4-BIT CLA ADDER

Anantha Eswar Kumar
2023102011
anantha.kumar@students.iiit.ac.in

Abstract: The objective of this project is to design and analyze a 4-bit Carry Lookahead Adder (CLA) using various logic technologies such as Dynamic CMOS, Complementary Pass-Transistor Logic (CPL), and traditional CMOS. The CLA is a critical component in high-performance digital systems, offering significant speed improvements over traditional ripple carry adders by reducing carry propagation time. The project will explore the design and implementation of the CLA in these different logic styles, evaluating their performance in terms of speed, power consumption, and area efficiency.

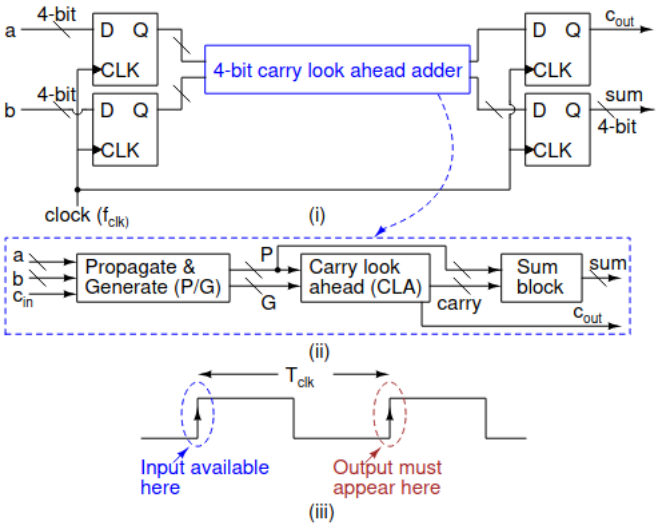


Fig. 1. Project Circuit Overview

I. INTRODUCTION:

A Carry Lookahead Adder (CLA) is an advanced digital adder designed to speed up binary addition by reducing the time required to propagate carries. Unlike traditional ripple carry adders, which propagate carries sequentially through each bit, the CLA generates carry signals in parallel using a lookahead mechanism. This significantly improves the overall addition speed, making it ideal for high-performance computing. The CLA is widely used in processors and digital circuits where fast arithmetic operations are essential.

If the numbers to be added are $a_4a_3a_2a_1$ and $b_4b_3b_2b_1$, then the propagate (p_i) and generate (g_i) signals for each bit position can be defined as (for $i = 1, 2, 3, 4$):

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

and the carry out (c_{i+1}) of the i^{th} bit position can be written as (assuming $c_0 = 0$):

$$c_{i+1} = (p_i \cdot c_i) + g_i, \quad i = 1, 2, 3, 4$$

Thus, c_{i+1} can be expressed entirely in terms of the p_i and g_i functions and sum can be represented as follows:

$$sum_i = p_i \oplus c_i$$

Now let us see the how the gates and flipflops designed and used in circuit

II. GATES:

A. INVERTOR

- The inverter is designed using CMOS technology, which combines complementary NMOS and PMOS transistors to achieve low power consumption and high noise immunity.
- The sizing of the transistors is chosen as follows:
  - NMOS transistor width: W
  - PMOS transistor width: $2W$

  This sizing ensures proper switching characteristics and minimizes delay, as the PMOS transistor typically has lower mobility compared to the NMOS transistor and thus requires a larger width to achieve equal current drive strength.

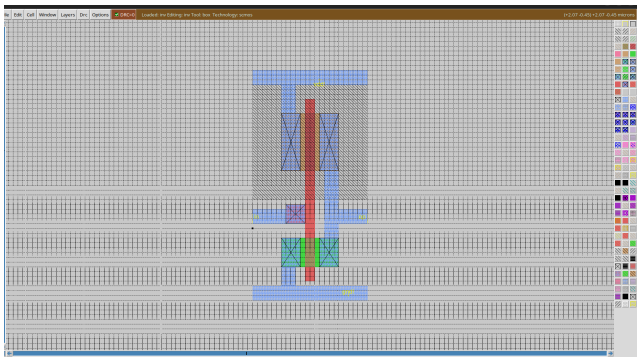


Fig. 2. Magic Layout

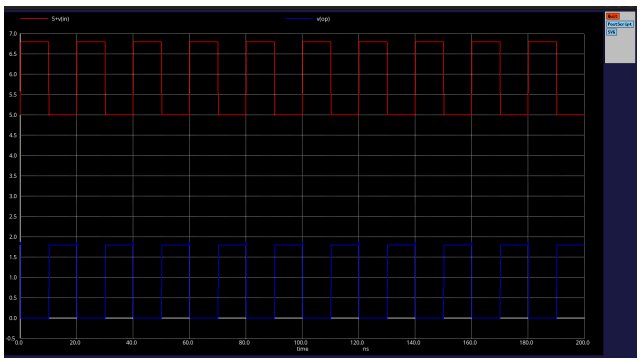


Fig. 3. Post Layout Spice Output

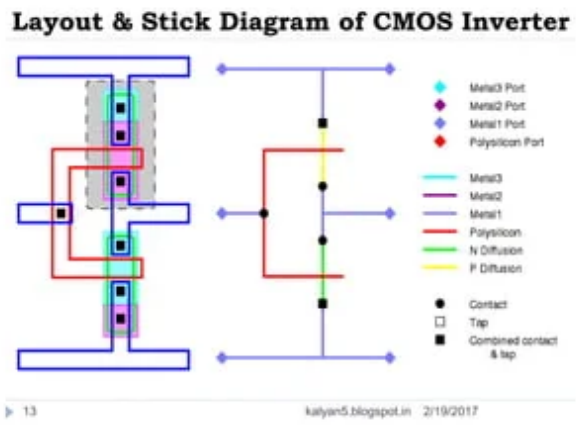


Fig. 4. Stick Diagram for Invertor

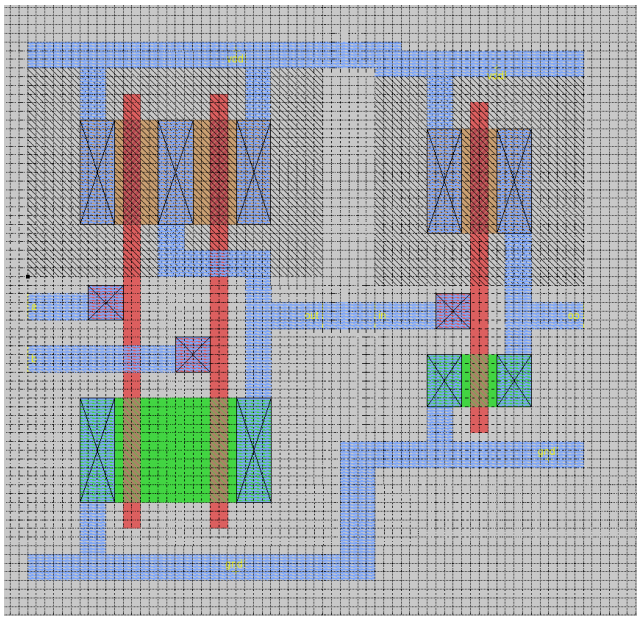


Fig. 6. Magic Layout of AND GATE

B. AND GATE

- The AND gate is implemented using CMOS technology, ensuring high noise margins and low static power consumption.
- The circuit consists of:
  - A pull-up network (PUN) made of PMOS transistors configured to provide a high output (1) only when both inputs are 1.
  - A pull-down network (PDN) made of NMOS transistors that pulls the output low (0) when any input is 0.
- The sizing of the transistors is chosen as follows:
  - NMOS transistor width: $2W$
  - PMOS transistor width: $2W$

  This sizing ensures proper voltage levels and balanced rise and fall times, considering the mobility difference between PMOS and NMOS transistors.

