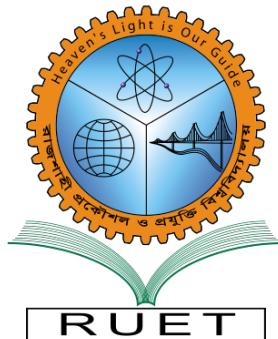


Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design Sessional

Lab Reports

Submitted to	Submitted by
Md. Faysal Ahamed Assistant Professor Dept of ECE, RUET & Moloy Kumar Ghosh Lecturer Dept of ECE, RUET	Md. Tajim An Noor Roll: 2010025

Table of Contents

Exp No.	Experiment Name	Date
1	Implementation of CMOS Inverter.	July 4, 2025
2	Implementation of NMOS Ratio-less Inverter.	July 4, 2025
3	Implementation NAND Gate and NOR Gate Using CMOS	August 8, 2025
4	Design and Observe the Characteristics Curve of CMOS Circuit	September 9, 2025
5	Study and Simulation of 1-Bit SRAM Using Microwind	October 12, 2025
6	Implementation of Half Adder Circuit Using CMOS in Microwind	October 20, 2025

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design

Experiment Date: July 4, 2025,
Submission Date: August 11, 2025

Lab Report 1:
Implementation of CMOS Inverter.

Submitted to Moloy Kumar Ghosh Lecturer Dept of ECE, Ruet	Submitted by Md. Tajim An Noor Roll: 2010025
---	---

Contents

Theory	1
Required Tools	2
Circuit Schematic	2
Output	3
Discussion and Conclusion	5
References	5

Implementation of CMOS Inverter.

Theory

A CMOS inverter is a fundamental building block in digital logic, consisting of a PMOS and an NMOS transistor connected in a complementary arrangement [1]. The PMOS transistor is linked to the supply voltage (V_{DD}), while the NMOS is connected to ground. Both transistors share the same input signal (V_{in}), and the output is taken from the junction between them. When the input is low (logic 0), the PMOS conducts and the NMOS is off, resulting in a high output (logic 1). Conversely, when the input is high (logic 1), the NMOS turns on, the PMOS switches off, and the output drops to logic 0. This complementary switching greatly reduces static power consumption, making CMOS technology highly energy efficient [2].

An important dynamic property of CMOS inverters is the transition time, which includes rise time (t_r) and fall time (t_f). These times indicate how quickly the output voltage responds to changes in the input. Rise time refers to the interval during which the output increases from about 10% to 90% of V_{DD} , while fall time measures the decrease from 90% to 10% of V_{DD} . The speed of these transitions is crucial for high-speed digital circuits [3].

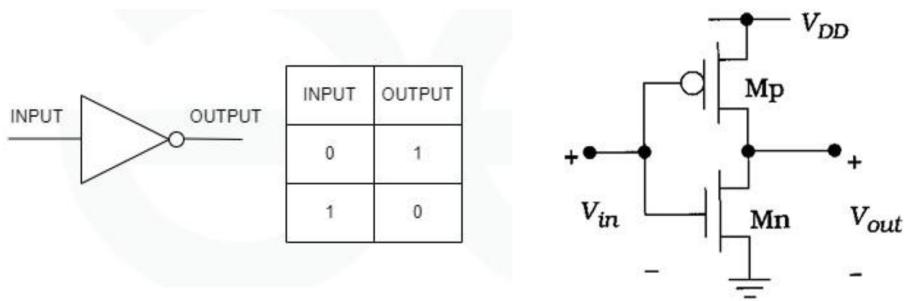


Figure 1: Block diagram of a CMOS inverter circuit.[4]

The rise and fall times of a CMOS inverter depend on the output load capacitance (C_{out}) and the aspect ratios (W/L) of the transistors. Because PMOS transistors have lower carrier mobility (μ_p) than NMOS transistors (μ_n), the PMOS is usually

made wider—often about twice the width of the NMOS—to equalize the speeds of the rising and falling edges [1]. This matching is achieved when:

$$\frac{W_1}{L_1} \mu_n = \frac{W_2}{L_2} \mu_p$$

where T_1 is the NMOS and T_2 is the PMOS. This relationship helps ensure that the inverter switches symmetrically.

The propagation delay due to the output capacitance can be approximated by:

$$t_r \approx 35 \cdot \frac{C_{out}}{W_1/L_1} \text{ ns}$$

where C_{out} is in picofarads and W_1/L_1 is the NMOS aspect ratio. This means that larger output capacitance or smaller transistor sizes result in slower signal transitions [2].

Required Tools

- Microwind
- MS Word
- L^AT_EX

Circuit Schematic

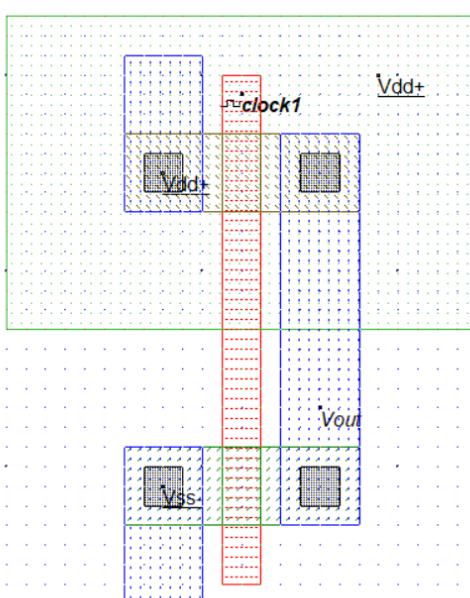


Figure 2: CMOS Inverter Schematic without Load Capacitor

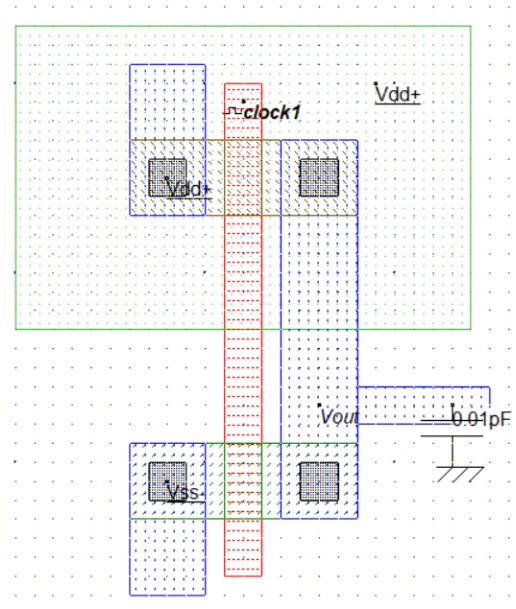


Figure 3: CMOS Inverter Schematic with Load Capacitor

Output

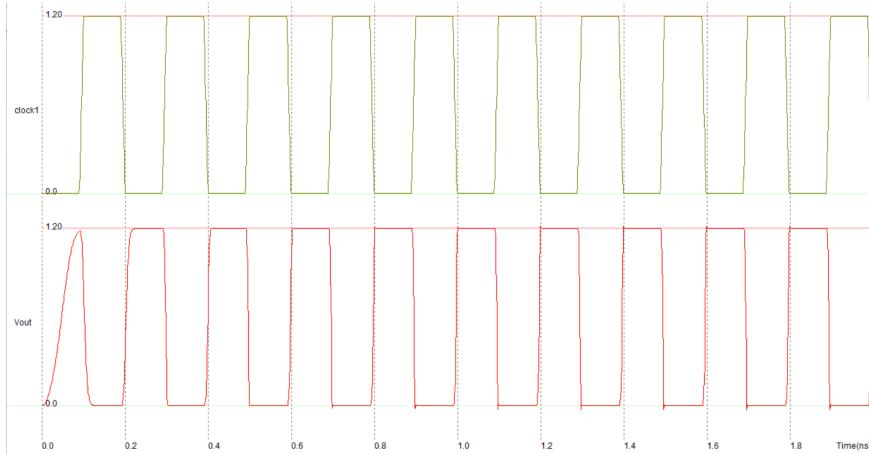


Figure 4: Plot of voltage over time without a capacitor.

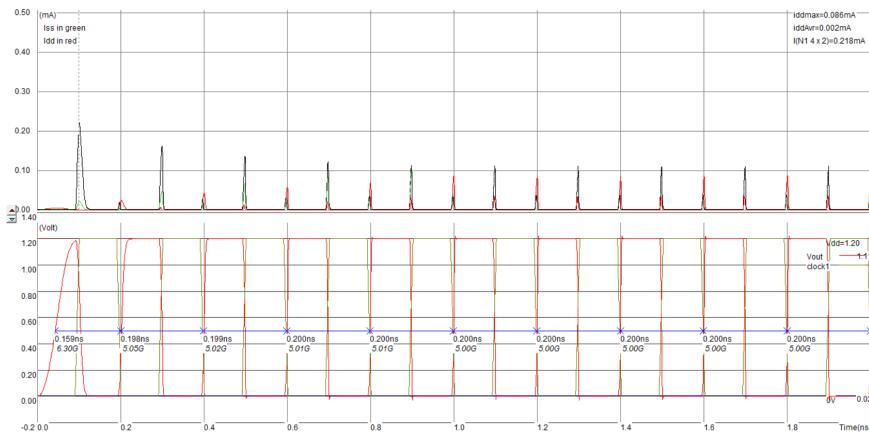


Figure 5: Plot of voltage versus current without a capacitor.

