



NOTRE DAME UNIVERSITY
BANGLADESH

VLSI Design Lab Report-02

Course Code: CSE-4218

Course Title: VLSI Design Lab

Experiment Name: Implementing Sequential Circuit

Submitted by:

Name: Istiak Alam

ID: 0692230005101005

Batch: CSE-20

Submission Date: April 14, 2026

Submitted to:

Nafisa Tabassum,

Lecturer, Dept. of CSE

Notre Dame University Bangladesh

Table of Contents

1	Introduction	1
2	Objective	1
3	Function 1: $(ABC + D)'$	1
4	Function 2: $((AB + CD)E)'$	3
5	Function 3: $((AB + C)(A + B))'$	4
6	Function 4: $(A + B + C)D$	6
7	Function 5: $((A + B)CD)'$	7
8	Conclusion	9

Abstract

This laboratory experiment presents the design and CMOS implementation of given Boolean logic functions using the DSCH2 schematic simulation tool. The circuits were designed following standard CMOS pull-up and pull-down network principles. For each logic function, the corresponding truth table was derived and verified through schematic simulation. Timing diagrams were analyzed to observe the dynamic response and switching behavior of the circuits. The experimental results confirm that the implemented CMOS designs satisfy the expected logical and temporal characteristics of the given Boolean expressions.

1 Introduction

The objective of this laboratory experiment is to implement given Boolean expressions using CMOS logic and verify their functional correctness through schematic design and timing simulation. The circuits were designed using the DSCH2 schematic editor, following standard CMOS pull-up and pull-down network principles. Truth tables and timing diagrams were analyzed to validate logical behavior.

2 Objective

The objectives of this VLSI-Lab experiment are as follows:

- To implement given Boolean expressions using CMOS logic design methodology.
- To design schematic-level CMOS circuits using the DSCH2 tool.
- To verify the logical correctness of the implemented circuits using truth tables.
- To analyze timing diagrams to observe signal transitions and propagation behavior.
- To develop a clear understanding of CMOS pull-up and pull-down network operation.

3 Function 1: $(ABC + D)'$

Boolean Function

$$Y = (ABC + D)'$$

Truth Table

A	B	C	D	Y
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	1	1	0	0
X	X	X	1	0