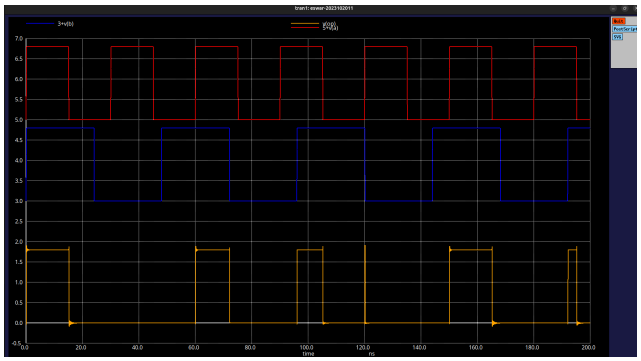


Fig. 7. Post layout spice output

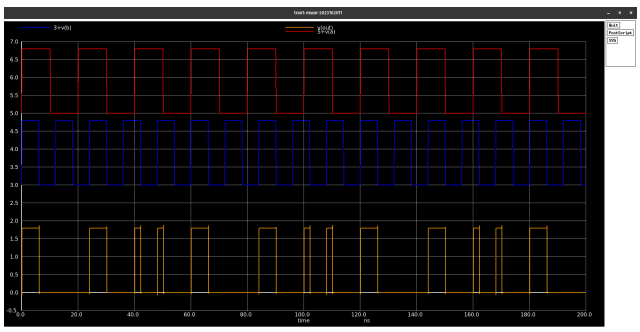
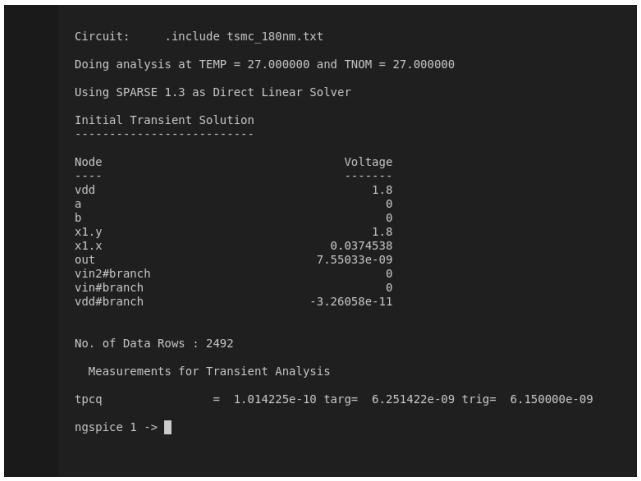


Fig. 5. Ngspice output of AND GATE



```
Circuit:     .include tsmc_180nm.txt

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Using SPARSE 1.3 as Direct Linear Solver

Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
a                                            0
b                                            0
x1.y                                       1.8
x1.x                                 0.0374538
out                                7.55033e-09
vin2#branch                                  0
vin#branch                                   0
vdd#branch                        -3.26058e-11


No. of Data Rows : 2492

  Measurements for Transient Analysis

tpcq               =  1.014225e-10 targ=  6.251422e-09 trig=  6.150000e-09

ngspice 1 ->
```

Fig. 8. Delay of the And gate in Ngspice

```
Initial Transient Solution
--------------------------

Node                                    Voltage
----                                    -------
vdd                                         1.8
a                                             0
b                                             0
in                                          1.8
op                                  2.08909e-08
vin2#branch                                   0
vin#branch                                    0
vdd#branch                         -2.68865e-11


No. of Data Rows : 2162

  Measurements for Transient Analysis

tpcq               =  9.013154e-11 targ=  1.524013e-08 trig=  1.515000e-08
```

Fig. 9. Delay of And gate in magic

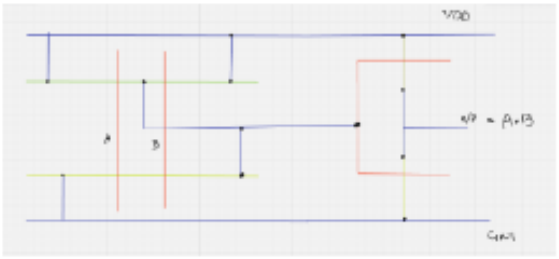


Fig. 10. Stick Diagram for And gate

C. OR GATE

- The OR gate is implemented using CMOS technology, which combines a pull-up network (PUN) and a pull-down network (PDN) to achieve high noise immunity and low power consumption.
- The circuit design includes:
  - A pull-up network (PUN) composed of two PMOS transistors in parallel. The output is pulled high (1) when at least one input is 1.
  - A pull-down network (PDN) consisting of two NMOS transistors in series. The output is pulled low (0) only when both inputs are 0.
- The sizing of the transistors is chosen as follows:
  - NMOS transistor width: W
  - PMOS transistor width: $4W$

  This sizing ensures proper switching characteristics, balancing the rise and fall times, and compensates for the mobility difference between NMOS and PMOS transistors.