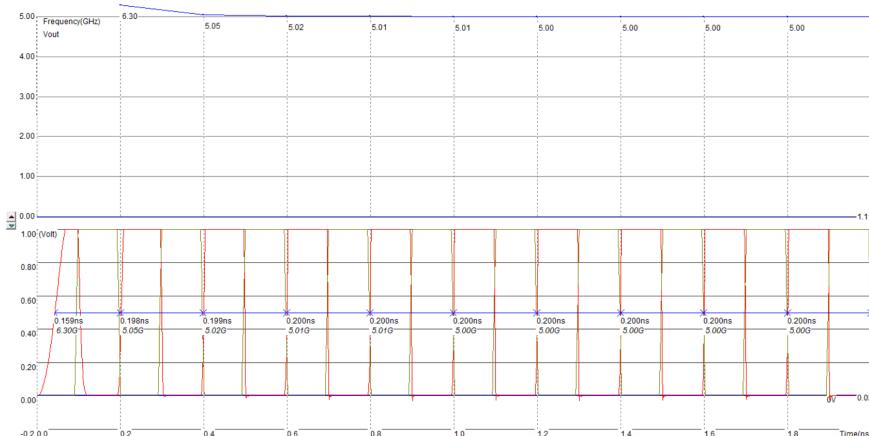


Figure 6: Plot of frequency over time without a capacitor.

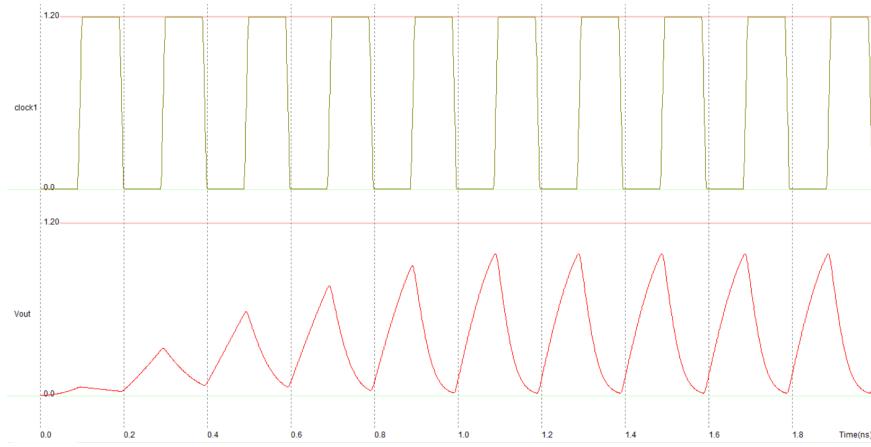


Figure 7: Plot of voltage over time with a capacitor.

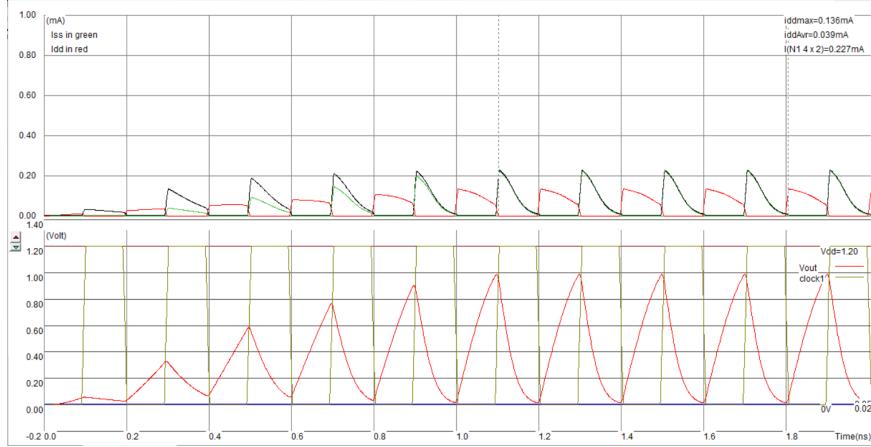


Figure 8: Plot of voltage versus current with a capacitor.

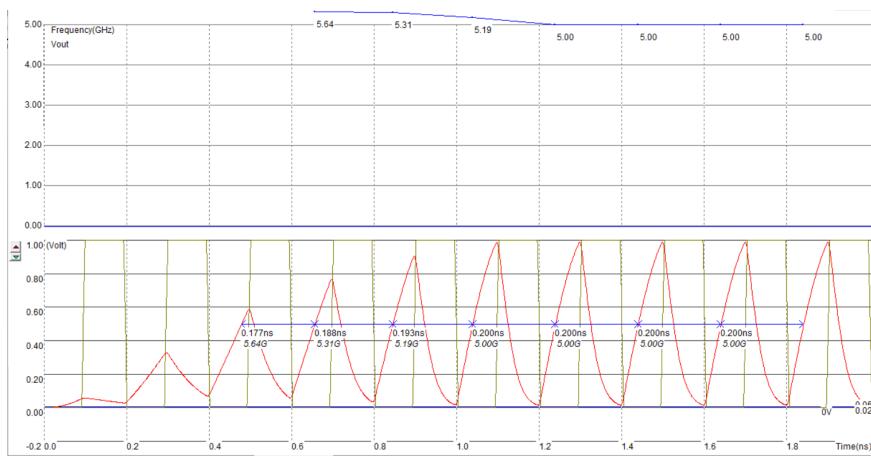


Figure 9: Plot of frequency over time with a capacitor.

Discussion and Conclusion

This experiment involved designing and simulating a CMOS inverter in Microwind to analyze its switching behavior with a clocked input cycling between 0 V and 1.2 V. The simulation results verified the inverter's basic function: the output reliably inverted the input signal. Various waveforms—including voltage versus time, current versus voltage, and frequency versus time—were examined for both unloaded and loaded output conditions.

Adding a load capacitor highlighted the rise and fall times, which stem from the charging and discharging of the output node, consistent with the relation $t_r = 35 \cdot C_{out} \text{ ns}$. In the absence of the capacitor, transitions appeared sharp and nearly ideal; with the capacitor, minor delays and current peaks were observed. These effects reflect fundamental CMOS principles, notably the impact of load capacitance and the necessity of balancing transistor dimensions to offset PMOS and NMOS mobility differences. Adjusting the aspect ratios achieved more symmetrical signal edges, enhancing timing characteristics.

In summary, the study confirmed theoretical expectations and practical behavior of CMOS inverters. It demonstrated that dynamic factors—such as output capacitance, transistor sizing, and carrier mobility—significantly influence propagation delay, switching speed, and power consumption. The findings reinforced the importance of careful design choices for efficient and robust VLSI circuits.

References

- [1] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, 2003.
- [2] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 3rd ed. McGraw-Hill, 2003.
- [3] N. H. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Pearson, 2015.
- [4] GeeksforGeeks, “CMOS Inverter,” *GeeksforGeeks*, Jul. 2025. [Online]. Available: <https://www.geeksforgeeks.org/electronics-engineering/cmos-inverter>

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design

Experiment Date: July 4, 2025,
Submission Date: August 11, 2025

Lab Report 2:
Implementation of NMOS Ratio-less Inverter.

Submitted to	Submitted by
Moloy Kumar Ghosh	
Lecturer	
Dept of ECE, RUET	Md. Tajim An Noor
&	Roll: 2010025
Md. Faysal Ahamed	
Lecturer	
Dept of ECE, RUET	

Contents

Theory	1
Operation	2
Required Tools	2
Circuit Schematic	2
Output	3
Discussion and Conclusion	4
References	5

Implementation of NMOS Ratio-less Inverter

Theory

An inverter is a fundamental digital logic gate that outputs the logical opposite of its input [1]. In transistor-based designs, inverters can be constructed using CMOS technology, which combines PMOS and NMOS transistors, or using NMOS-only approaches [2]. In NMOS logic, both the pull-up and pull-down networks utilize NMOS transistors. Traditionally, a load resistor connects the supply voltage (VDD) to the output node, while an NMOS transistor pulls the output to ground. However, physical resistors are inefficient in terms of area and speed for integrated circuits, especially as circuit complexity increases [3].

To address these limitations, active load NMOS transistors can replace the resistor, resulting in a ratio-less inverter design [4]. This configuration, known as the 3-NMOS ratio-less inverter, uses three NMOS transistors: one as the main driver and two as active loads. The driver transistor receives the input signal at its gate and connects its source to ground. The active load transistors are placed between VDD and the output node, with their gates biased to remain partially on, acting as constant current sources. This approach simplifies fabrication in NMOS technology and eliminates the need for precise transistor sizing ratios, making it suitable for large-scale integration [2].

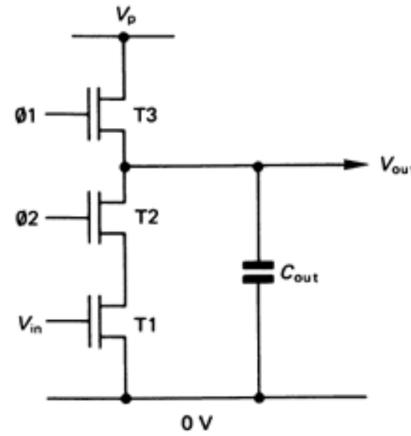


Figure 1: Schematic diagram of the NMOS ratio-less inverter

Operation

1. **Input Low (Logic 0):** When the input is at logic 0, transistor T1 remains OFF. The active load transistors (T2 and T3) conduct sufficient current to raise the output voltage to logic 1.
2. **Input High (Logic 1):** When the input switches to logic 1, T1 turns ON, creating a low-resistance path from the output node to ground. This pulls the output down to logic 0. The active load transistors continue to conduct a small amount of current, resulting in static power consumption.