Schematic Image

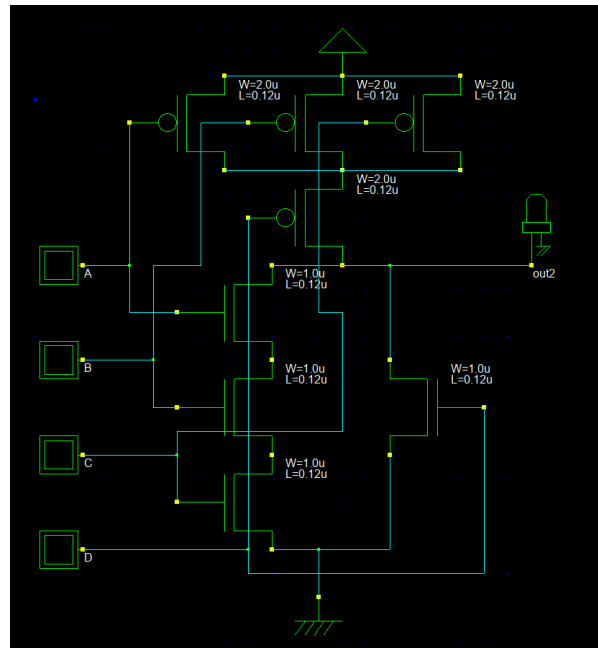


Figure 1: CMOS Schematic of $(ABC + D)'$

Timing Diagram

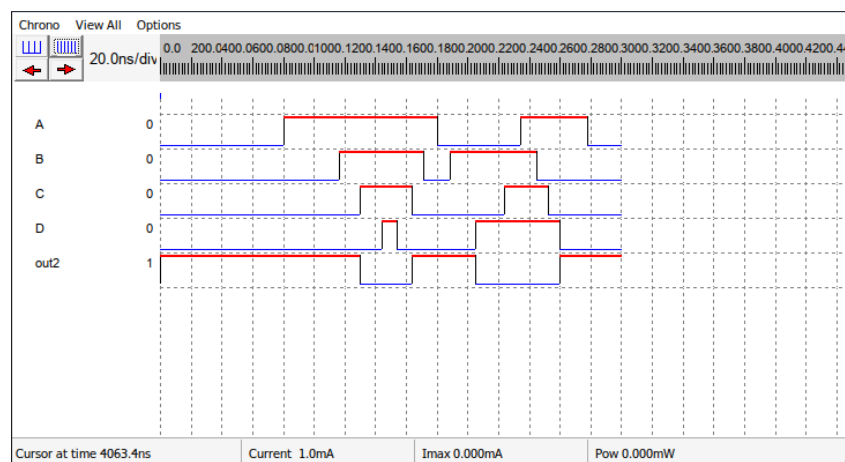


Figure 2: Timing Diagram of $(ABC + D)'$

The timing diagram shows that the output remains HIGH except when all inputs A, B, and C are HIGH or when input D is HIGH.

Observation

The simulation confirms that the output goes LOW when either $ABC = 1$ or $D = 1$. This matches the expected inverted logic behavior.