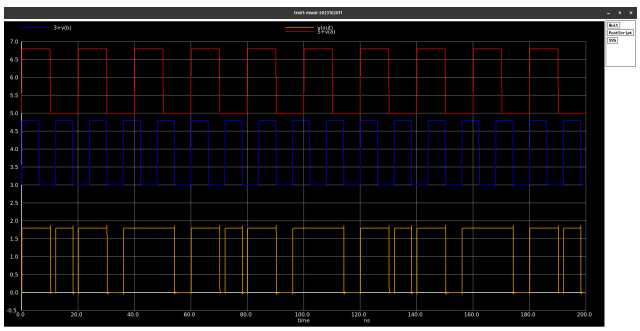


Fig. 11. Ngspice output of OR GATE

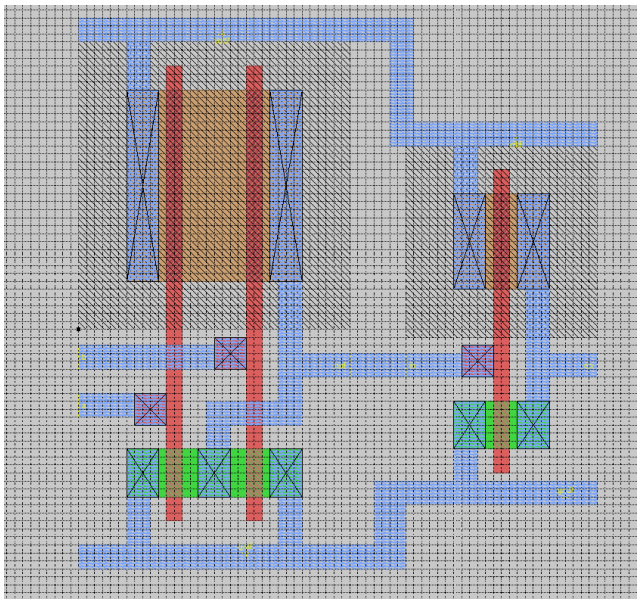


Fig. 12. Magic Layout of OR GATE

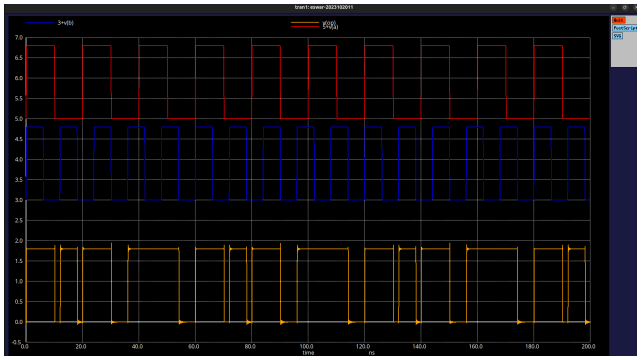


Fig. 13. Post layout spice Output

```
Initial Transient Solution
--------------------------

Node                                    Voltage
----                                    -------
vdd                                         1.8
a                                             0
b                                             0
x1.x                                        1.8
x1.y                                        1.8
out                                 7.55037e-09
vin2#branch                                   0
vin#branch                                    0
vdd#branch                         -6.10094e-11


No. of Data Rows : 2518

  Measurements for Transient Analysis

tpcq               =  9.669731e-11 targ=  1.024670e-08 trig=  1.015000e-08
```

Fig. 14. delay of or gate in ngspice

```
Initial Transient Solution
--------------------------

Node                                    Voltage
----                                    -------
vdd                                         1.8
a                                             0
b                                             0
op                                  2.08909e-08
in                                          1.8
vin2#branch                                   0
vin#branch                                    0
vdd#branch                         -4.28024e-11


No. of Data Rows : 2369

  Measurements for Transient Analysis

tpcq               =  8.243414e-11 targ=  1.023243e-08 trig=  1.015000e-08
```

Fig. 15. Delay of Or gate in magic

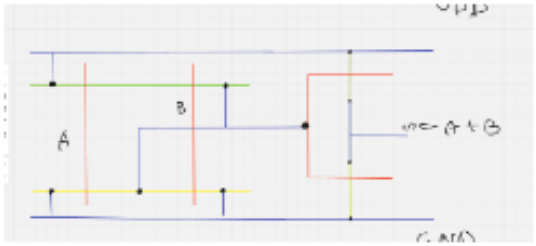


Fig. 16. Stick Diagram of Or gate

D. XOR GATE (CPL Logic)

- The XOR gate is designed using Complementary Pass-Transistor Logic (CPL), which is highly efficient in terms of speed and power consumption for arithmetic circuits.
- CPL implementation relies on the use of NMOS transistors to pass logic levels directly, reducing the number of transistors compared to traditional CMOS.
- The design includes:
  - Two complementary NMOS transistor pairs, controlled by the input signals A and B, to generate both $A \oplus B$.
- The sizing of transistors is as follows:
  - NMOS transistors: Width W (standard sizing for CPL).
  - PMOS transistors in the output inverters: Width $2W$, ensuring full rail-to-rail output voltage swing and balanced rise and fall times.
- Advantages of CPL logic:
  - Reduces the number of transistors compared to traditional CMOS implementation.
  - Provides both the logic output and its complement inherently, which is useful in many digital designs.
  - Low power consumption due to reduced short-circuit currents.

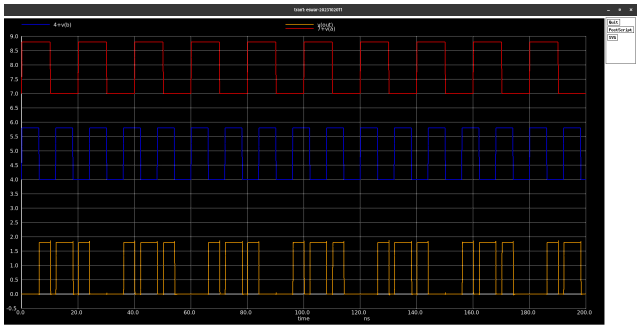


Fig. 17. Ngspice output of XOR GATE

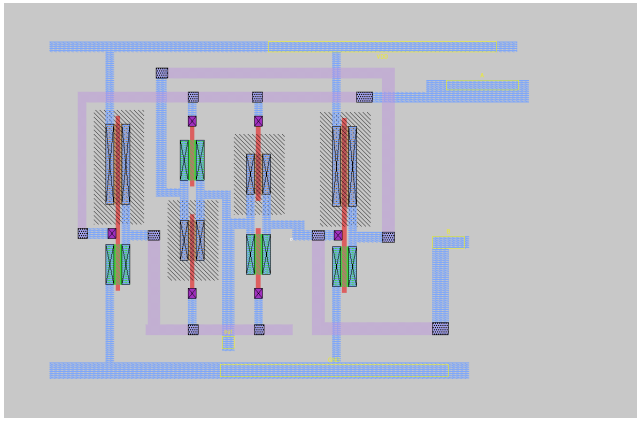


Fig. 18. Magic Layout of XOR GATE

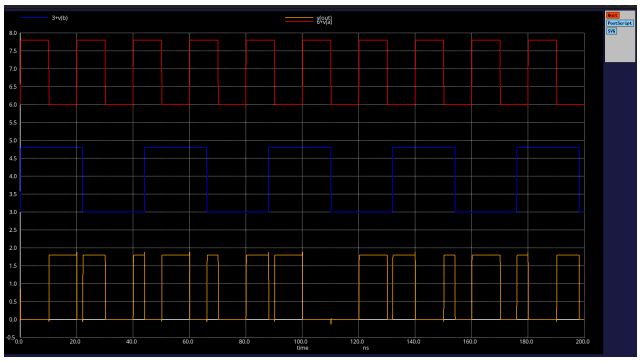


Fig. 19. PostLayout spice Output

```
Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
a                                            0
b                                            0
x4.a_bar                                   1.8
x4.b_bar                                   1.8
x4.y_bar                                   1.8
out                                7.55034e-09
vin2#branch                        3.90815e-11
vin#branch                                   0
vdd#branch                        -1.05491e-10


No. of Data Rows : 2566

  Measurements for Transient Analysis

tpcq             =  6.028998e-11 targ=  6.210290e-09 trig=  6.150000e-09
```