Required Tools

- Microwind
- MS Word
- L^AT_EX

Circuit Schematic

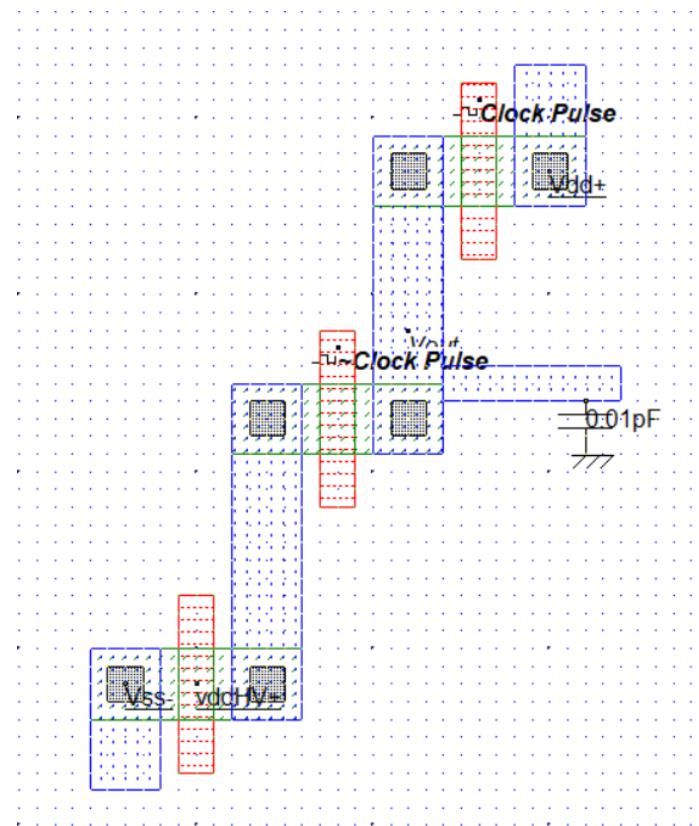


Figure 2: Schematic of the NMOS ratio-less inverter circuit

Output

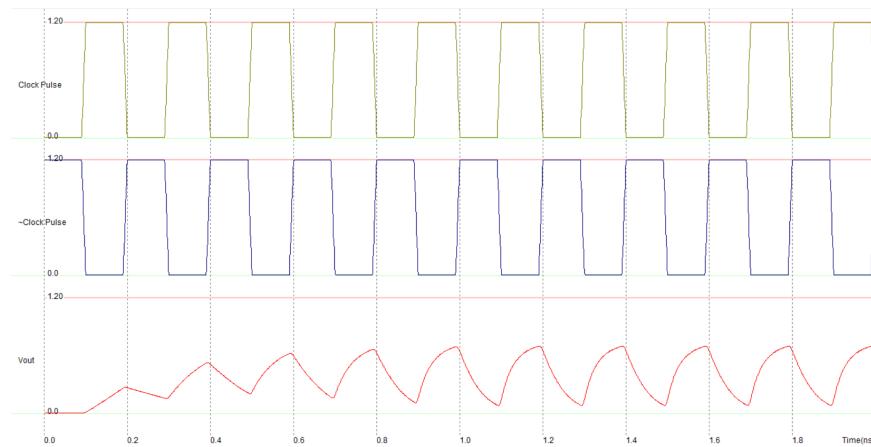


Figure 3: Time-domain voltage response for $V_i n = 1V$.

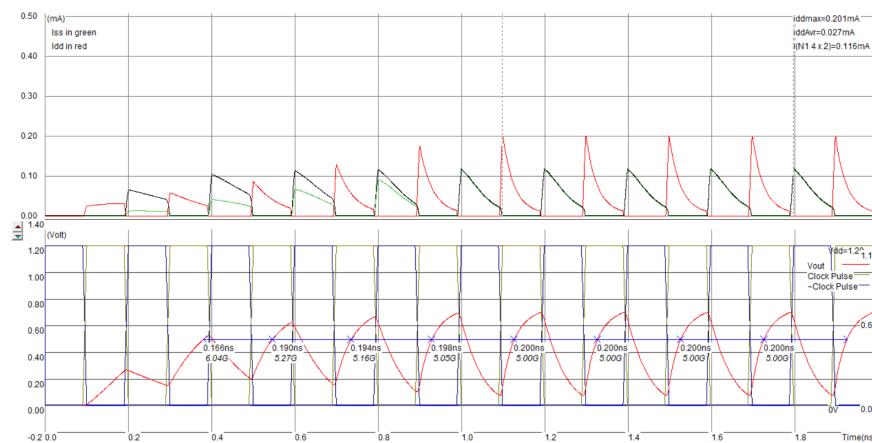


Figure 4: Current-voltage characteristics for $V_i n = 1V$.

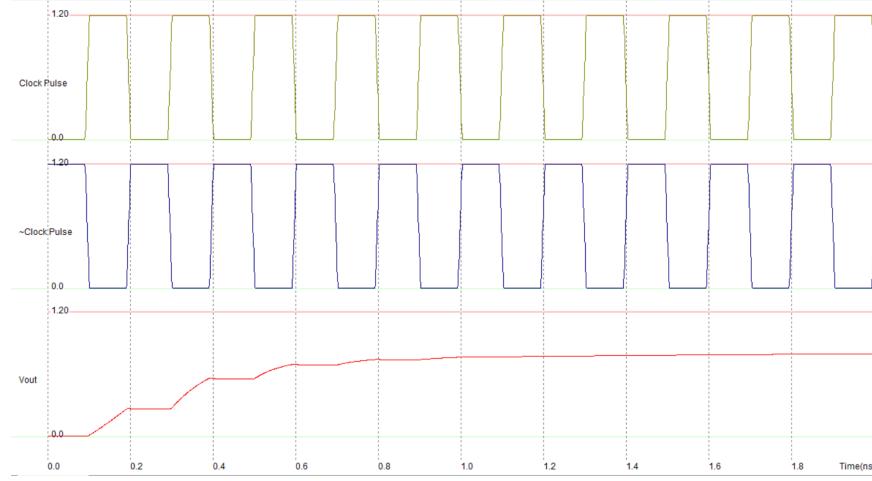


Figure 5: Time-domain voltage response for $V_{in} = 0V$.

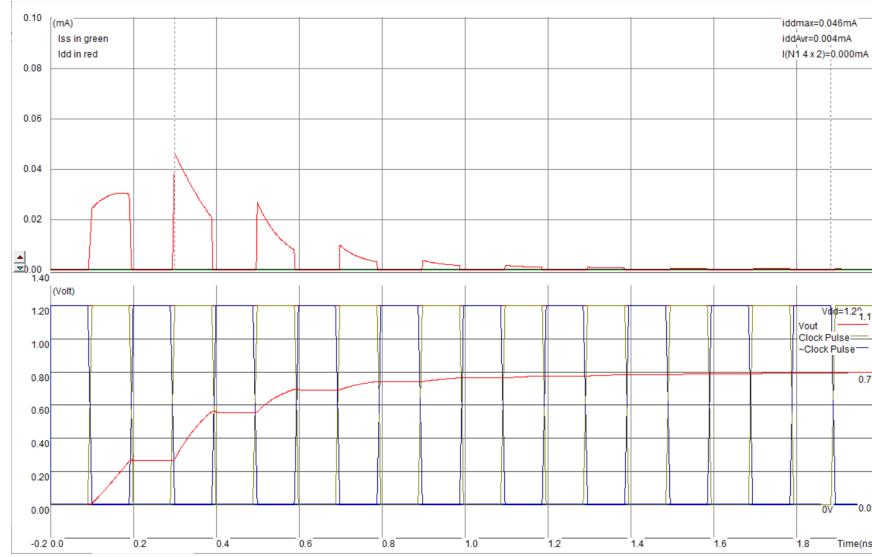


Figure 6: Current-voltage characteristics for $V_{in} = 0V$.

Discussion and Conclusion

In this experiment, a 3-NMOS ratio-less inverter was designed and simulated to analyze its switching characteristics and overall performance. The circuit consisted of one NMOS driver transistor and two NMOS active load transistors, removing the need for resistors. Simulations verified that the output was the logical inverse of the input: when the input was low, the active loads pulled the output high; when the input was high, the driver transistor pulled the output low.

The results indicated that the rise time was slower than the fall time, attributed to the weaker pull-up capability of the active loads. Additionally, static power dissipation was observed during the low output state due to continuous current flow. These

outcomes are consistent with theoretical expectations, demonstrating how transistor biasing and active load design influence switching speed, delay, and power consumption.

A key observation was that the inverter's switching threshold did not depend on transistor sizing ratios, which is a defining feature of ratio-less designs. Eliminating resistors makes the design more compatible with NMOS IC fabrication. Overall, this study reinforced important concepts in transistor operation, active loading, and the trade-offs between speed, power, and simplicity, providing valuable insights for VLSI design.

References

- [1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Pearson, 2015.
- [2] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 3rd ed. McGraw-Hill, 2003.
- [3] S. M. Sze and K. K. Ng, *Physics of Semiconductor Devices*, 3rd ed. Wiley, 2008.
- [4] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, 2003.

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design

Experiment Date: August 8, 2025,
Submission Date: September 15, 2025

Lab Report 3:
Implementation NAND Gate and NOR Gate Using CMOS

Submitted to	Submitted by
Moloy Kumar Ghosh	
Lecturer	
Dept of ECE, RUET	Md. Tajim An Noor
&	Roll: 2010025
Md. Faysal Ahamed	
Lecturer	
Dept of ECE, RUET	

Contents

Theory	1
CMOS Technology Overview	1
CMOS NAND Gate	1
CMOS NOR Gate	2
Advantages of CMOS NAND and NOR Gates	2
Required Tools	2
Circuit Schematic	3
Output	3
Output Analysis	4
Discussion and Conclusion	4
References	5

Implementation NAND Gate and NOR Gate Using CMOS

Theory

Logic gates are fundamental components in digital electronics, enabling the implementation of arithmetic, memory, and control operations within integrated circuits [1]. NAND and NOR gates are particularly important because they are universal gates—meaning any logic function (such as AND, OR, NOT, XOR) can be constructed using only NAND or NOR gates [2]. As a result, understanding the design and operation of CMOS NAND and NOR gates is essential for grasping the foundations of digital VLSI (Very Large Scale Integration) systems [3].

CMOS Technology Overview

CMOS (Complementary Metal-Oxide-Semiconductor) is the predominant technology for digital circuit design, valued for its low power consumption and scalability [1]. CMOS circuits utilize both PMOS and NMOS transistors in a complementary configuration. The structure consists of a pull-up network (PUN) made of PMOS transistors connected to the supply voltage (VDD) and a pull-down network (PDN) of NMOS transistors connected to ground [3]. Inputs are applied to both networks, and the output is taken from their shared node. This arrangement ensures that, ideally, only one network conducts at a time, minimizing static power usage and producing sharp logic transitions.