4 Function 2: $((AB + CD)E)'$

Boolean Function

$$Y = ((AB + CD)E)'$$

Truth Table

A	B	C	D	E	Y
0	X	X	X	X	1
1	1	X	X	1	0
0	X	1	1	1	0
1	X	X	X	1	1

Schematic Image

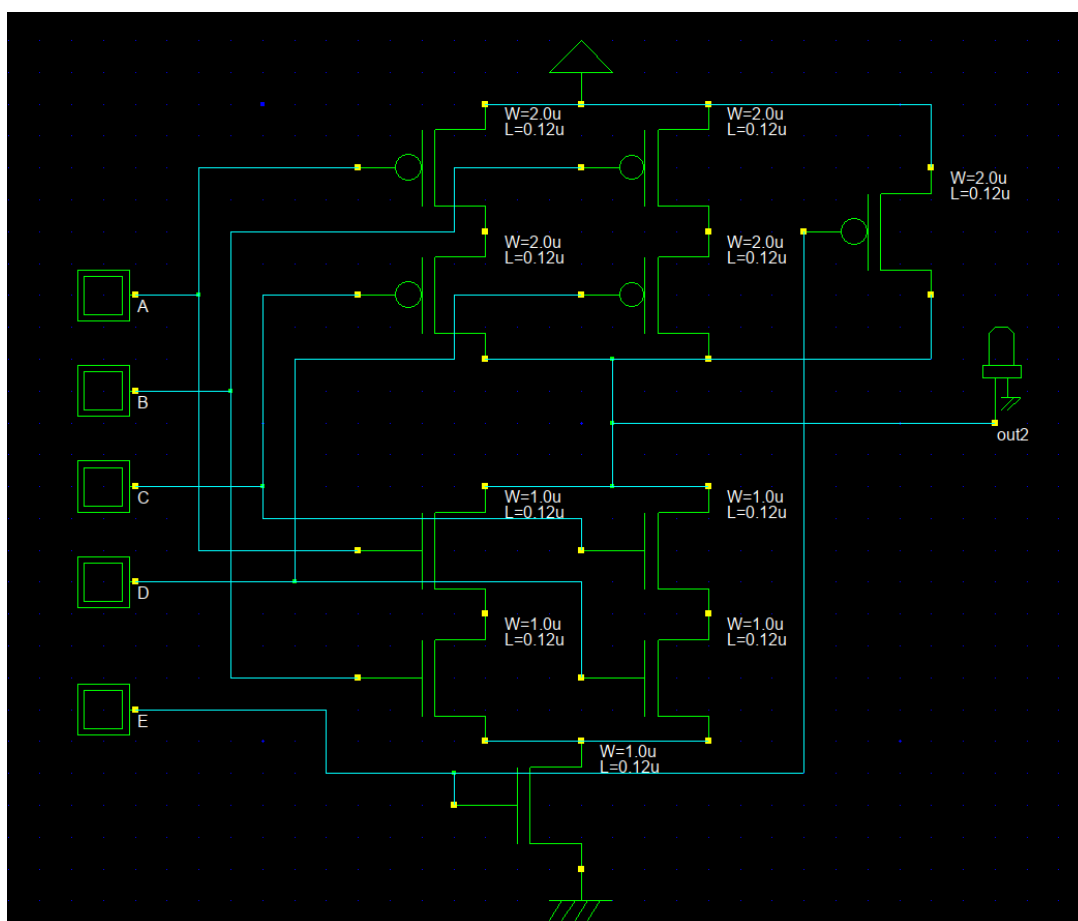


Figure 3: CMOS Schematic of $((AB + CD)E)'$

Timing Diagram