Fig. 20. Delay of Xor gate in Ngspice

```
Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
a                                            0
b                                            0
out                                1.74679e-08
vin2#branch                        4.80146e-11
vin#branch                                   0
vdd#branch                        -9.00805e-11


No. of Data Rows : 2250

  Measurements for Transient Analysis

tpcq             =  4.150567e-11 targ=  1.019151e-08 trig=  1.015000e-08
```

Fig. 21. Delay of Xor gate in magic

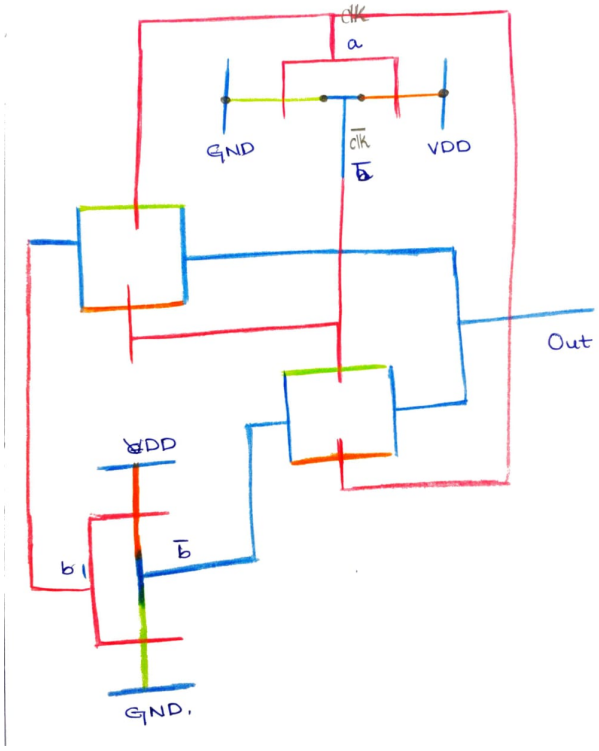


Fig. 22. Stick Diagram of Xor Gate

III. D FLIP-FLOP (TSPC LOGIC)

- The D Flip-Flop is implemented using True Single-Phase Clocking (TSPC) logic, which dynamically stores and trans-

fers data with reduced transistor count and a single clock phase.

- This design uses 11 MOSFETs instead of the traditional 14 by optimizing the logic as follows:
  - The design combines pull-up and pull-down paths efficiently, using fewer transistors while maintaining functionality.
  - A clocked NMOS network is used for data capture, and the stored data is dynamically transferred to the output through successive stages.
- The sizing of transistors:
  - NMOS and PMOS transistor widths are chosen to ensure proper rise and fall times. Typical sizing:
    - NMOS transistors: $W_n = W$.
    - PMOS transistors: $W_p = 2W$.
- Advantages of this TSPC implementation:
  - Reduced power consumption due to fewer transistors and dynamic operation.
  - Simplified clocking scheme with a single-phase clock, reducing clock skew issues.
  - Compact design with fewer MOSFETs, saving area.

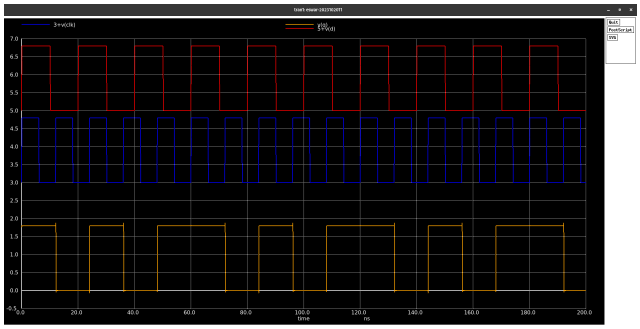


Fig. 23. Ngspice Output

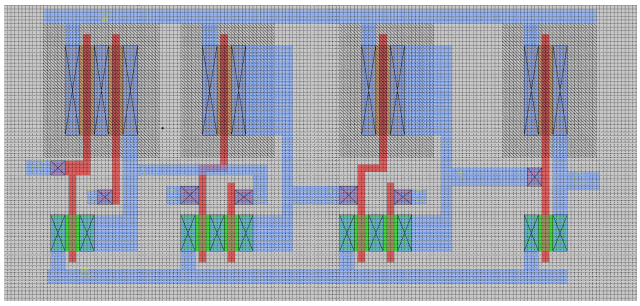


Fig. 24. Magic Layout D flip flop

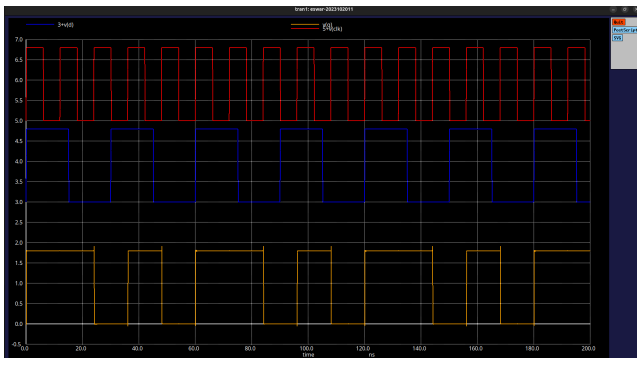


Fig. 25. PostLayout spice Output

```
Circuit:     .include tsmc_180nm.txt

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Using SPARSE 1.3 as Direct Linear Solver

Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
d                                            0
clk                                          0
x1.y                                       1.8
x1.temp                                    1.8
x1.var                                     1.8
x1.x                                   1.41716
x1.last                                0.473434
x1.z                                2.85078e-09
q                                      1.79801
vin2#branch                                  0
vin#branch                                   0
vdd#branch                         -2.91866e-06


No. of Data Rows : 2501

  Measurements for Transient Analysis

tpcq             =  1.319250e-10 targ=  1.218192e-08 trig=  1.205000e-08
```

Fig. 26. Delay of the Flip Flop in Ngspice

```
Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
clk                                          0
d                                            0
x                                          1.8
q                                  0.00291963
qb                                     1.31028
y                                          1.8
vin#branch                                   0
vin0#branch                                  0
vdd#branch                         -3.35667e-06


No. of Data Rows : 2345

  Measurements for Transient Analysis

tpcq               =  1.317921e-10 targ=  2.418179e-08 trig=  2.405000e-08
```

Fig. 27. Delay of the Flip flop in magic

```
thold                =  2.700000e-11
```

Fig. 28. D-flip flop t-hold in Ngspice

```
tpcq_0_to_1          =  6.009264e-11
tpcq_1_to_0          =  1.363941e-10
```

Fig. 29. tcpq 0-1 and 1-0 in Ngspice

```
tsetup               =  5.183000e-10
```

Fig. 30. tsetup in Ngspice

```
thold_post           =  2.400000e-11
```

Fig. 31. thold in magic

```
tsetup_post          =  4.880000e-10
```

Fig. 32. tsetup in magic

```
tpcq_0_to_1_post     =  7.663109e-11
tpcq_1_to_0_post     =  1.264172e-10
```