CMOS NAND Gate

A CMOS NAND gate features two PMOS transistors in parallel for the pull-up network and two NMOS transistors in series for the pull-down network [1]. Its operation is as follows:

- When both inputs are high (logic 1), both NMOS transistors conduct, creating a low-resistance path to ground and pulling the output to logic 0.
- For any other input combination (00, 01, or 10), at least one PMOS transistor conducts in the parallel pull-up network, pulling the output to logic 1.

This configuration matches the NAND gate truth table, with the parallel PMOS arrangement ensuring a strong pull-up except when both inputs are high.

CMOS NOR Gate

The CMOS NOR gate is the logical counterpart to the NAND gate. It uses two PMOS transistors in series for the pull-up network and two NMOS transistors in parallel for the pull-down network [3]. Its operation is:

- When both inputs are low (logic 0), both PMOS transistors conduct, providing a path from VDD to the output and driving it to logic 1.
- For all other input combinations (01, 10, or 11), at least one NMOS transistor conducts in the parallel pull-down network, pulling the output to logic 0.

This setup produces the NOR gate truth table, where the output is high only when both inputs are low.

Advantages of CMOS NAND and NOR Gates

CMOS implementations of NAND and NOR gates offer several benefits. The complementary design results in minimal static power dissipation, as there is no direct path from VDD to ground except during switching [2]. CMOS gates also provide high noise immunity, sharp switching thresholds, and are easily scalable with modern fabrication techniques [1]. Their universality makes NAND and NOR gates essential building blocks for more complex logic circuits, underscoring their importance in VLSI and digital system design [3].

Required Tools

- Microwind
- MS Word
- L^AT_EX

Circuit Schematic

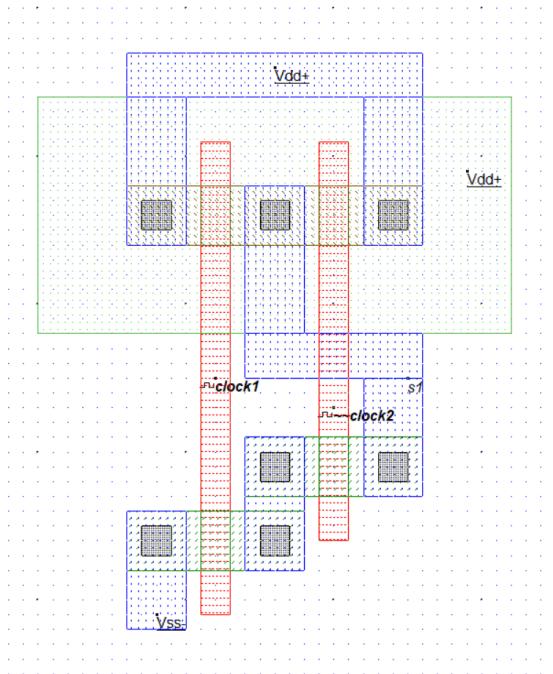


Figure 1: CMOS NAND Gate Schematic

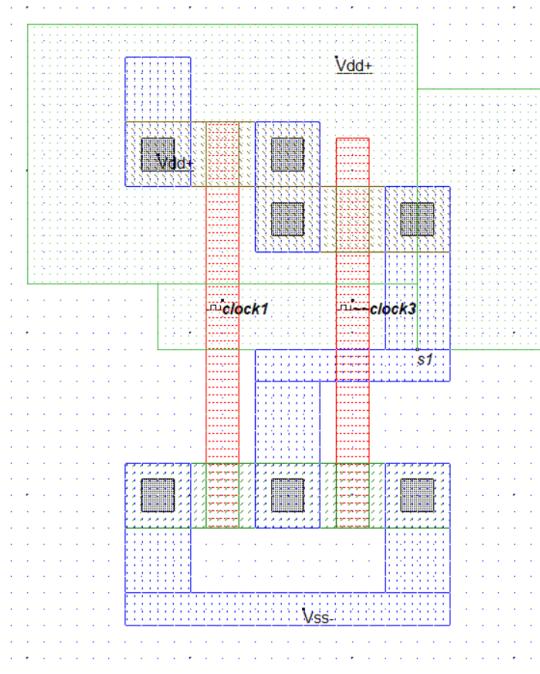


Figure 2: CMOS NOR Gate Schematic

Output

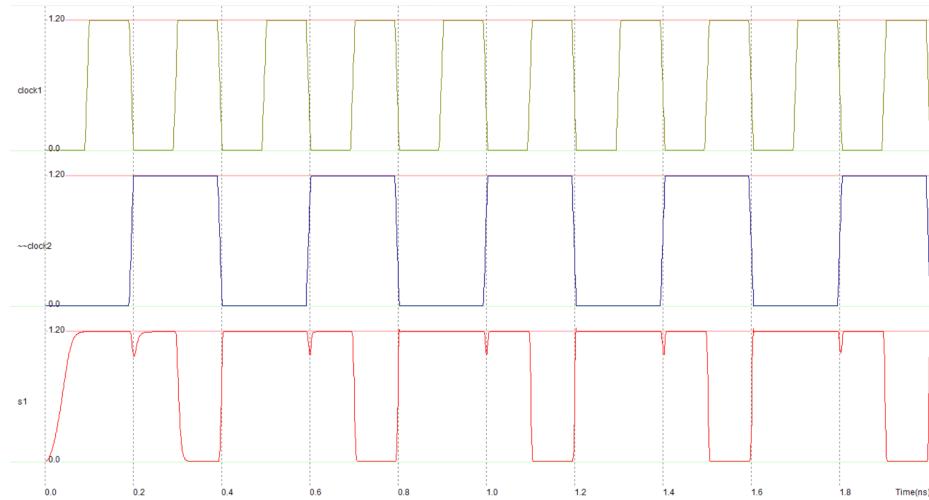


Figure 3: Output Waveform of CMOS NAND Gate

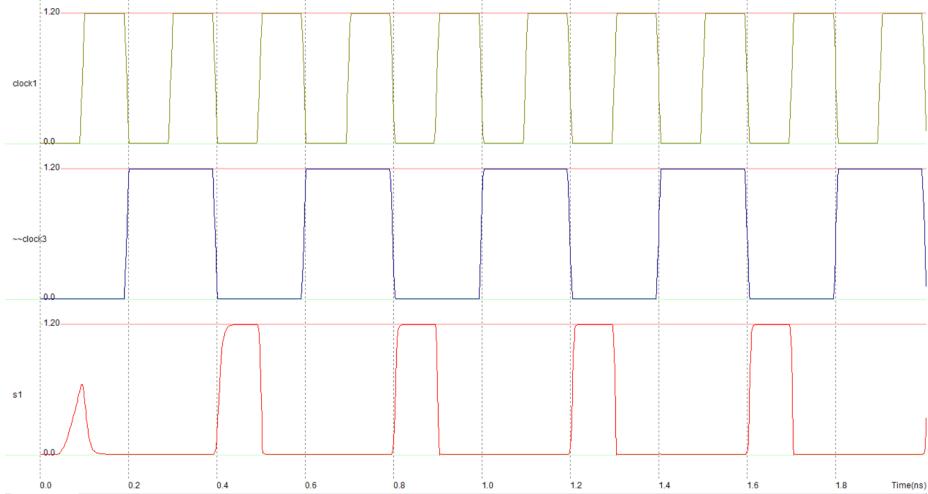


Figure 4: Output Waveform of CMOS NOR Gate

Output Analysis

NOR Gate:

The output signal s_1 is high only when both $clock1$ and $clock3$ are at logic 0. If either input is at logic 1, s_1 transitions low, which is consistent with the NOR gate truth table. The waveform segments correspond to the following input combinations:

- (0,0): $s_1 = 1$
- (1,0), (0,1), (1,1): $s_1 = 0$

NAND Gate:

The output s_1 is low only when both $clock1$ and $clock2$ are at logic 1. For all other input combinations, s_1 remains high, matching the NAND gate truth table. The waveform mapping is as follows:

- (1,1): $s_1 = 0$
- (0,0), (0,1), (1,0): $s_1 = 1$

Discussion and Conclusion

In this experiment, CMOS NAND and NOR gates were implemented and simulated to observe their logical behavior. The simulation results aligned with the theoretical truth tables: the NAND gate produced a low output only when both inputs were high, and the NOR gate generated a high output only when both inputs were low. These outcomes verified the correct functioning of the complementary pull-up and pull-down networks in CMOS logic circuits.

The simulation further demonstrated the efficiency of CMOS technology. Static power dissipation was minimal, as only one network (either pull-up or pull-down) was active

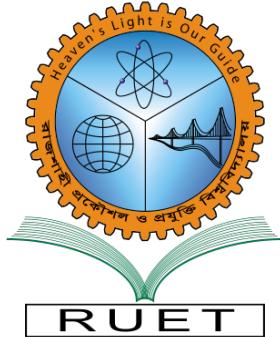
at any given time, with transient power consumption occurring during input transitions. The configuration of PMOS and NMOS transistors—whether in series or parallel—determined both the logic operation and the robustness of output transitions.

In summary, the experiment confirmed the theoretical foundations of CMOS logic design, highlighting its advantages in low power consumption, noise immunity, and scalability. It also underscored the significance of NAND and NOR gates as universal elements for constructing complex digital systems.

References

- [1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Pearson, 2015.
- [2] R. K. Manohar, *Digital Logic Design*. Oxford University Press, 2017.
- [3] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 4th ed. McGraw-Hill Education, 2019.

