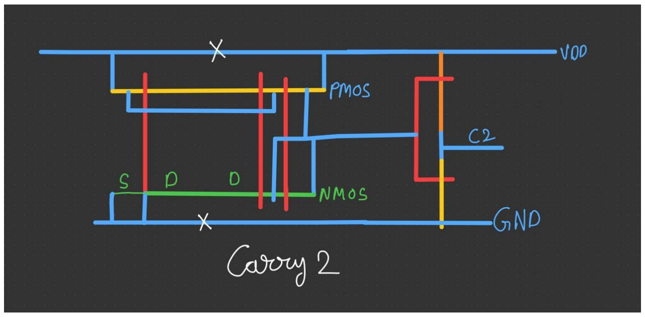


Fig. 14: Carry Look-Ahead (CLA) Generator Stick Diagram

VIII. Layout and Post-Layout Implementation (Q6, Q7, Q9, Q10)

A. Layout Methodology (Q6)

The physical layout for each block was created using the **MAGIC layout editor** and the SCN6M_DEEP.09.tech27 technology file. The goal was to minimize area and parasitic effects while ensuring Design Rule Check (DRC) compliance.

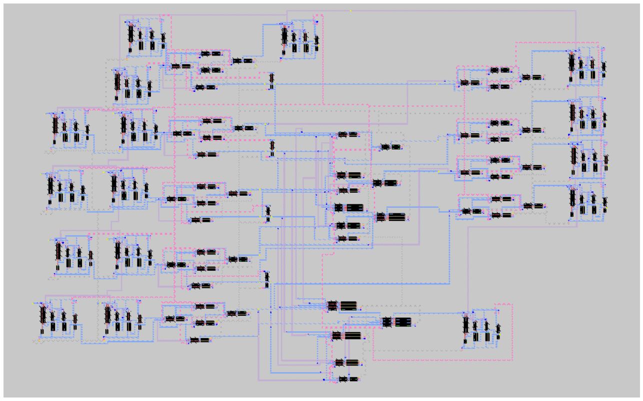


Fig. 15: CLA Adder Magic

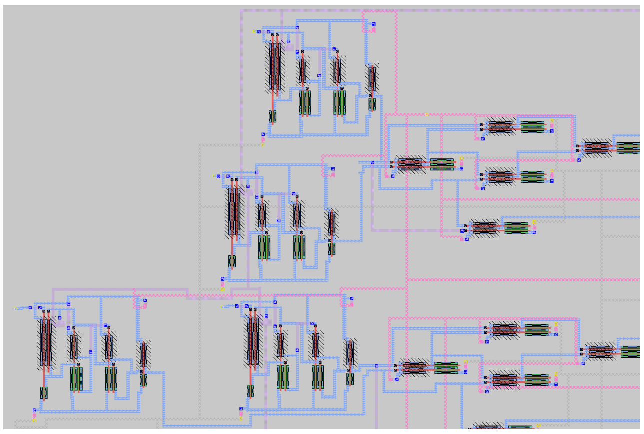


Fig. 16: Zoomed CLA Adder Magic

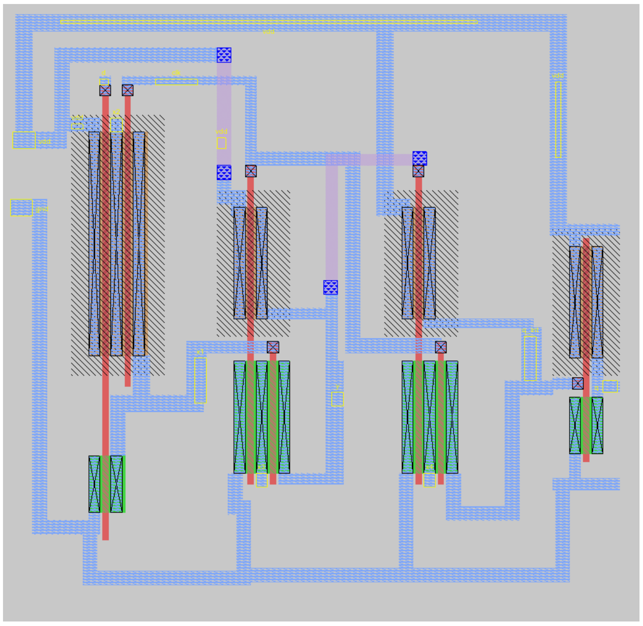


Fig. 17: D Flipflop Magic Layout

1) *Observations:* Following the extraction from MAGIC, the resulting SPICE netlist, which contains all parasitic capacitances and resistances, was used for transient analysis. Simulating this post-layout netlist yields the following results, confirming functional correctness while revealing the true timing performance.