Fig. 33. tcpq 0-1 1-0 in magic

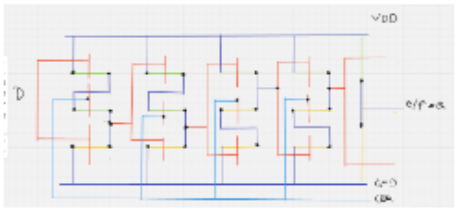


Fig. 34. Stick Diagram of D flip flop

IV. PROPOGATOR AND GENERATOR IN CLA

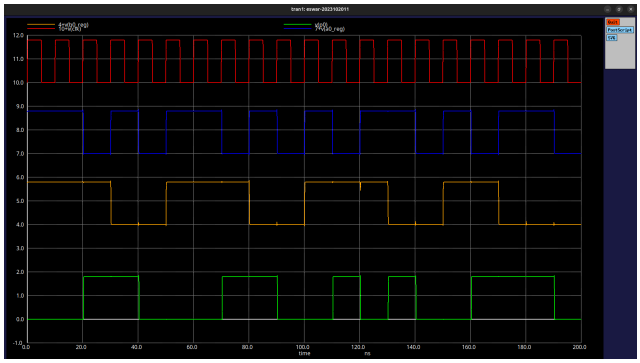


Fig. 35. Propogator ngspice output

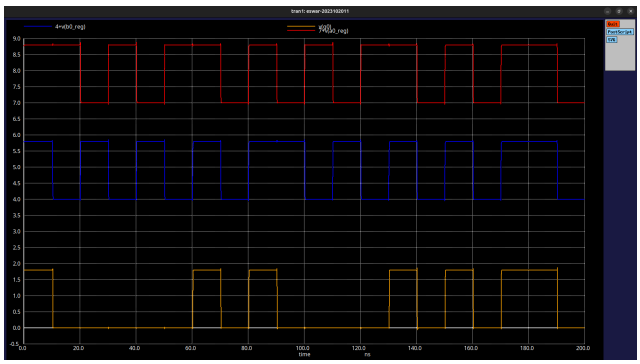


Fig. 36. Generator ngspice output

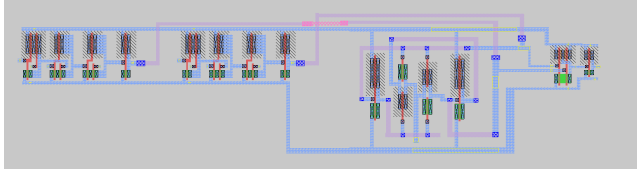


Fig. 37. PG Magic Layout

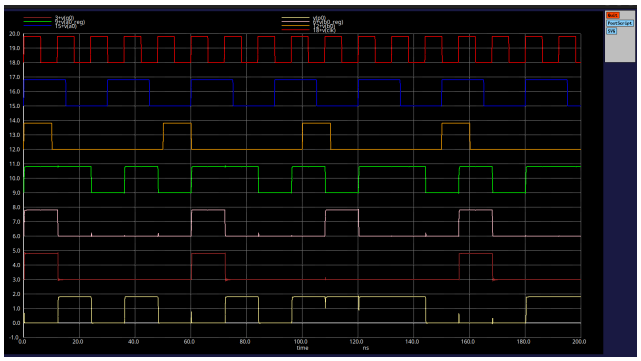


Fig. 38. Post layout of PG block

V. CARRY LOOKAHEAD LOGIC

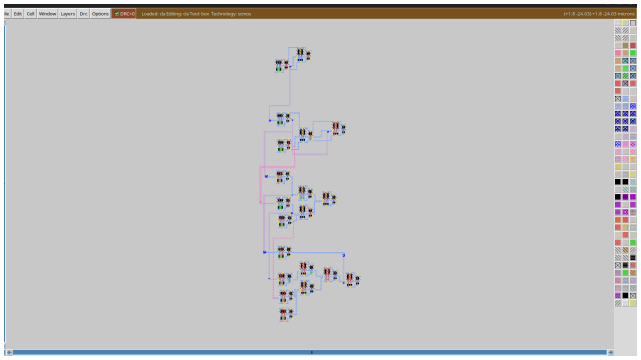


Fig. 39. CLA Magic Layout

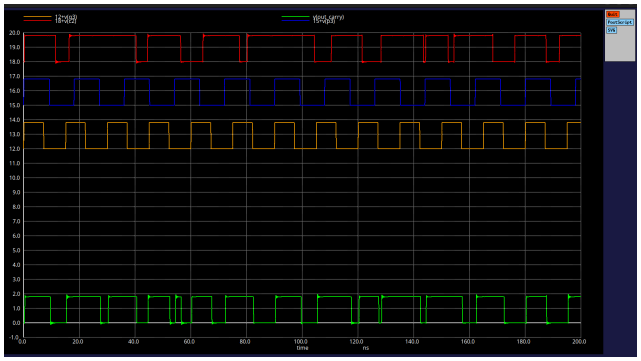


Fig. 40. post Layout of CLA

```
Measurements for Transient Analysis
tpcq                =  3.920890e-10 targ=  1.070943e-08 trig=  1.031734e-08
```

Fig. 41. cla delay in ngspice

```
Measurements for Transient Analysis
tpcq                =  3.147631e-10 targ=  1.068770e-08 trig=  1.037294e-08
```