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design Sessional

Experiment Date: September 9, 2025,
Submission Date: October 12, 2025

Lab Report 4:
Design and Observe the Characteristics Curve of CMOS Circuit

Submitted to	Submitted by
Md. Faysal Ahamed Assistant Professor Dept of ECE, RUET & Moloy Kumar Ghosh Lecturer Dept of ECE, RUET	Md. Tajim An Noor Roll: 2010025

Contents

Task	1
Theory	1
Used Tools	3
Circuit Schematic in Microwind	3
Output	4
Output Analysis	5
Discussion	6
Conclusion	6
References	6

Design and Observe the Characteristics Curve of CMOS Circuit

Task

Design and observe the characteristics curve of CMOS circuits using the following equations:

1. **Equation I:** $Y = \overline{(A + B) \cdot C}$
2. **Equation II:** $Y = A\overline{B} + ABC$

For each equation:

- Design the corresponding CMOS logic gate schematic.
- Simulate the circuit to obtain the output characteristics curve.
- Analyze the output waveform and discuss the behavior of the circuit.

Theory

Complex CMOS logic gates implement Boolean functions in a single stage, offering greater efficiency compared to constructing the same function from multiple NAND or NOR gates [1]. These gates are composed of a pull-up network (PUN) using PMOS transistors and a pull-down network (PDN) using NMOS transistors. The PDN provides a path to ground (V_{SS}) when the output should be logic '0', while the PUN connects the output to the supply voltage (V_{DD}) when the output should be logic '1'. The PUN is the logical dual of the PDN, meaning series connections in one correspond to parallel connections in the other [2].

Equation I: $Y = \overline{(A + B) \cdot C}$ This equation describes a 2-1 AND-OR-Invert (AOI21) gate [1].

- The pull-down network implements the logic $(A + B) \cdot C$, which consists of two parallel NMOS transistors (for A and B) in series with a third NMOS transistor (for C) [2].

- The complementary pull-up network implements $(\bar{A} \cdot \bar{B}) + \bar{C}$, realized by two series PMOS transistors (for A and B) in parallel with a third PMOS transistor (for C) [1].

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

Table 1: Truth Table for $Y = \overline{(A + B) \cdot C}$

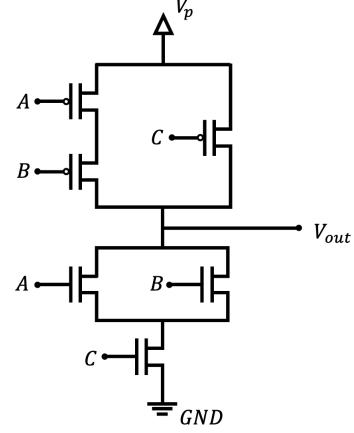


Figure 1: Connection diagram of the CMOS circuit for $Y = \overline{(A + B) \cdot C}$

Equation II: $Y = A\bar{B} + ABC$

- The pull-down network (PDN) implements the logic for (after simplification) $\bar{A} + B\bar{C}$. This requires one NMOS transistor for \bar{A} (A input inverted), and a branch with NMOS transistors for $B\bar{C}$ (B in series with C inverted), both branches connected in parallel [1, 2].
- The pull-up network (PUN) implements the dual logic, which for $Y = A\bar{B} + ABC$ is $A(\bar{B} + C)$. This is realized by two PMOS transistors for B and C in parallel, both in series with a PMOS transistor for A [1].
- The circuit implements the function $Y = \bar{A} + B\bar{C}$, which can be realized efficiently using CMOS logic [2].

Table 2: Truth Table for $Y = A\bar{B} + ABC$

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

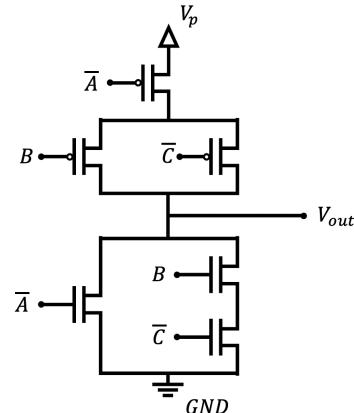


Figure 2: Connection diagram of the CMOS circuit for $Y = A\bar{B} + ABC$

Used Tools

- Microwind
- MS Word
- MS PowerPoint
- L^AT_EX

Circuit Schematic in Microwind

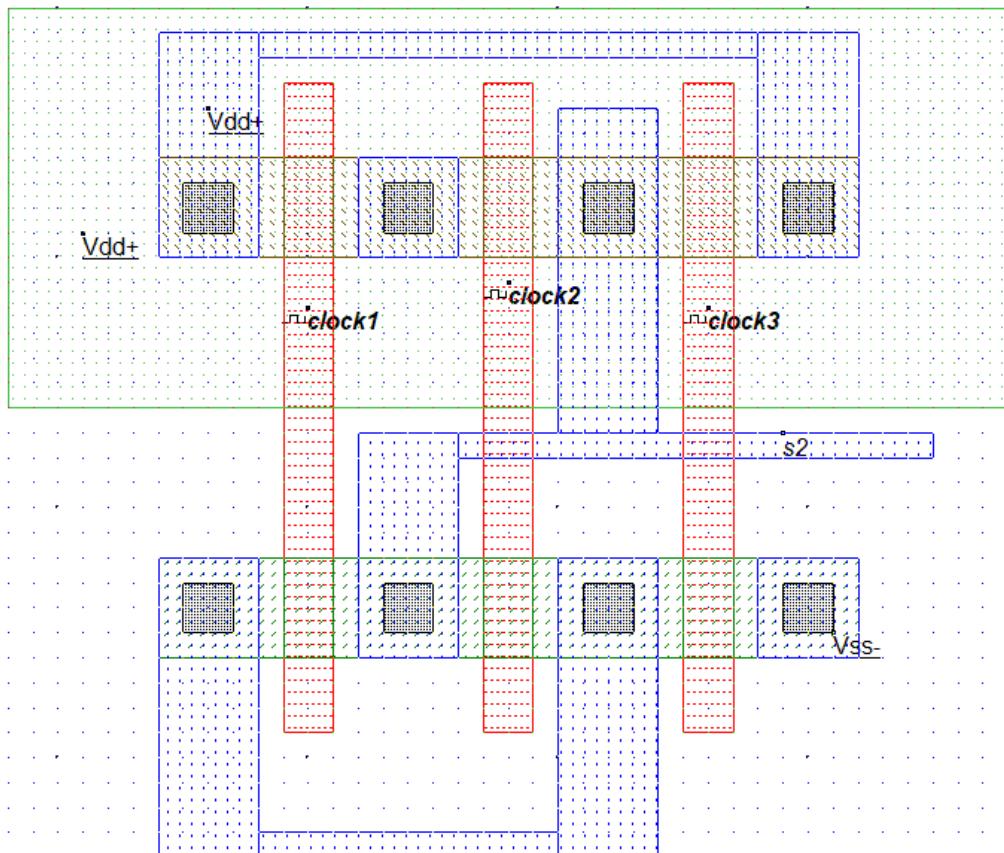


Figure 3: Connection diagram of the CMOS circuit for $Y = \overline{(A + B)} \cdot C$