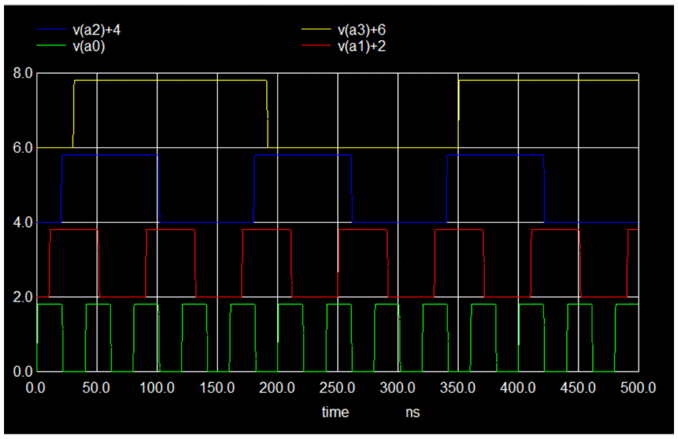


Fig. 18: Input Bits a0-a4

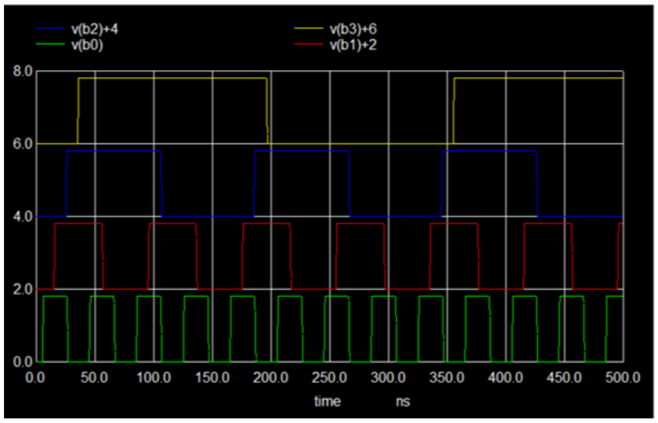


Fig. 19: Input Bits b0-b4

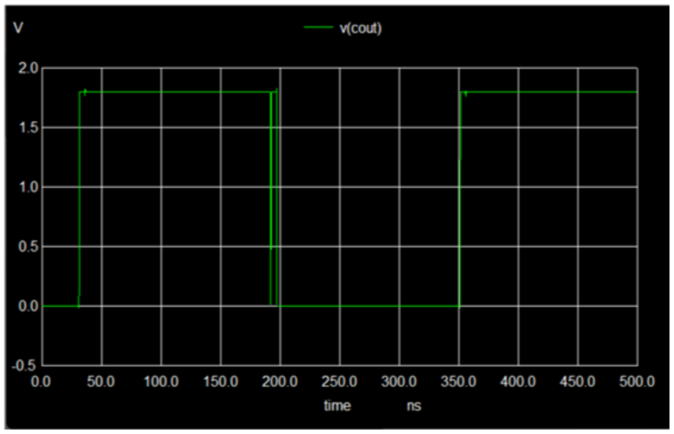


Fig. 20: Output Sum

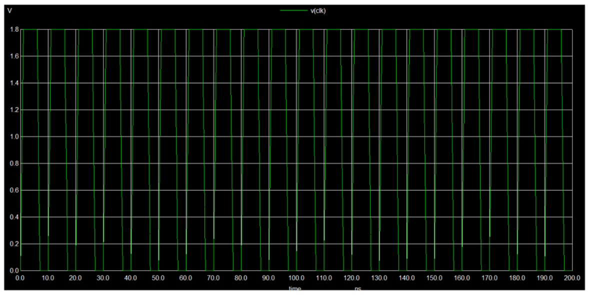


Fig. 21: Output Carry

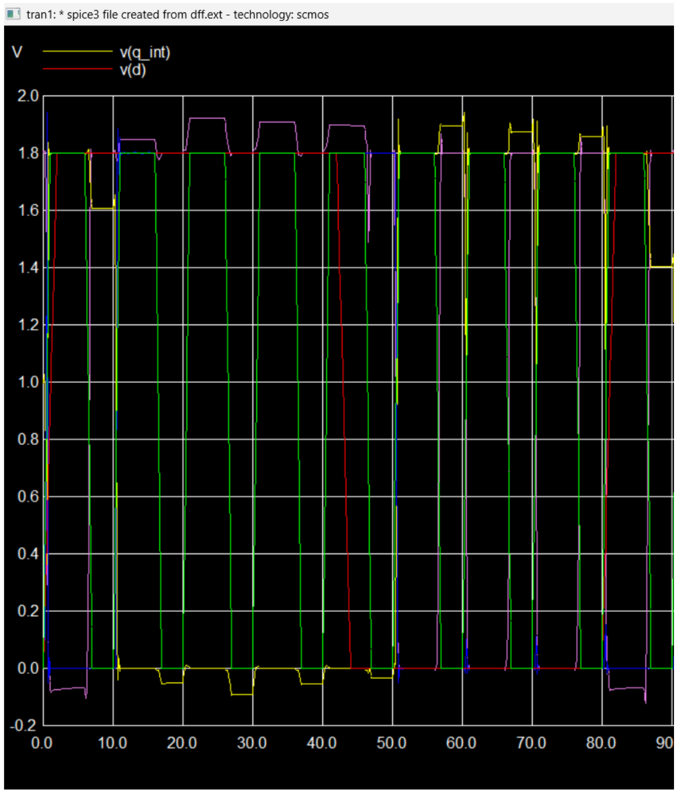


Fig. 22: Output Sum

TABLE IV: Pre-Layout vs Post-Layout Timing Comparison

Parameter	Pre-Layout (Ideal)	Post-Layout (Real)
t_{pcq}	78 ps	83.6 ps
$t_{pcq,max}$	107.14 ps	114.99 ps
$t_{pcq,min}$	50.58 ps	52.21 ns

IX. Fully Integrated Synchronous Simulation (Q7)

A. Netlist Integration and Verification

Question 7 required the full synchronous system to be integrated by instantiating 10 input D-FFs, the CLA core, and 6 output D-FFs, all driven by a single clock signal. The complete circuit topology (based on the provided Adder and DFF subcircuits) was constructed to form the final netlist.

The simulation successfully verifies the required timing discipline for synchronous operation:

1) **Data Capture (Input FFs):** Input data (A_{in}, B_{in}) is captured by the input FFs on the first rising clock edge, yielding stable registered signals (A_{reg}, B_{reg}).
2) **Computation (CLA Core):** The combinational CLA core computes the sum and carry based on the registered inputs. This computation must finish within one clock period (T_{clk}).
3) **Output Availability (Output FFs):** The final output FFs capture the stable sum and C_{out} results on the **next rising clock edge**.