Fig. 42. cla delay in magic

VI. SUM BLOCK WITH CARRY OUT

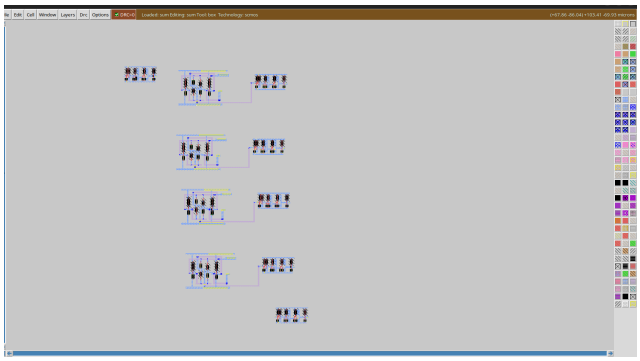


Fig. 43. Sum block Magic Layout

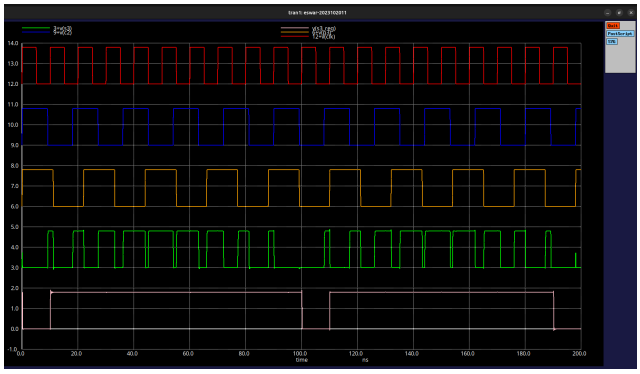


Fig. 44. Post Layout sum

VII. COMPLETE CIRCUIT

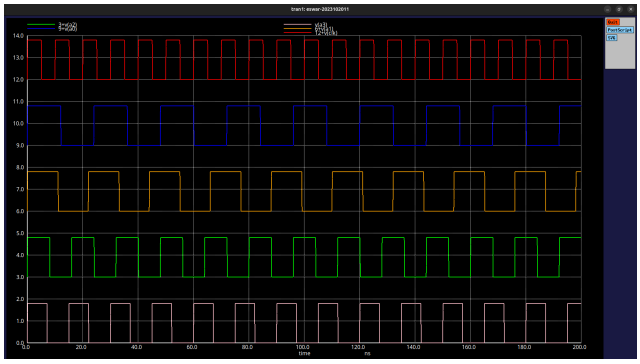


Fig. 45. INPUTS-1

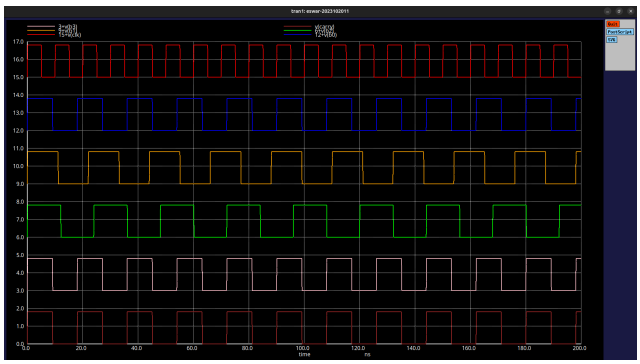


Fig. 46. INPUTS-2

A. CASCADED CIRCUIT

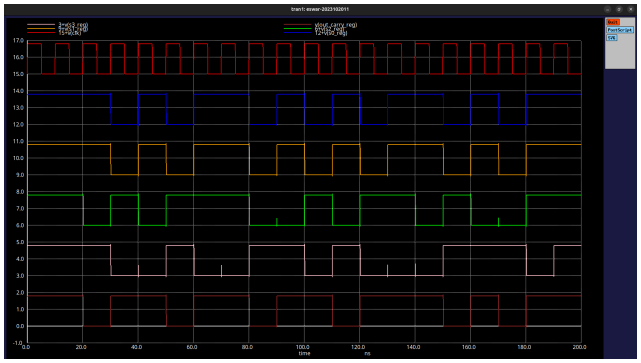


Fig. 47. Cascaded Circuit Output Ngspice

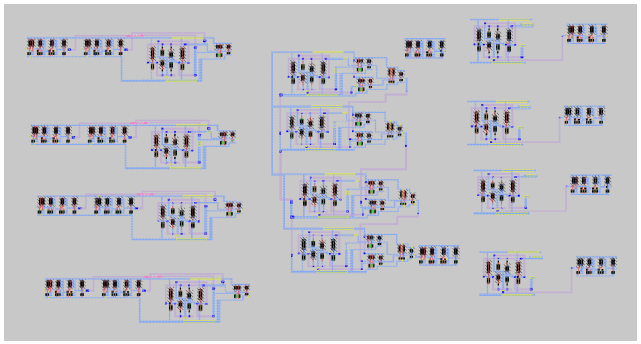


Fig. 48. Complete Magic Layout for cascaded circuit

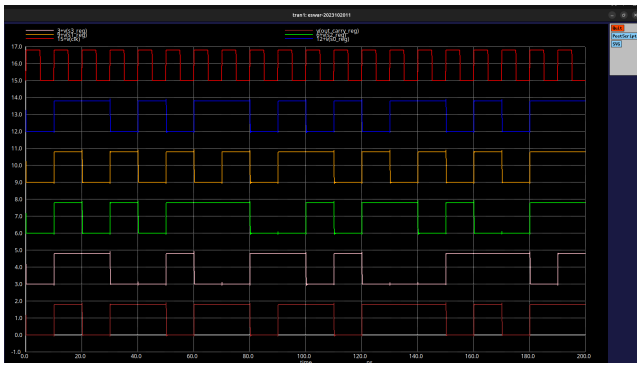


Fig. 49. PostLayout for Cascaded circuit

B. CLA CIRCUIT

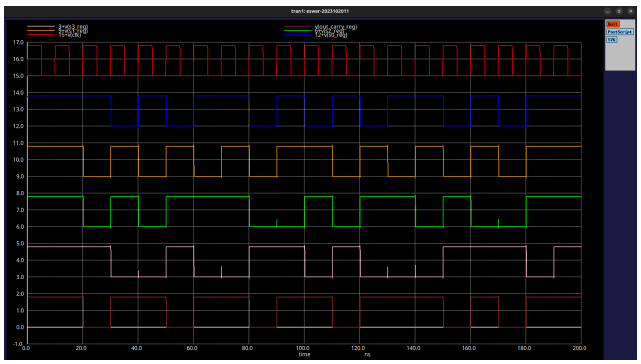


Fig. 50. CLA Circuit Output Ngspice

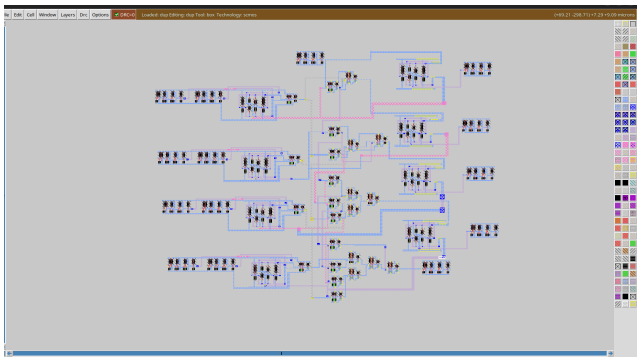


Fig. 51. Complete Magic Layout for CLA circuit

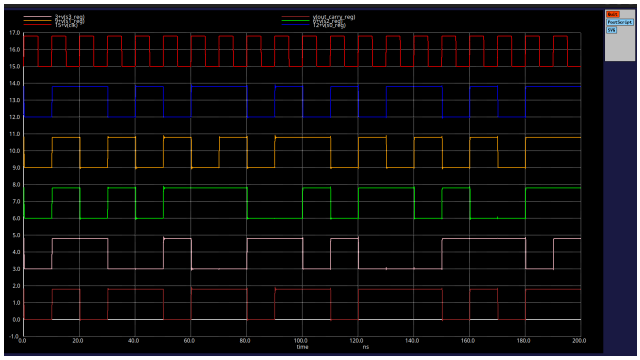


Fig. 52. PostLayout for CLA circuit

```
  Measurements for Transient Analysis
tpd_rise            =  6.515677e-10 targ=  1.591299e-08 trig=  1.526142e-08
tpd_fall            =  7.264903e-10 targ=  2.602036e-08 trig=  2.529387e-08
total_prop_delay    =  6.89029e-10
```