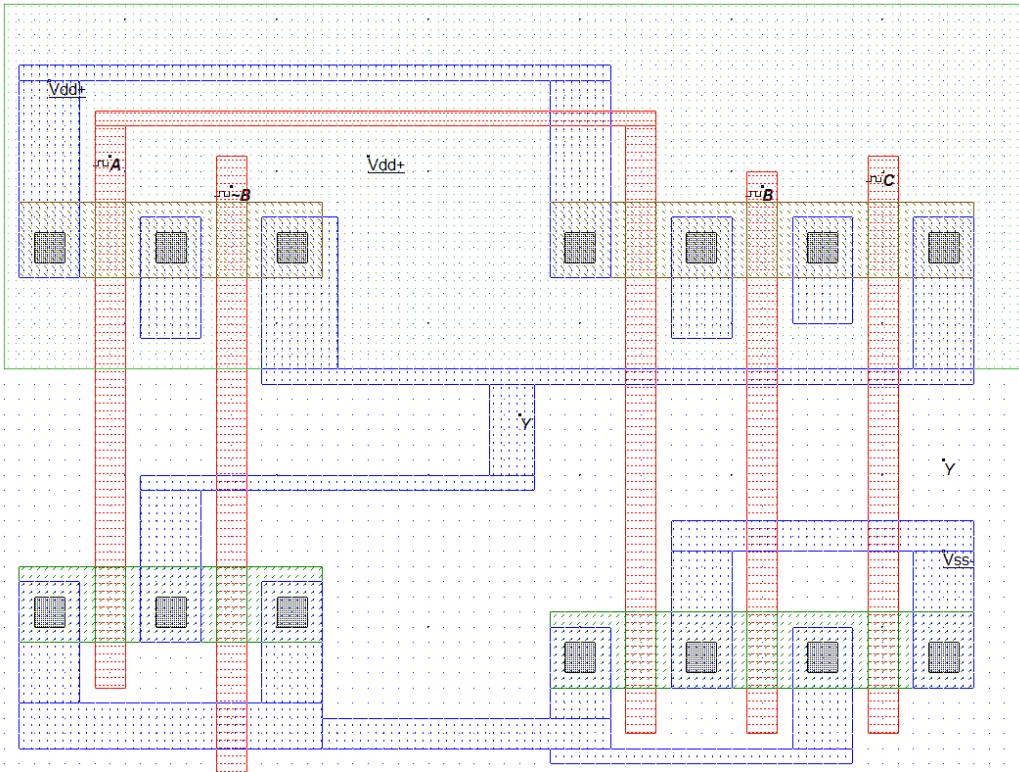


Figure 4: Connection diagram of the CMOS circuit for $Y = A\bar{B} + ABC$

Output

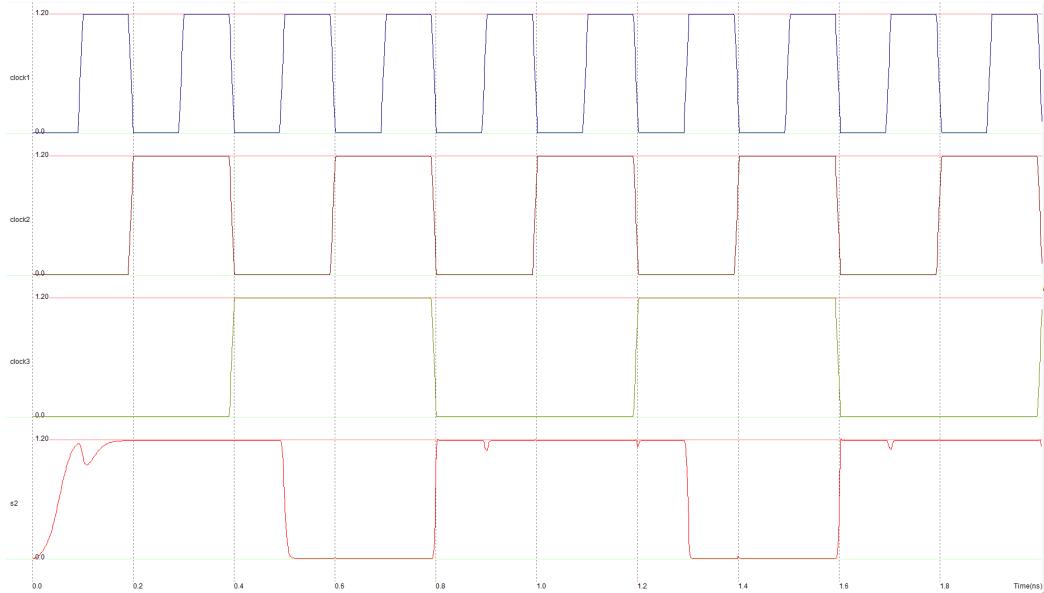


Figure 5: Output Waveform of $Y = \overline{(A + B)} \cdot C$

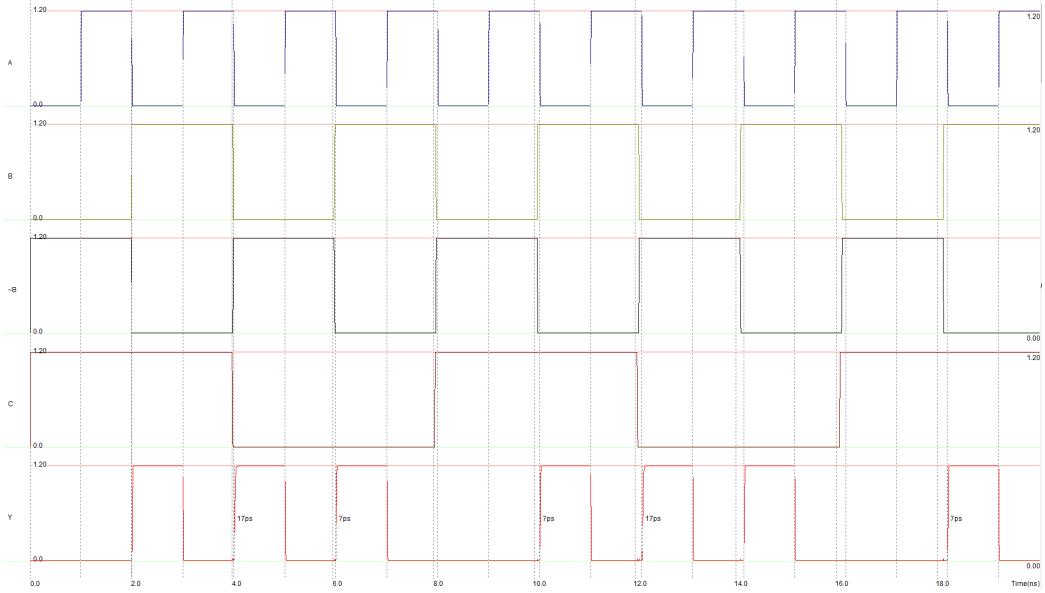


Figure 6: Output Waveform of $Y = A\bar{B} + ABC$

Output Analysis

The complex CMOS circuits were implemented and simulated using Microwind. The resulting output waveforms align with the expected logical behavior as defined by the truth tables for each equation.

Case I: $Y = \overline{(A + B) \cdot C}$

- When $C = 0$, the output remains HIGH (1) for all values of A and B .
- When $C = 1$ and either $A = 1$ or $B = 1$, the output switches to LOW (0).
- When $C = 1$ and both $A = 0$ and $B = 0$, the output is HIGH (1).
- Simulation results confirm that the output is LOW only for input combinations (A,B,C) of $(0,1,1)$, $(1,0,1)$, and $(1,1,1)$, consistent with the truth table.

Case II: $Y = A\bar{B} + ABC$

- When $A = 1$ and $B = 0$, the output is HIGH (1), regardless of C .
- When $A = 1$, $B = 1$, and $C = 1$, the output is HIGH (1).
- For all other input combinations, the output remains LOW (0).
- The simulation confirms that the output is HIGH only for (A,B,C) values of $(1,0,0)$, $(1,0,1)$, and $(1,1,1)$, matching the truth table.

Discussion

This experiment explored the design and simulation of complex CMOS logic gates at the transistor level. By creating complementary pull-up (PMOS) and pull-down (NMOS) networks, complex Boolean functions such as AOI and OAI can be implemented in a single logic stage. This method is typically more efficient and faster than building the same function from multiple universal gates like NAND or NOR. The simulated output waveforms in Microwind closely matched the expected results from the truth tables, confirming both the dynamic and static behavior of the circuits.

Conclusion

To summarize, the experiment confirmed the correct operation of complex CMOS circuits for the specified Boolean equations using Microwind. The observed results validated the design approach of using complementary pull-up and pull-down networks. This reinforces that complex logic functions can be directly and efficiently realized in CMOS technology, which is essential for integrated circuit design.

References

- [1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Pearson, 2010.
- [2] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, 2003.

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design Sessional

Experiment Date: October 12, 2025,
Submission Date: October 20, 2025

Lab Report 5:
Study and Simulation of 1-Bit SRAM Using Microwind

Submitted to	Submitted by
Md. Faysal Ahamed Assistant Professor Dept of ECE, RUET & Moloy Kumar Ghosh Lecturer Dept of ECE, RUET	Md. Tajim An Noor Roll: 2010025

Contents

Theory	1
Used Tools	2
Circuit Schematic in Microwind	3
Output	3
Output Analysis	4
Discussion	4
Conclusion	4
References	4

Study and Simulation of 1-Bit SRAM Using Microwind

Theory

A 1-bit SRAM cell is most often realised with a 6-transistor (6T) topology: two cross-coupled CMOS inverters create a bistable latch that holds complementary storage nodes Q and \bar{Q} . Two NMOS access transistors, enabled by the word line WL , connect the latch to the differential bit-lines BL and \bar{BL} [1, 2].

- Hold: With $WL = 0$ the access transistors are off and the latch preserves its state by positive feedback.
- Write: With $WL = 1$ forcing $BL/\bar{BL} = 0/1$ drives the cell to store a logical ‘0’ at Q ; conversely forcing $BL/\bar{BL} = 1/0$ writes a ‘1’.
- Read: Both bit-lines are precharged high. Asserting $WL = 1$ weakly discharges the bit-line tied to the internal ‘0’ node, producing a small differential ΔV that a sense amplifier detects; ideally the stored state is not destroyed (practical read behaviour depends on sizing and device nonidealities) [3].

Authenticity check: the above description and operation modes are consistent with standard textbook treatments of SRAM fundamentals. Note that the behavioural update equations below are an idealised logical abstraction and do not capture analog effects (finite drive, threshold, body effect, leakage, or read-disturb) which must be evaluated by transistor-level simulation [1, 2].

Idealised behavioural update equations (logical model):

$$\begin{aligned} Q(t+1) &= (WL \cdot BL) + (\bar{WL} \cdot Q(t)), \\ \bar{Q}(t+1) &= (WL \cdot \bar{BL}) + (\bar{WL} \cdot \bar{Q}(t)), \end{aligned}$$

where \cdot denotes logical AND and $+$ denotes logical OR. During a read (both bit-lines precharged) the hold term typically dominates.

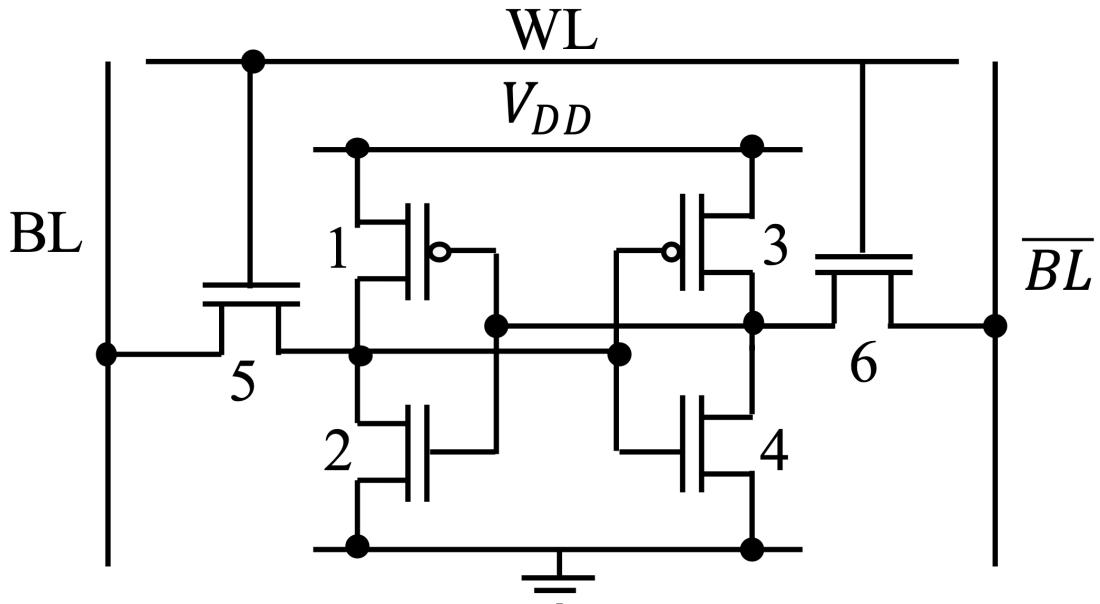


Figure 1: Circuit diagram of 1 Bit SRAM cell

WL	BL	\overline{BL}	Before (Q)	Action	After (Q)
0	X	X	0 or 1	Hold (cell isolated)	Unchanged
1	1	0	0 or 1	Write '1' (drive nodes)	1
1	0	1	0 or 1	Write '0' (drive nodes)	0
1	1	1*	0 or 1	Read (bitlines precharged)	Unchanged (sense ΔV)
1	0	0	0 or 1	Invalid / weak read (both BLs low)	No reliable change

* precharged.