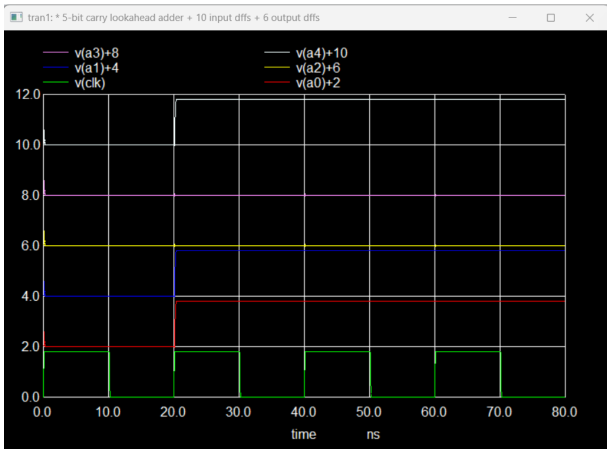


Fig. 23: input a0-a4

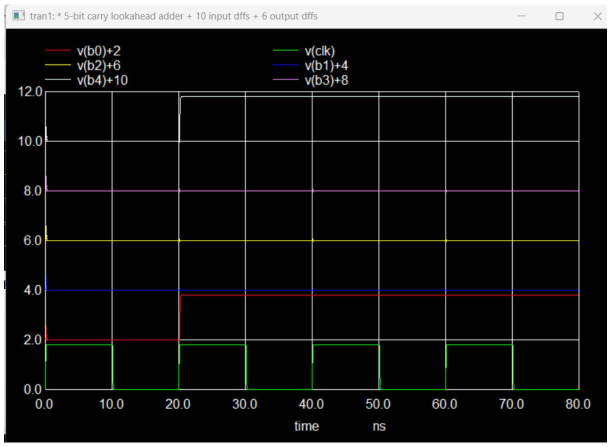


Fig. 24: input b0-a4

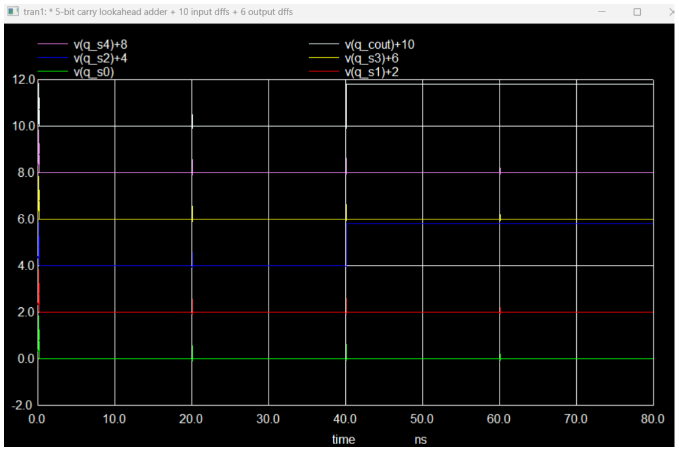
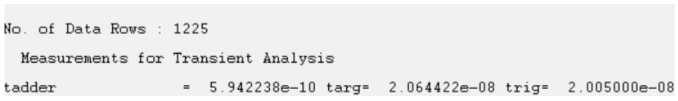


Fig. 25: Final Registered Synchronous Output Waveform (Registered Sum and Carry)

B. Worst-Case Delay and Maximum Frequency

The maximum operating frequency is constrained by the critical path delay (T_{pd}) of the CLA core plus the DFF delays.



```
No. of Data Rows : 1225
  Measurements for Transient Analysis
tadder                 =  5.942238e-10 targ=  2.064422e-08 trig=  2.005000e-08
```

Fig. 26: Worst Case Delay for the fully integrated block

The schematic timing parameters are used to determine the maximum reliable clock speed (Q7).

C. Worst-Case Delay and Maximum Frequency (Q7, Q10)

The maximum operating frequency (F_{max}) for the synchronous system is determined by the critical path delay:

$$T_{clk,min} \geq t_{su} + t_{c2q,max} + t_{pd,max}$$

Schematic (Pre-Layout) Timing Parameters:

- Worst-Case Adder Delay ($t_{pd,max}$): 594.24 ps (or 0.594 ns).