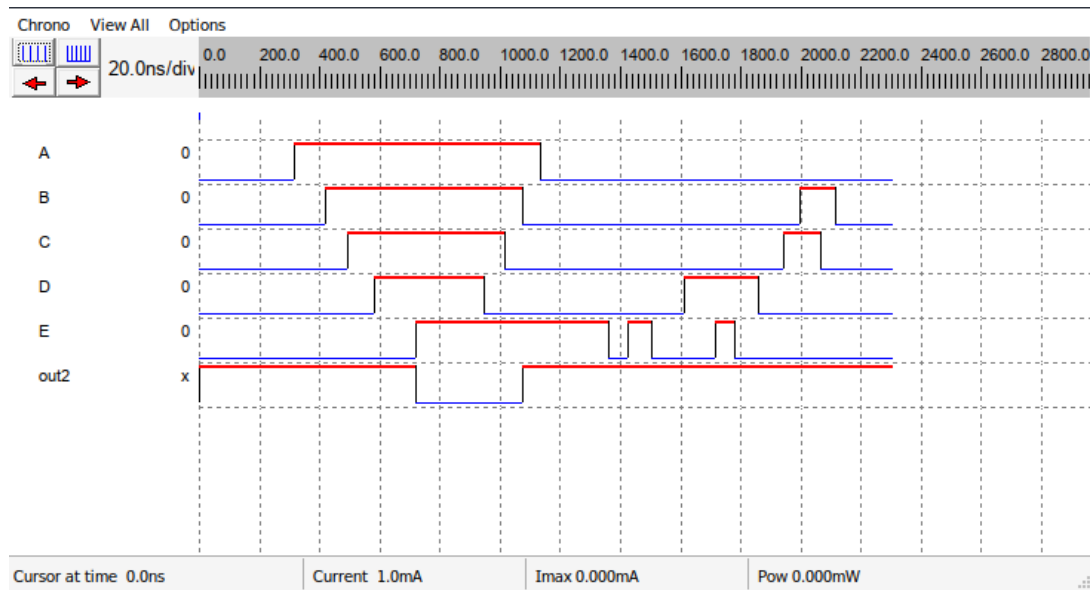


Figure 4: Timing Diagram of $((AB + CD)E)'$

The output transitions LOW only when E is HIGH and either AB or CD is satisfied.

Observation

The circuit output follows the complemented logic, validating correct CMOS pull-up and pull-down network operation.