Table 1: SRAM 6T cell operation summary

Used Tools

- Microwind
- MS Word
- MS PowerPoint
- L^AT_EX

Circuit Schematic in Microwind

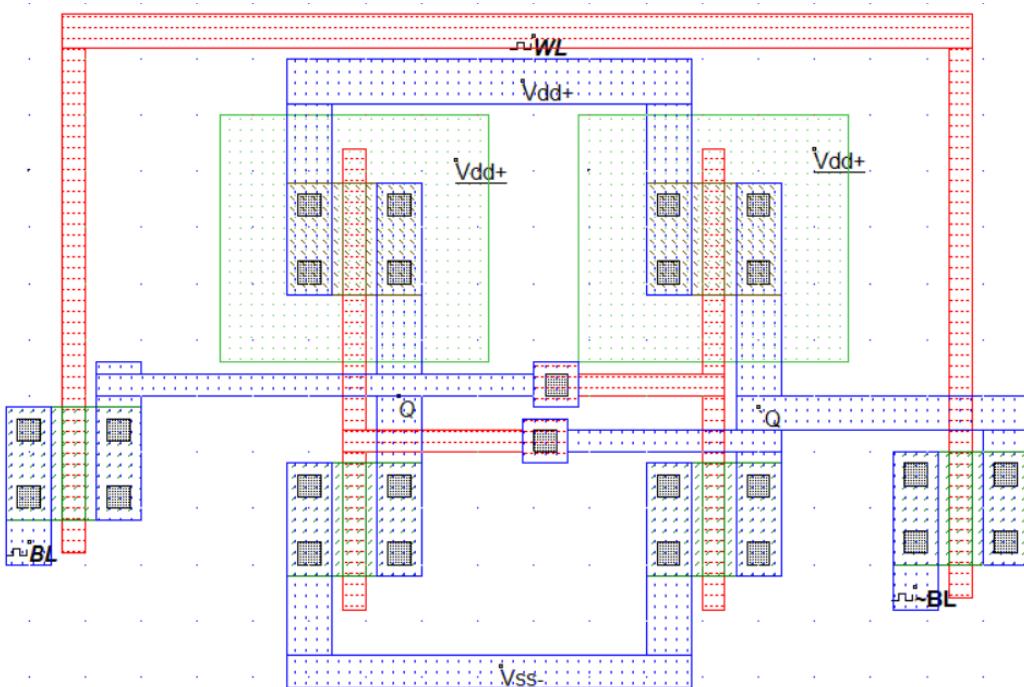


Figure 2: Connection Diagram of 1 Bit SRAM cell in Microwind

Output

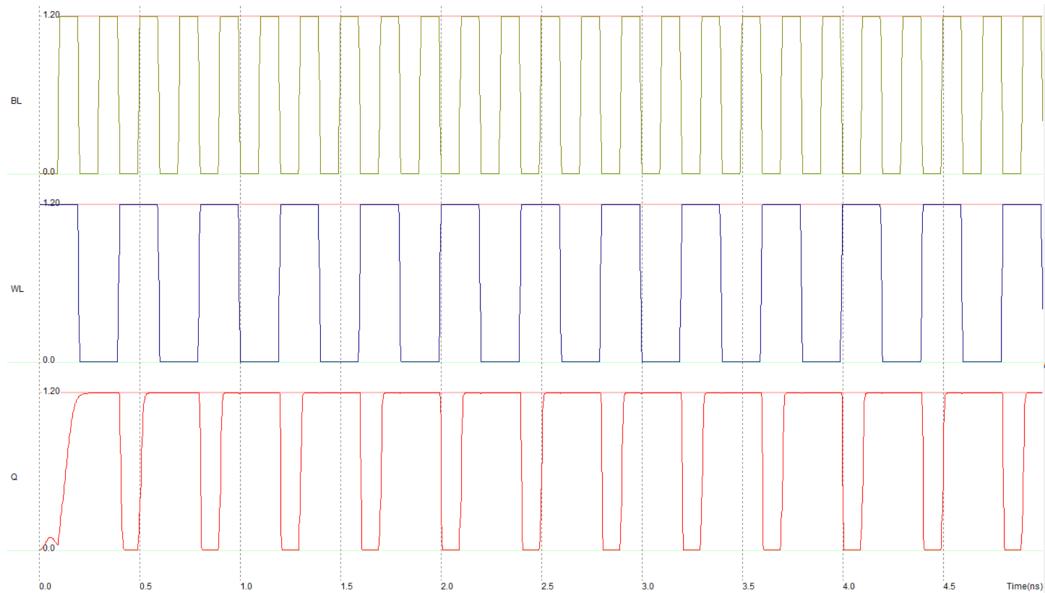


Figure 3: Output of the designed 1 Bit SRAM cell in Microwind

Output Analysis

The Microwind transient simulation confirms correct 1-bit SRAM operation:

- With $WL = 0$ the cell holds its previous state (Q remains constant).
- With $WL = 1$ and $BL/\overline{BL} = 1/0$, Q is written to ‘1’; with $BL/\overline{BL} = 0/1$, Q is written to ‘0’.
- During a read with precharged bit-lines, asserting WL produces a small differential on the bit-lines consistent with the stored value and does not corrupt the cell.

Key observations:

- Write-0 is faster than write-1 due to asymmetric drive strengths.
- Q may not reach V_{DD} within the WL pulse window, which can limit write-1 and read margins.
- Reads produce only a small BL differential, requiring sensitive sensing and careful transistor sizing.

Discussion

The results show that cell retention is provided by the positive feedback loop of the cross-coupled inverters, while the word-line gated access transistors permit external forcing during write operations and only lightly load the cell during reads. Correct sizing of devices is critical: access transistors must be large enough to reliably overwrite the latch during a write but small enough to avoid compromising the stored value during a read (preserving static noise margin). A symmetric layout and matched inverter device dimensions help equalize bit-line loading and improve read margin. The transient waveforms follow the expected timing — Q is driven by BL while WL is asserted and remains stable once WL is released — confirming the 6T cell’s functional behaviour.

Conclusion

A 1-bit, 6-transistor SRAM cell was implemented and simulated in Microwind. Read, write and hold operations were verified both conceptually and by transient waveform inspection, demonstrating correct operation. The study underlines important design choices for robust SRAM: appropriate transistor sizing, proper bit-line precharge and equalization, and use of differential sensing to maximize read reliability and scalability.

References

- [1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Boston, MA: Addison-Wesley, 2010.
- [2] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- [3] A. D. Calhoun and A. P. Chandrakasan, “(please replace) detailed reference for the calhoun & chandrakasan source cited in the text,” XXXX, replace with full bibliographic details (conference/journal, volume, pages, DOI).

Heaven's Light is Our Guide
Rajshahi University of Engineering and Technology



Course Code
ECE 4128

Course Title
VLSI Design Sessional

Experiment Date: October 20, 2025,
Submission Date: October 29, 2025

Lab Report 6:
Implementation of Half Adder Circuit Using CMOS in Microwind

Submitted to	Submitted by
Md. Faysal Ahamed Assistant Professor Dept of ECE, RUET & Moloy Kumar Ghosh Lecturer Dept of ECE, RUET	Md. Tajim An Noor Roll: 2010025

Contents

Theory	1
CMOS implementation	2
Used Tools	3
Circuit Schematic in Microwind	3
Output	4
Output Analysis	4
Discussion	4
Conclusion	5
References	5

Implementation of Half Adder Circuit Using CMOS in Microwind

Theory

A half adder is a combinational logic circuit that adds two single-bit binary numbers A and B and produces two outputs: Sum (S) and Carry (C). It performs the least-significant-bit addition without an incoming carry. The signals have the following meanings:

- A, B: single-bit binary inputs to be added.
- Sum (S): the least-significant bit of the addition result.
- Carry (C): the carry-out bit (1 if the sum exceeds 1 and must be carried to the next higher bit).

The half-adder logic and practical CMOS realizations are discussed in standard texts on digital and VLSI design[1, 2, 3].

Boolean expressions

$$S = A \oplus B = A\bar{B} + \bar{A}B$$

$$C = A \cdot B$$

These expressions and their transistor-level mappings (XOR implemented via complementary networks or transmission gates, AND via NAND+inverter or static AND) are treated in detail in the cited books[2, 1].

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table of Half Adder (definition after [3])

Logic (gate-level) circuit:

- Sum: XOR gate taking inputs A and B.
- Carry: AND gate taking inputs A and B.

Both inputs A and B feed the XOR (for S) and the AND (for C) so the block diagram is the two inputs branching to these two gates.