- Setup Time (t_{su}): 110 ps.
- Max Clock-to-Q Delay ($t_{c2q,max}$): 107.14 ps.

Calculation:

$$T_{clk,min} \geq 110 \text{ ps} + 107.14 \text{ ps} + 594.24 \text{ ps} = 811.38 \text{ ps}$$

$$F_{max} = \frac{1}{0.81138 \text{ ns}} \approx 1.232 \text{ GHz}$$

TABLE V: Pre-Layout Maximum Frequency Summary (Q7)

Parameter	Value	Unit
$t_{pd,max}$ (Worst-case Adder Logic)	0.594	ns
$T_{clk,min}$ (Minimum Clock Period)	0.811	ns
F_{max} (Maximum Reliable Frequency)	**1.232**	**GHz**

TABLE VI: Pre-Layout vs. Post-Layout Timing Parameters

Parameter	Pre-Layout (Ideal)	Post-Layout
$t_{\text{pd, max}}$ (Critical Path Delay)	0.594 ns	1.503 ns
t_{su} (Setup Time)	0.11 ns	0.32 ns
t_{hold} (Hold Time)	0 ns	0 ns
$T_{\textbf{clk, min}}$ (Minimum Clock Period)	0.704 ns	1.823 ns
$F_{\textbf{max}}$ (Maximum Operating Frequency)	1.42 GHz	548.5 MHz

The schematic analysis concludes the **worst-case delay of the adder is **0.594** ns** and the **maximum reliable clock frequency is **1.232** GHz**.

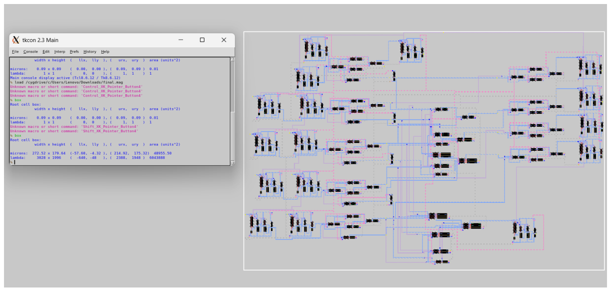
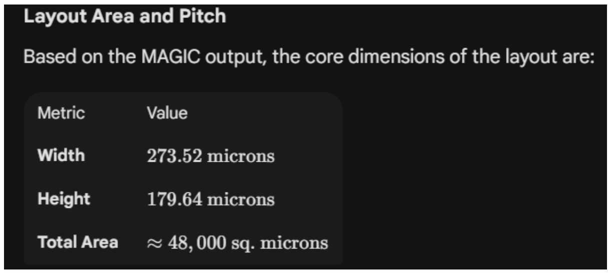