5 Function 3: $((AB + C)(A + B))'$

Boolean Function

$$Y = ((AB + C)(A + B))'$$

Truth Table

A	B	C	Y
0	0	0	1
0	1	0	1
1	0	0	1
1	1	0	0
1	0	1	0
1	1	1	0

Schematic Image

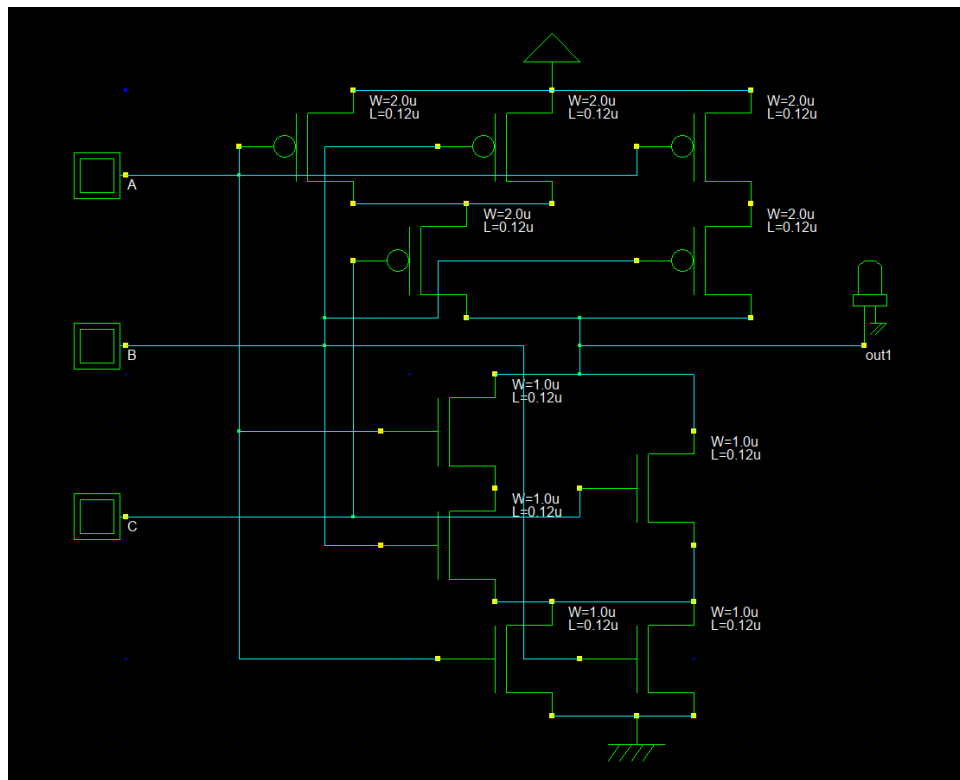


Figure 5: CMOS Schematic of $((AB + C)(A + B))'$

Timing Diagram

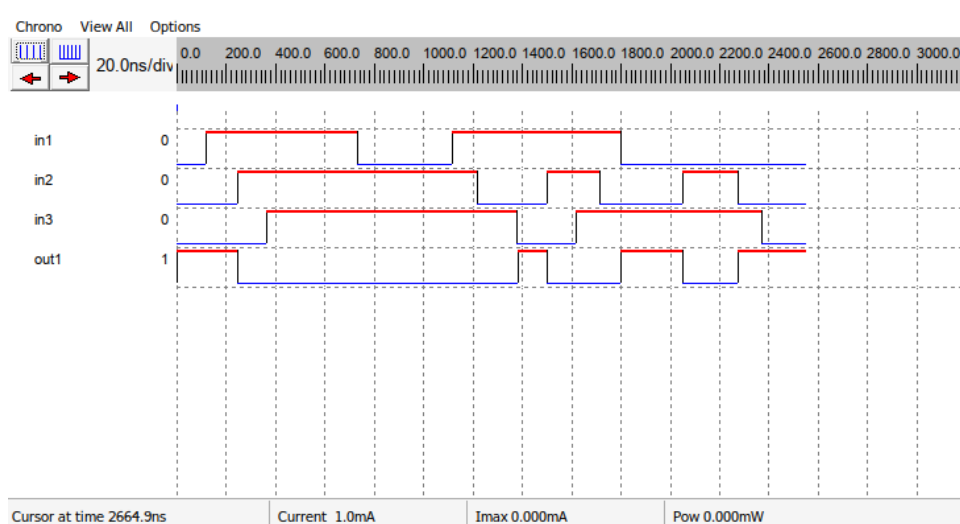


Figure 6: Timing Diagram of $((AB + C)(A + B))'$

The output remains HIGH only when the AND of the two expressions is false.