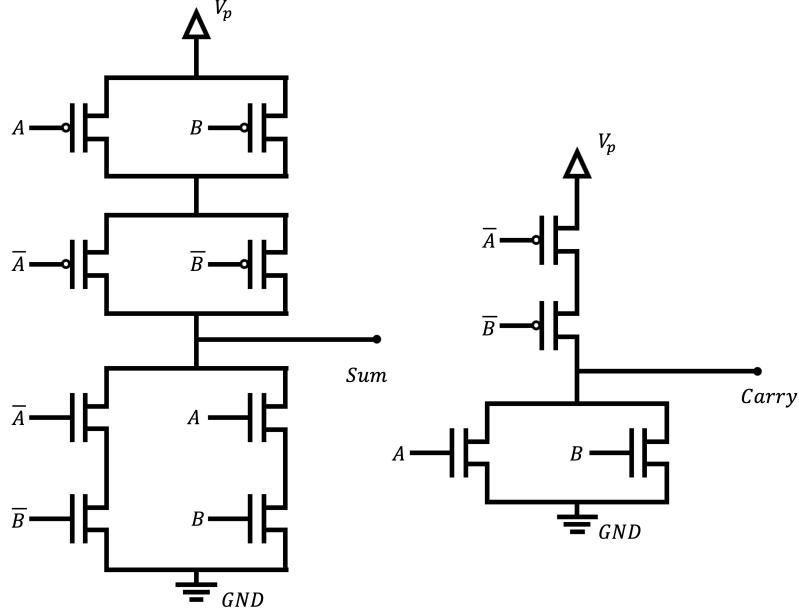


Figure 1: CMOS Circuit Diagram of Half Adder

CMOS implementation[4]:

CMOS implementation

Sum (XOR)

$$S = A \oplus B = \overline{A}B + A\overline{B}$$

Complement (output low condition), obtained by De Morgan:

$$\overline{S} = \overline{\overline{A}B + A\overline{B}} = (A + \overline{B})(\overline{A} + B) = AB + \overline{A}\overline{B} \quad (\text{XNOR})$$

Implementation mapping:

- nMOS pull-down network (implements \overline{S}): two parallel branches — one branch is nMOS(A) in series with nMOS(B); the other branch is nMOS(\overline{A}) in series with nMOS(\overline{B}). (If complementary signals \overline{A} , \overline{B} are not available, provide inverters or realize XOR with transmission gates.)
- pMOS pull-up network (dual of \overline{S}): series combination of two parallel pairs, i.e. $(pA \parallel p\overline{B})$ in series with $(p\overline{A} \parallel pB)$. This network pulls the output high when $\overline{S} = 0$.

Carry (AND)

Equation for Carry (Pull-up network):

$$C = A \cdot B$$

Complement (for the nMOS pull-down network), by De Morgan:

$$\overline{C} = \overline{A \cdot B} = \overline{A} + \overline{B}$$

Used Tools

- Microwind
- MS Word
- MS PowerPoint
- L^AT_EX

Circuit Schematic in Microwind

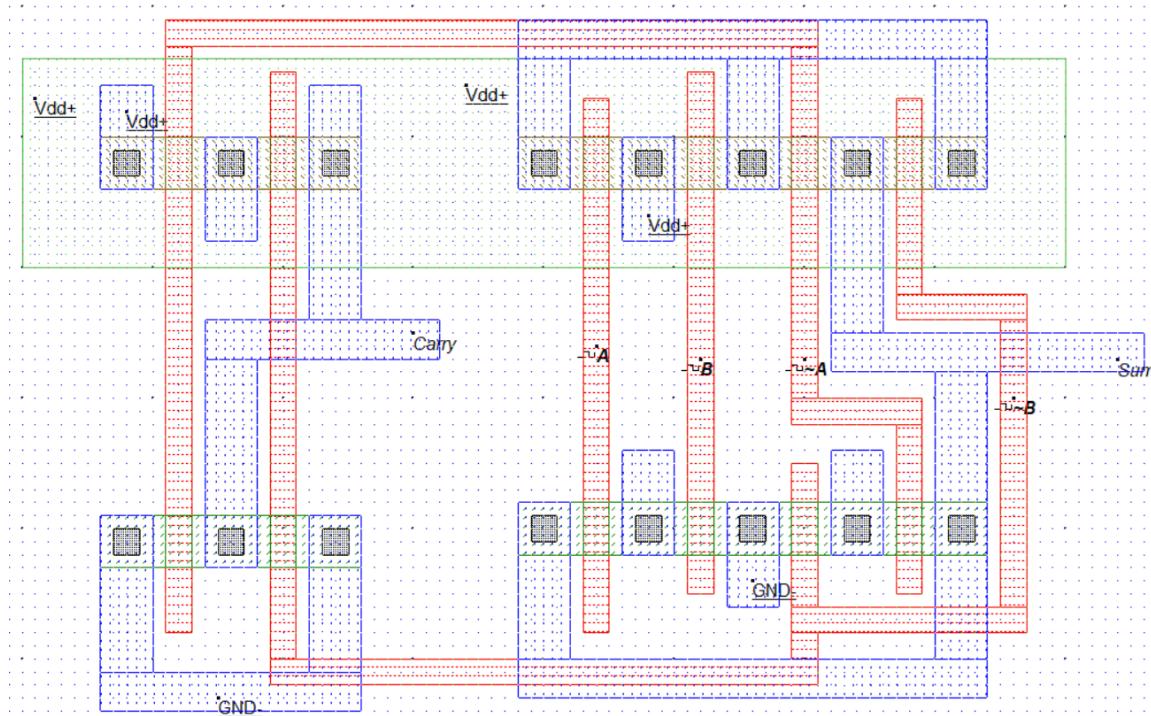


Figure 2: Half Adder Circuit Schematic in Microwind

Output

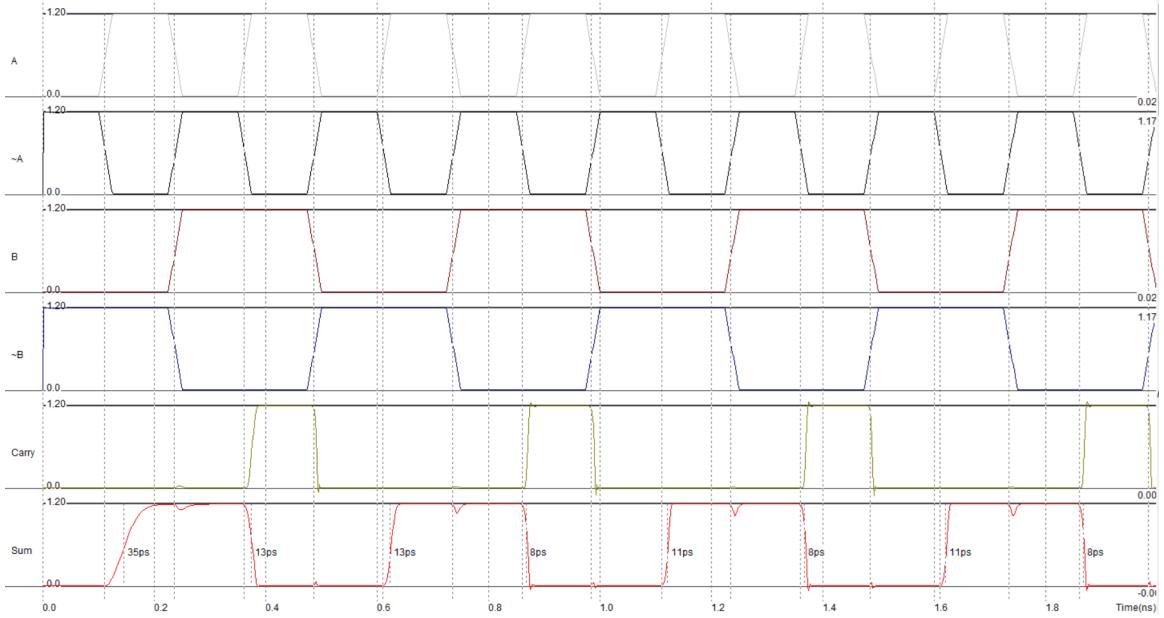


Figure 3: Sum & Carry Output Waveform of Half Adder Circuit

Output Analysis

1. Testbench and signal summary
 - Supply: $VDD \approx 1.20$ V (top trace marker).
 - Inputs: A and B driven with periodic pulses (top traces).
 - Outputs: Carry (green) and Sum (red). Time base in the plot is nanoseconds; zoomed markers show pulses spaced 0–2 ns.
2. Logic-level correctness
 - Carry ($C = A \cdot B$): goes high (1.20 V) only when both A and B are high; remains near 0 V otherwise. Example: cycle near 0.3–0.45 ns both inputs high and Carry rises to VDD.
 - Sum ($S = A \oplus B$): high when exactly one input is high, low when inputs are equal (both 0 or both 1). Sum falls to 0 when both inputs are high while Carry goes high, matching expected half-adder outputs.

Discussion

The implemented half adder ($S = A \oplus B$, $C = A \cdot B$) was simulated in Microwind and matches the truth table: Carry rises only when A and B are high and Sum follows XOR behavior. Waveforms show finite propagation delays and non-ideal rise/fall

times — the XOR (Sum) path is slower due to a larger transistor network, and PMOS devices were widened to better balance rise/fall. Dynamic switching dominates power; buffering the outputs with inverters and using a carefully sized static complementary or transmission-gate XOR reduces delay and voltage degradation.

Overall the simulation confirms a correct CMOS half adder implementation while highlighting the usual tradeoffs between speed, area (transistor count), and power.

Conclusion

A CMOS half adder was implemented and simulated in Microwind. The circuit produced the expected Sum and Carry waveforms for all input combinations, demonstrating correct logical operation. The exercise illustrates important design considerations for CMOS combinational circuits: choose transistor sizes to balance speed and noise margin, use buffering for improved drive and signal swing, and evaluate power/delay tradeoffs. Future work could measure propagation delay and energy per operation quantitatively, optimize the XOR topology for speed or area, and extend the design to a full adder and multi-bit adder chains.

References

- [1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Boston, MA: Addison-Wesley, 2011.
- [2] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- [3] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. New York, NY: Oxford University Press, 2015.
- [4] “Implement half adder circuit using static CMOS.” Oct. 2025, [Online; accessed 23. Oct. 2025]. [Online]. Available: <https://www.ques10.com/p/36440/implement-half-adder-circuit-using-static-cmos>