CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Electronics and Telecommunication Engineering

VLSI Technology Sessional

ETE 404

Experiment No:07

Schematic Driven Layout Design of a NAND Gate Using Virtuoso Layout Suite Editor XL

Submitted by:

Shamanza Chowdhury

ID: 1908008

Submitted to:

Arif Istiaque Rupom

Lecturer

Dept. of ETE,CUET

October 6, 2024

1 Objectives:

- To familiarize with schematic-driven layout design.
- To perform schematic-level verification, DRC and LVS.
- To perform post-layout simulation of NAND Gate.

2 Apparatus:

- **Software:** VMware Workstation v10
- **Hardware:** PC

3 Design Process:

3.1 Schematic Entry:

- A new cell view named ”nand2x1” was created in the CIW to intiate the creation of the schematic for a 2-input NAND gate.
- Proper connections were made based on the logic of the NAND gate, ensuring accurate input-output relations.The schematic diagram of a NAND gate is given below:

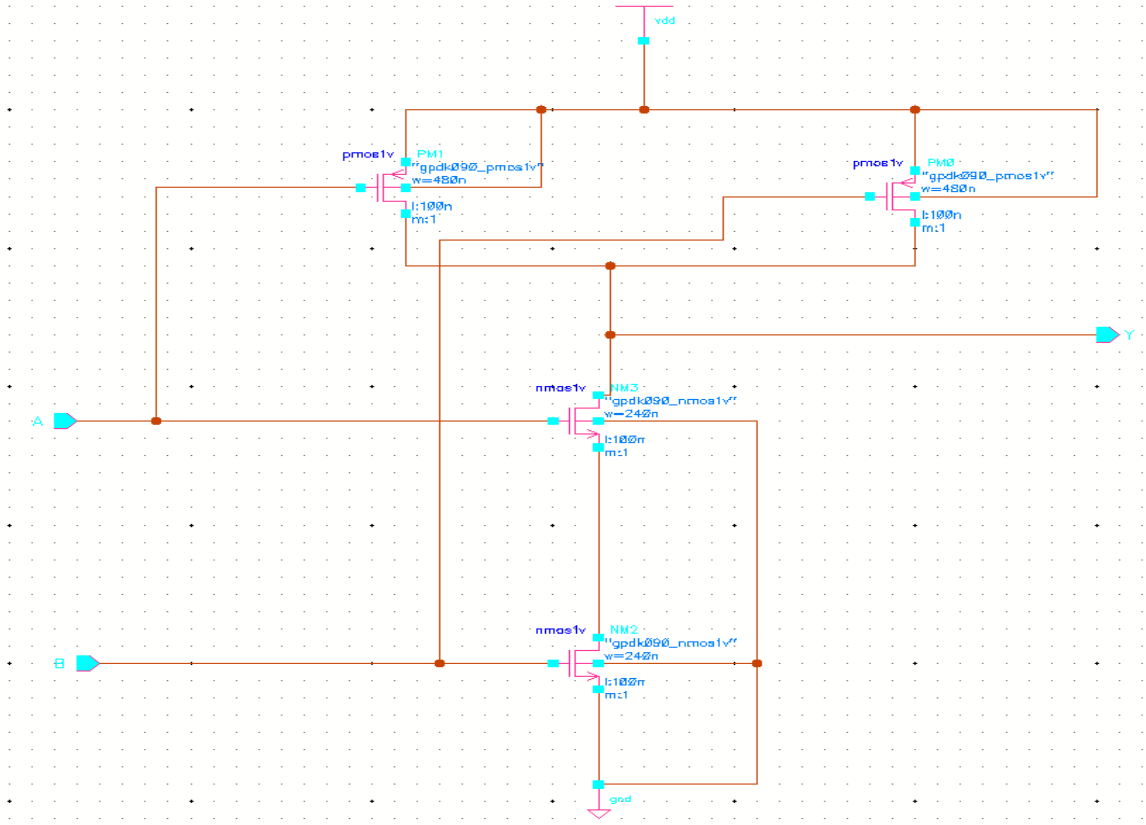

Figure 01: Schematic diagram of an NAND gate.

3.2 Schematic-Level Simulation:

- Analog analysis was performed by launching the ADE L simulation environment.
- The stimuli for inputs A and B had been configured such that the pulse width and period of input B were set to be half of those for input A .
- The simulation had been executed, and the expected output waveform had been generated, showing the all four input combinations (00, 01, 10, 11).The output of an AND gate is given below:

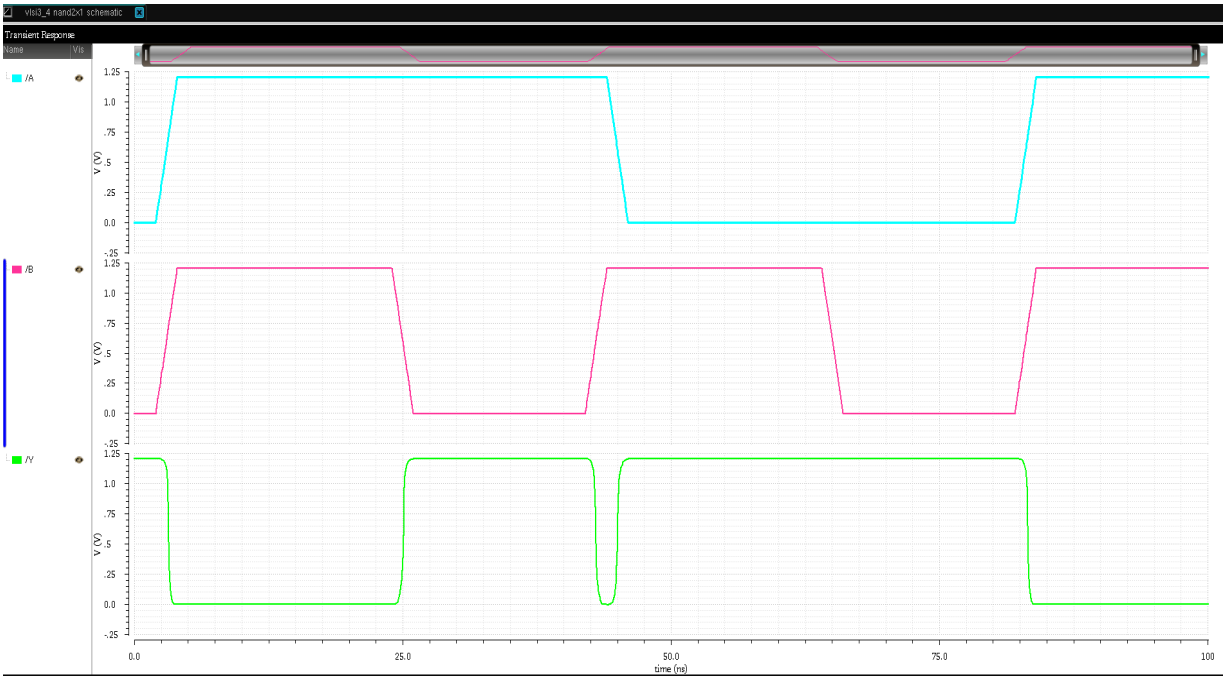

Figure 02:Output waveform of an NAND gate.

- A small glitch(voltage dip) appears at the output of the NAND gate due to unequal delays in the two input paths. When input A transitions from high to low, input B takes some time to switch to high because of the inverter delay. As input A falls but remains above the NMOS threshold of 1V, the NMOS transistor connected to A stays on. Meanwhile, input B has already risen above 1V, turning the NMOS transistor connected to B on as well. During this overlap, both NMOS transistors conduct briefly, causing a glitch at the NAND gate's output.

3.3 Symbol Creation:

- A symbol was generated from the newly created NAND gate schematic to facilitate its use in higher-level designs. The symbol of a NAND gate is given below:

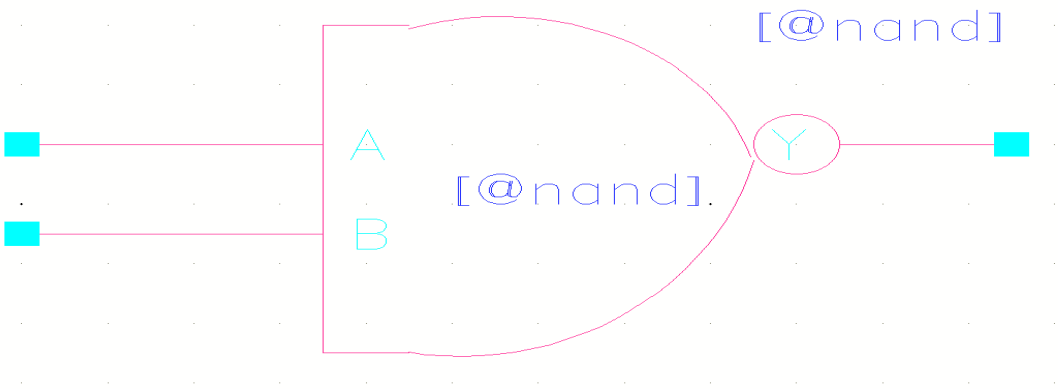

Figure 03: Symbol of an NAND gate.

3.4 Layout Generation:

- Layout XL was launched from the schematic view of the NAND gate.
- The connectivity information, such as pins and transistor placements, was extracted from the schematic.
- The I/O pins were set to Metal1, and the display options were configured for optimal visibility.
- the PR Boundary was removed to allow for flexible resizing.
- Connections between layers were made using the path and rectangle tools, and vias were inserted where necessary.
- M1_PSUB and M1_NWELL cells, which had been created earlier, were instantiated using the ‘i’ command and integrated into the layout.
- The vdd! and gnd! pins were moved to the appropriate power rails.
- Vertical wiring was routed using Metal1, and horizontal wiring was implemented using Metal2 to maintain a structured layout, and adjusting the cell height to 5 µm for standardization.