Observation

Simulation results matched the inverted product-of-sums logic expression.

6 Function 4: $(A + B + C)D$

Boolean Function

$$Y = (A + B + C)D$$

Truth Table

A	B	C	D	Y
X	X	X	0	0
1	X	X	1	1
0	1	X	1	1
0	0	1	1	1
0	0	0	1	0

Schematic Image

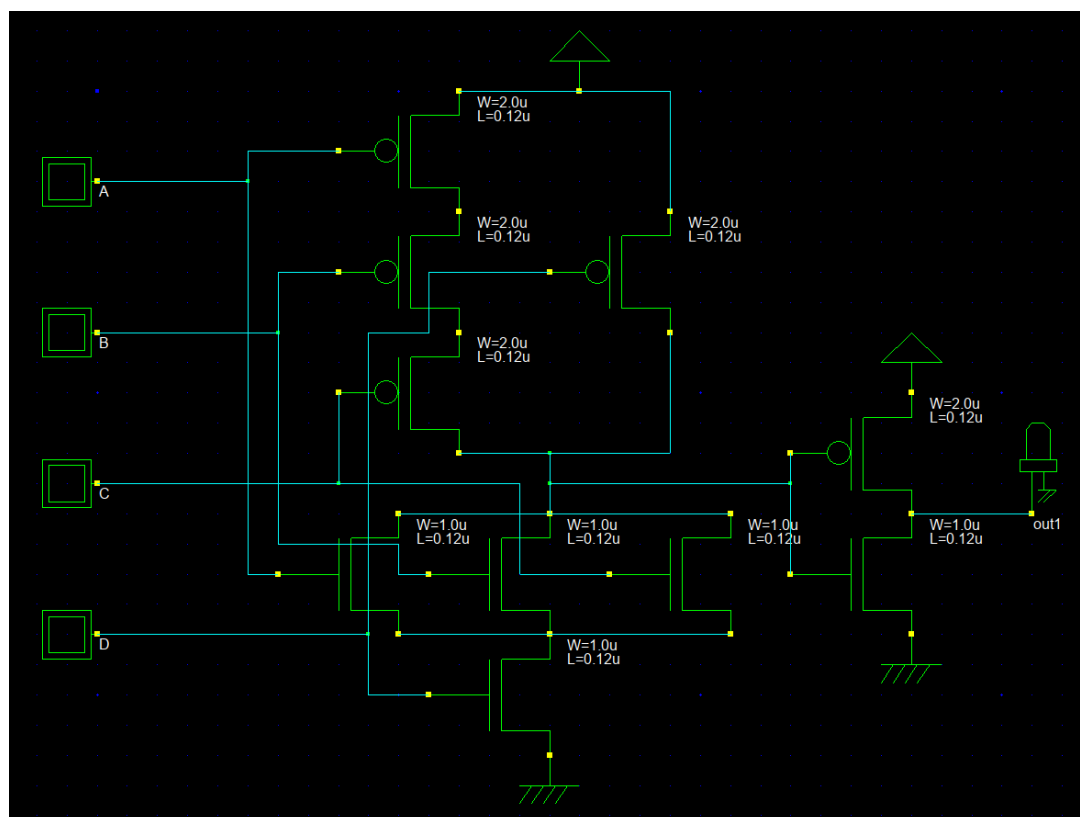


Figure 7: CMOS Schematic of $(A + B + C)D$

Timing Diagram

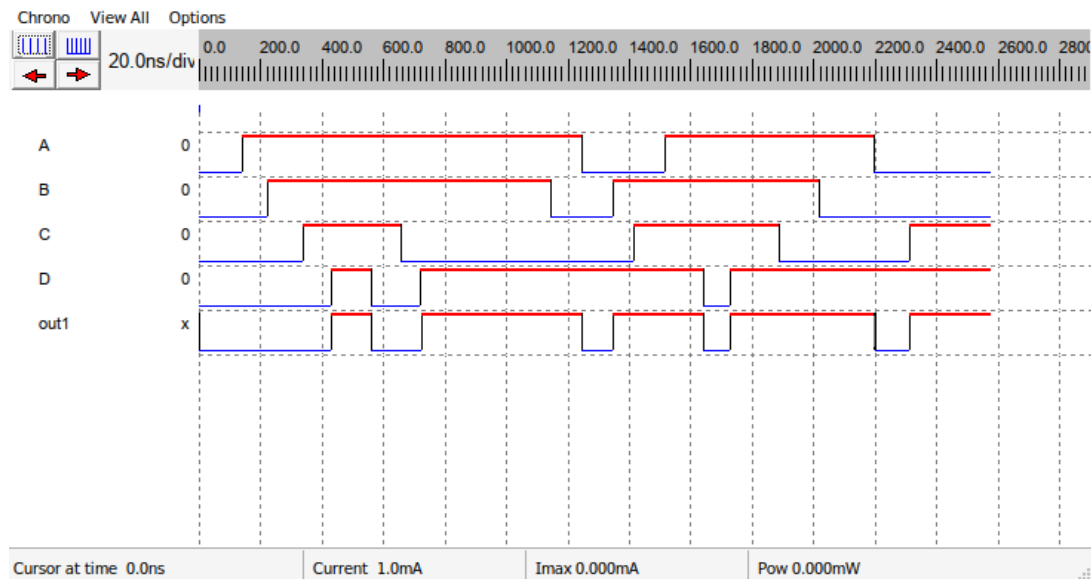


Figure 8: Timing Diagram of $(A + B + C)D$

The output is HIGH only when D is HIGH and at least one of A, B, or C is HIGH.

Observation

The simulation confirms correct AND-OR logic behavior.