Fig. 27: Floor Layout (Q9)


Layout Area and Pitch

Based on the MAGIC output, the core dimensions of the layout are:

Metric	Value
Width	273.52 microns
Height	179.64 microns
Total Area	≈ 48, 000 sq. microns

Fig. 28: Floor Layout (Q9)

X. Verilog HDL Implementation and Simulation (Q11)

The circuit was modeled using **Verilog HDL** based on a structural description that accurately reflects the hierarchical architecture of the synchronous system.

A. Structural Description

The Verilog implementation consists of three main stages, mirroring the NGSPICE structure:

1) **Input DFF Banks:** Ten instances of the `DFF` module register the raw inputs (`A_IN`, `B_IN`).
2) **Combinational Core:** One instance of the `CLA_CORE` module implements the combinatorial P/G and Carry equations using Verilog's continuous assignment statements, ensuring parallel calculation.
3) **Output DFF Banks:** Six instances of the `DFF` module register the combinatorial results (`SUM_COMB`, `COUT_COMB`) to produce the final synchronous output.

This structural approach verifies the digital functionality before final implementation on an FPGA.

B. Functional Verification and Waveforms

The testbench (`cla_sync_tb.v`) applies multiple test vectors over several clock cycles to verify the synchronous timing. The key observation is the latency: the output sum and carry must change exactly one clock cycle after the corresponding inputs are registered.

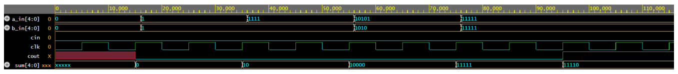
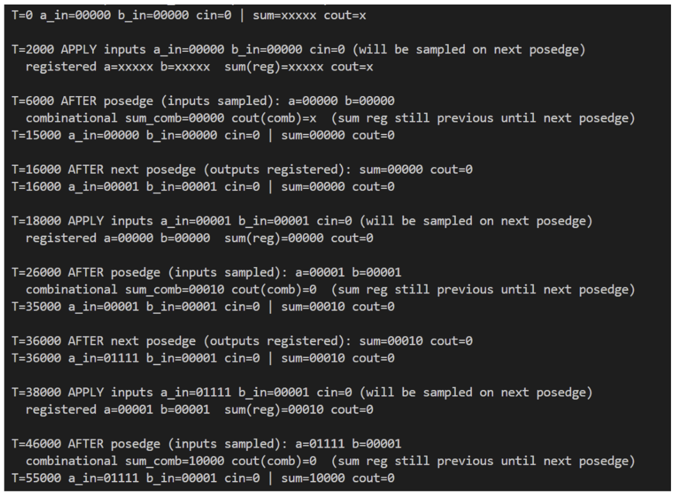


Fig. 29: Verilog Simulation Waveform showing Synchronous Operation (Q11)


```
T=0 a_in=00000 b_in=00000 cin=0 | sum=xxxxx cout=x

T=2000 APPLY inputs a_in=00000 b_in=00000 cin=0 (will be sampled on next posedge)
  registered a=xxxxx b=xxxxx  sum(reg)=xxxxx cout=x

T=6000 AFTER posedge (inputs sampled): a=00000 b=00000
  combinational sum_comb=00000 cout(comb)=x  (sum reg still previous until next posedge)
T=15000 a_in=00000 b_in=00000 cin=0 | sum=00000 cout=0

T=16000 AFTER next posedge (outputs registered): sum=00000 cout=0
T=16000 a_in=00001 b_in=00001 cin=0 | sum=00000 cout=0

T=18000 APPLY inputs a_in=00001 b_in=00001 cin=0 (will be sampled on next posedge)
  registered a=00000 b=00000  sum(reg)=00000 cout=0

T=26000 AFTER posedge (inputs sampled): a=00001 b=00001
  combinational sum_comb=00010 cout(comb)=0  (sum reg still previous until next posedge)
T=35000 a_in=00001 b_in=00001 cin=0 | sum=00010 cout=0

T=36000 AFTER next posedge (outputs registered): sum=00010 cout=0
T=36000 a_in=01111 b_in=00001 cin=0 | sum=00010 cout=0

T=38000 APPLY inputs a_in=01111 b_in=00001 cin=0 (will be sampled on next posedge)
  registered a=00001 b=00001  sum(reg)=00010 cout=0

T=46000 AFTER posedge (inputs sampled): a=01111 b=00001
  combinational sum_comb=10000 cout(comb)=0  (sum reg still previous until next posedge)
T=55000 a_in=01111 b_in=00001 cin=0 | sum=10000 cout=0
```