Fig. 53. tpd of circuit in Ngspice

```
No. of Data Rows : 1474
  Measurements for Transient Analysis
tpd_rise            =  5.272663e-10 targ=  1.578255e-08 trig=  1.525529e-08
tpd_fall            =  6.258237e-10 targ=  2.592671e-08 trig=  2.530089e-08
total_prop_delay    =  5.76545e-10
```

Fig. 54. tpd of the circuit in Magic

$$T_{\text{clk min}} = t_{\text{setup}} + t_{\text{pcq}} + t_{\text{pds}}$$
$$T_{\text{clk min}} = 1.32\,\text{ns (NGSPICE)}$$
$$T_{\text{clk min}} = 1.21\,\text{ns (Magic)}$$

$$f_{\text{max}} = \frac{1}{T_{\text{clk min}}}$$
$$f_{\text{max, NGSPICE}} = \frac{1}{1.32 \times 10^{-9}} = 757.58\,\text{MHz}$$
$$f_{\text{max, Magic}} = \frac{1}{1.21 \times 10^{-9}} = 826.45\,\text{MHz}$$

C. OPTIMIZED CIRCUIT

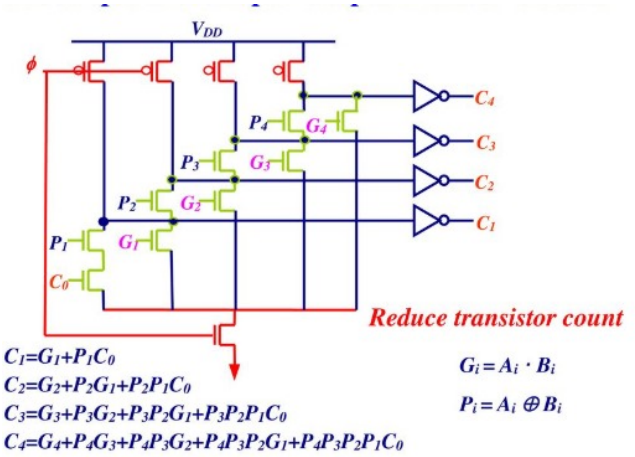


Fig. 55. Layout

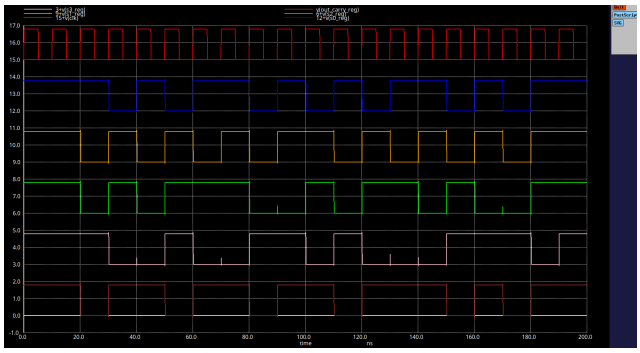


Fig. 56. Optimized Circuit output ngspice

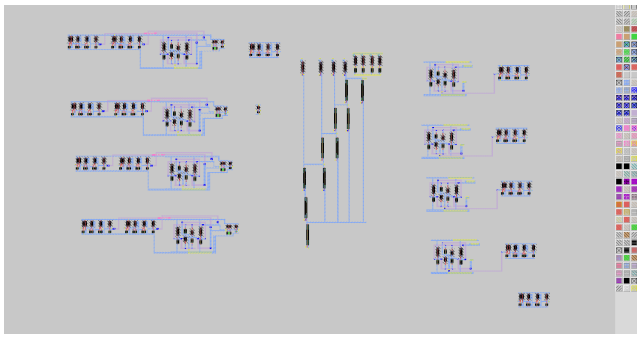


Fig. 57. Complete Magic Layout for Optimized Circuit

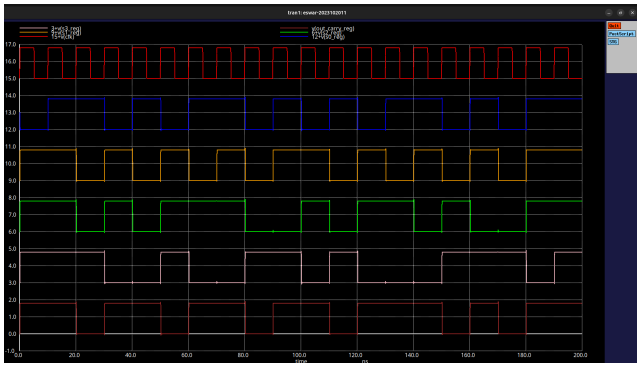


Fig. 58. PostLayout for Optimized circuit

```
tpd_max                =  5.039673e-10
```

Fig. 59. tpd of the circuit

$$T_{\text{clk min}} = t_{\text{setup}} + t_{\text{pcq}} + t_{\text{pds}}$$
$$T_{\text{clk min}} = 1.143\,\text{ns (NGSPICE)}$$
$$T_{\text{clk min}} = 1.109\,\text{ns (Magic)}$$

$$f_{\text{max}} = \frac{1}{T_{\text{clk min}}}$$
$$f_{\text{max, NGSPICE}} = \frac{1}{1.143 \times 10^{-9}} = 874.89\,\text{MHz}$$
$$f_{\text{max, Magic}} = \frac{1}{1.109 \times 10^{-9}} = 901.71\,\text{MHz}$$

VIII. COMPARSION

Component	Prelayout Simulations	Postlayout Simulations
AND Gate	0.101 ns	0.090 ns
OR Gate	0.090 ns	0.082 ns
XOR Gate	0.060 ns	0.0415 ns
t-hold	0.027 ns	0.024 ns
$t_{\text{pcq0}\to1}$	0.060 ns	0.076 ns
$t_{\text{pcq1}\to0}$	0.136 ns	0.124 ns
CLA	0.392 ns	0.314 ns
tpd	0.689 ns	0.576 ns
$T_{\text{clk min}}$	1.32 ns	1.21 ns
f_{max}	757.58 MHz	826.45 MHz