7 Function 5: $((A + B)CD)'$

Boolean Function

$$Y = ((A + B)CD)'$$

Truth Table

A	B	C	D	Y
X	X	X	0	1
0	0	1	1	1
1	X	1	1	0
X	1	1	1	0

Schematic Image

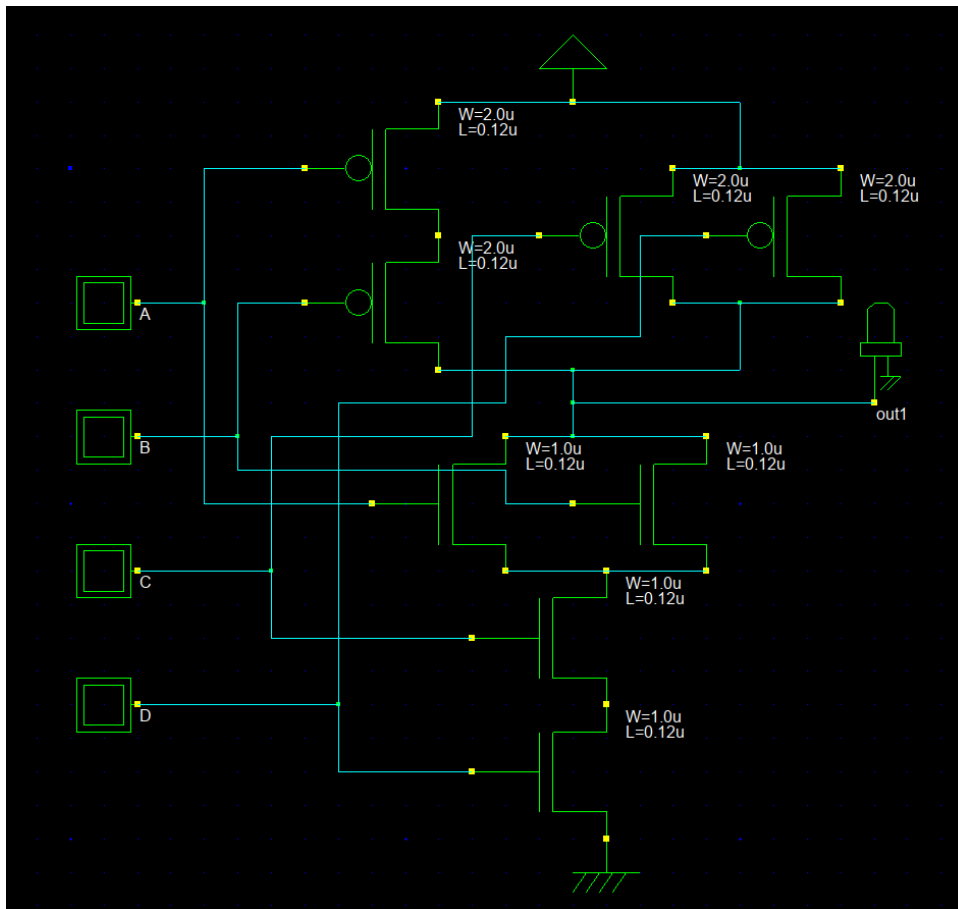


Figure 9: CMOS Schematic of $((A + B)CD)'$

Timing Diagram

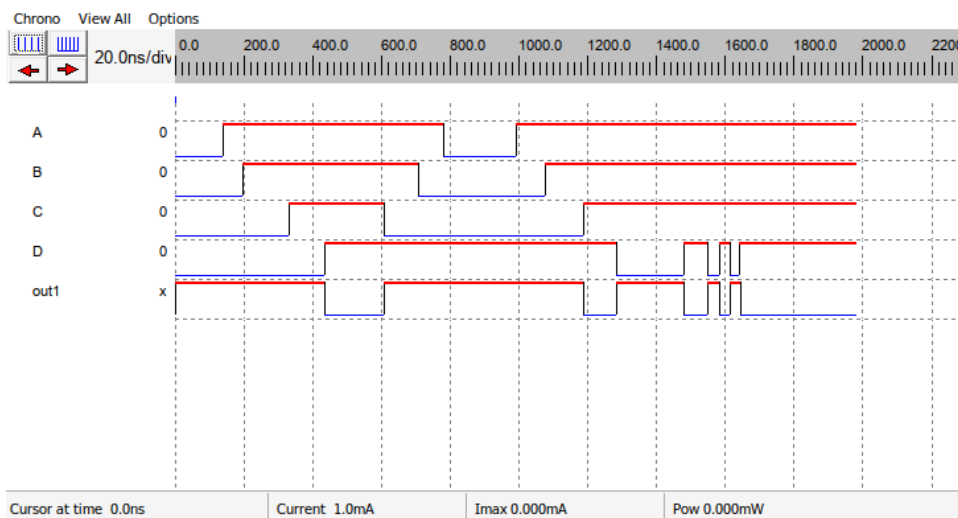


Figure 10: Timing Diagram of $((A + B)CD)'$

Observation

The output goes LOW only when C and D are HIGH and either A or B is HIGH. Simulation results confirm correct inverted AND–OR CMOS implementation.

8 Conclusion

All CMOS circuits were successfully implemented and verified using DSCH2. Truth tables and timing diagrams matched theoretical expectations, confirming correct logical and temporal behavior of each function.