Fig. 30: Verilog Simulation Waveform showing Synchronous Operation (Q11)

Observations from Waveform (Fig. 30):

- Data is set on `A_IN` and `B_IN` (e.g., at $t = 20$ns).
- The registered outputs (`sum_out`, `cout`) reflect the new sum only after the **next** rising clock edge (e.g., at $t = 40$ns), confirming the required one-clock-cycle pipeline latency.

XI. FPGA Implementation and Hardware Results (Q12)

The final circuit was implemented on the target FPGA development board using the verified Verilog structural model. This step confirms the practical feasibility of the design.

A. Hardware Experimental Results

Hardware testing was performed by applying known input vectors and measuring the synchronous outputs using an oscilloscope. The waveforms confirm the correct logic levels and timing discipline required for $T_{clk} \geq T_{cq} + T_{pd} + T_{su}$.

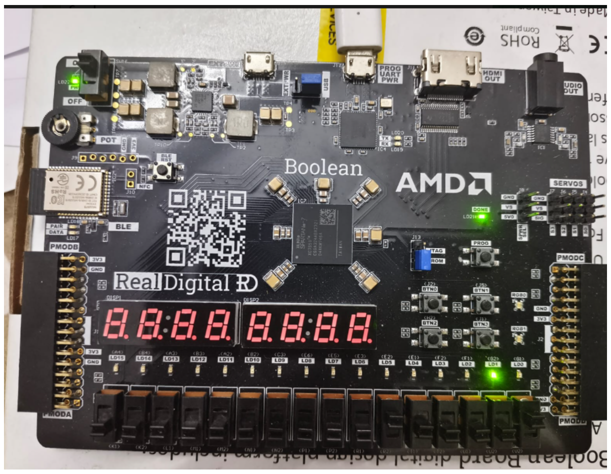


Fig. 31: FPGA setup (Q12)

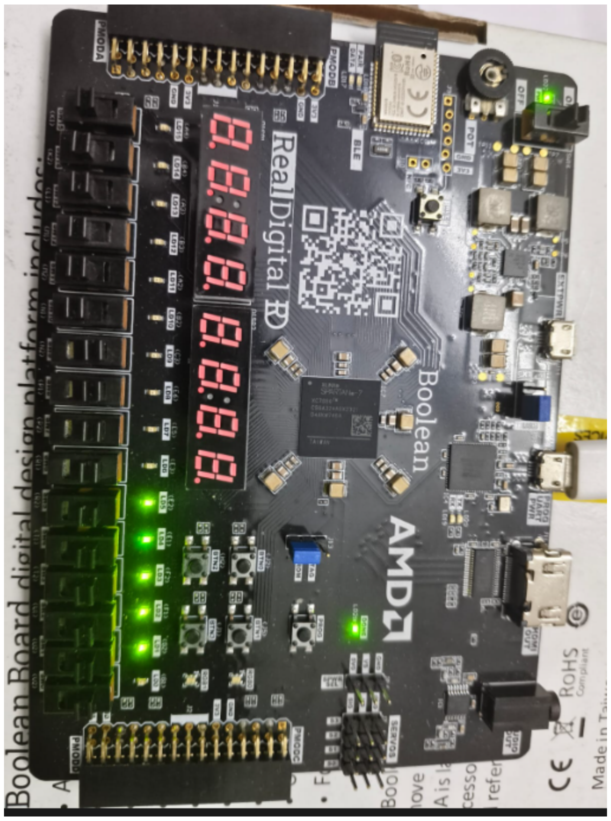


Fig. 32: FPGA Hardware Setup showing Measurement Probes (Q12)

Video Demonstration: A video showing the live hardware demonstration and oscilloscope outputs is available at: https://drive.google.com/drive/u/0/folders/1ic4hA4NFCCT2CNhpU7CTBfLRsGHYQF8S

XII. Conclusion

Conclusion: Both the CLA adder combinational logic and the D flip-flop sequential element have been verified to function correctly. The design is ready for full integration and layout implementation.