TABLE I
PRELAYOUT AND POSTLAYOUT SIMULATION RESULTS

IX. VERILOG

```
time=0  | clock=0 | a=xxxx | b=xxxx | carry=0 |out_carry=x | sum=xxxx
time=5  | clock=1 | a=0000 | b=0000 | carry=0 |out_carry=x | sum=xxxx
time=10 | clock=0 | a=0000 | b=0000 | carry=0 |out_carry=x | sum=xxxx
time=15 | clock=1 | a=0001 | b=0001 | carry=0 |out_carry=0 | sum=0000
time=20 | clock=0 | a=0001 | b=0001 | carry=0 |out_carry=0 | sum=0000
time=25 | clock=1 | a=0001 | b=0001 | carry=0 |out_carry=0 | sum=0010
time=30 | clock=0 | a=0001 | b=0001 | carry=1 |out_carry=0 | sum=0010
time=35 | clock=1 | a=0011 | b=0011 | carry=1 |out_carry=0 | sum=0010
time=40 | clock=0 | a=0011 | b=0011 | carry=1 |out_carry=0 | sum=0010
time=45 | clock=1 | a=0111 | b=0101 | carry=0 |out_carry=0 | sum=0111
time=50 | clock=0 | a=0111 | b=0101 | carry=0 |out_carry=0 | sum=0111
time=55 | clock=1 | a=0111 | b=0101 | carry=0 |out_carry=0 | sum=1100
time=60 | clock=0 | a=0111 | b=0101 | carry=1 |out_carry=0 | sum=1100
time=65 | clock=1 | a=1111 | b=0001 | carry=1 |out_carry=0 | sum=1100
time=70 | clock=0 | a=1111 | b=0001 | carry=1 |out_carry=0 | sum=1100
time=75 | clock=1 | a=0101 | b=0011 | carry=1 |out_carry=1 | sum=0001
time=80 | clock=0 | a=0101 | b=0011 | carry=1 |out_carry=1 | sum=0001
tb_four_bit_adder_cla.v:48: $finish called at 85 (1s)
time=85 | clock=1 | a=0101 | b=0011 | carry=1 |out_carry=0 | sum=1001
```

Fig. 60. Verilog Outputs

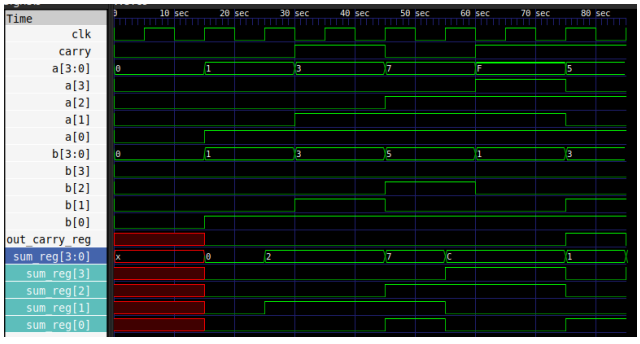


Fig. 61. GTKWAVE output

X. OSCIL0SCOPE AND VIVADO IMAGES

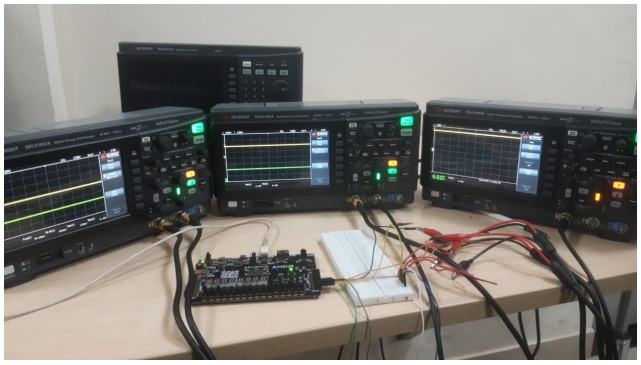


Fig. 62. Oscilloscope image

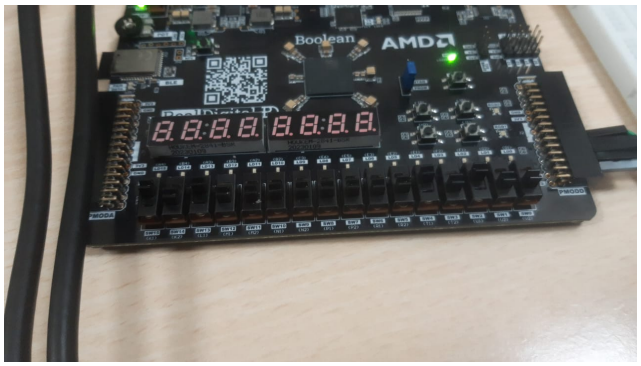


Fig. 63. Vivado Image for Oscilloscope

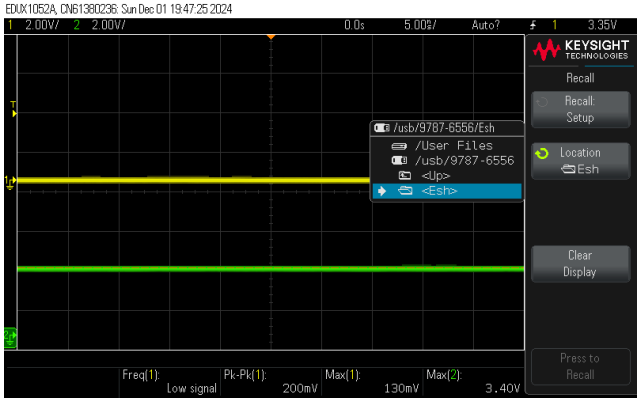


Fig. 64. Oscilloscope image-1

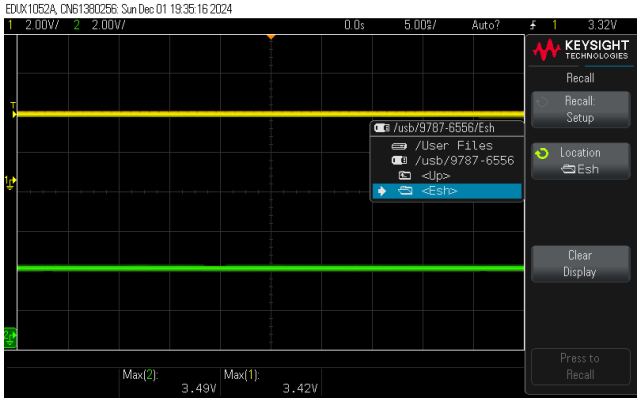


Fig. 65. Oscilloscope image-2

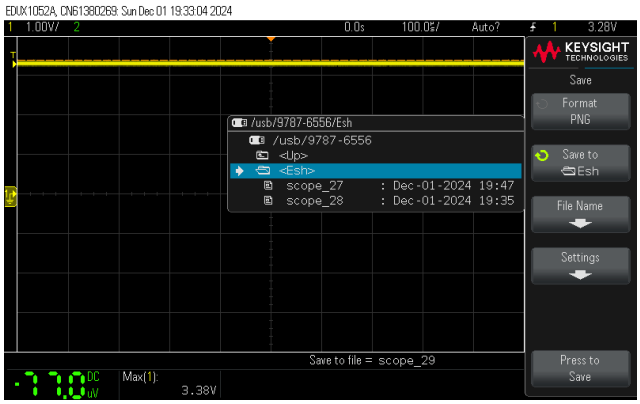


Fig. 66. Oscilloscope image-3

REFERENCES

[1] Digital Logic and Computer Design by Morris Mano.
[2] Computer Architecture and Organization by John P. Hayes
[3] CMOS VLSI Design (fourth edition) by Weste and Harris
[4] Fundamentals of Digital Logic with Verilog Design by Brown and Vranesic.
[5] Lecture notes, tutorials in this course (VLSI)