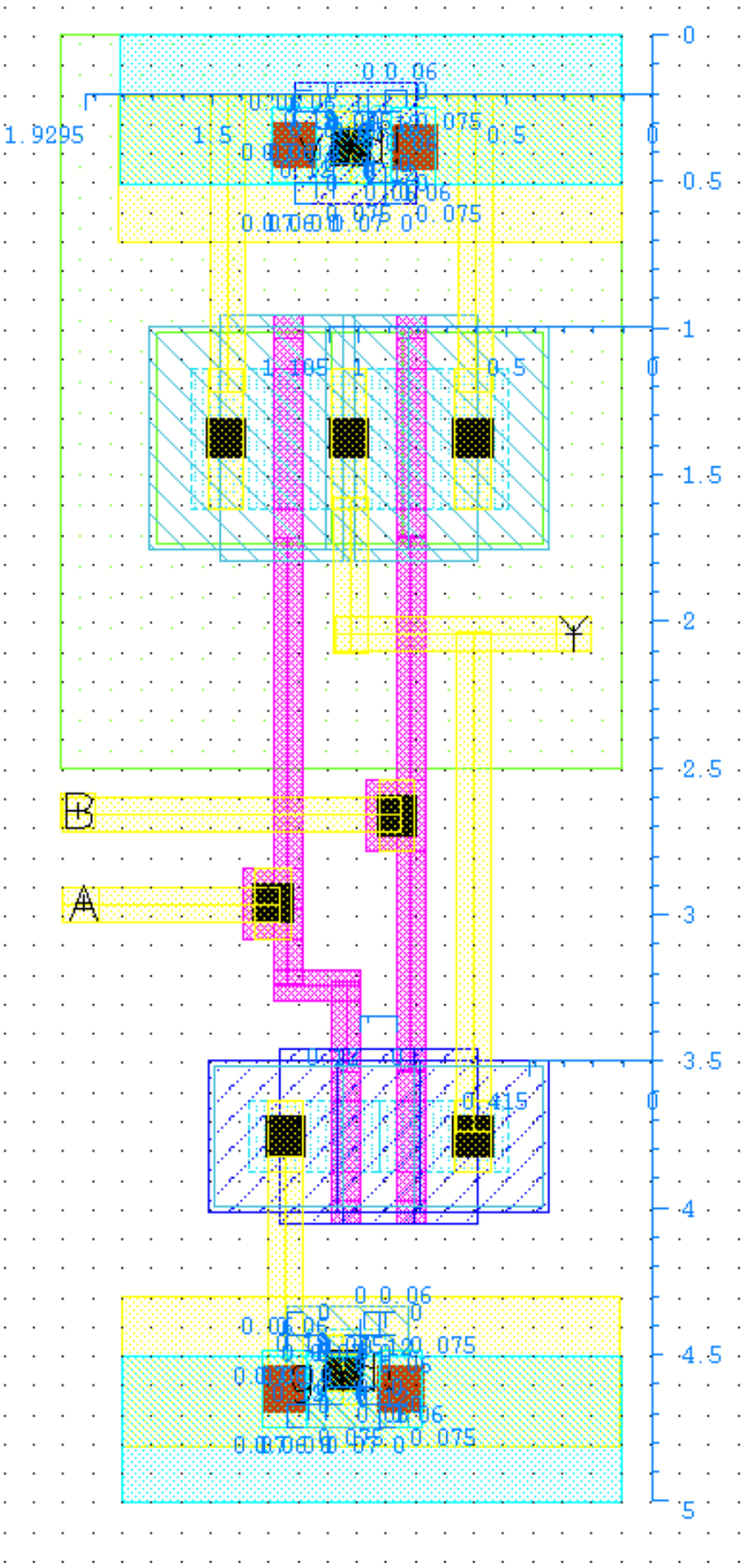

Figure 04:Layout of an NAND gate.

3.5 Design Rules Check (DRC) Verification:

- Design Rule Check (DRC) was performed to ensure that the layout adhered to the foundry's design rules.
- The errors identified during the DRC run were corrected, and a successful DRC completion was achieved.

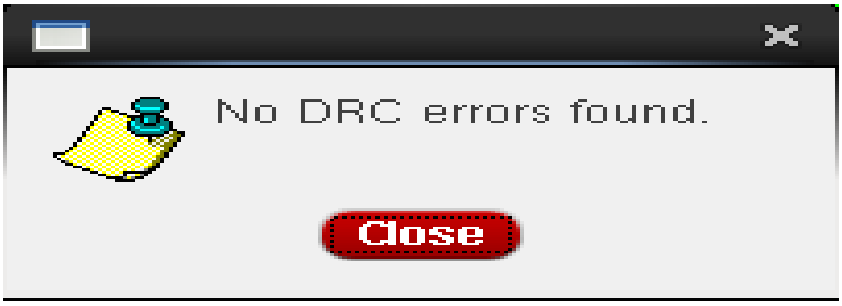

Figure 05: DRC Completion Window.

3.6 Layout vs. Schematic (LVS) Verification:

- Layout vs. Schematic (LVS) verification was conducted to ensure that the netlist generated from layout matched the schematic design. Any mismatch errors were resolved by verifying the interconnections and pin placements.

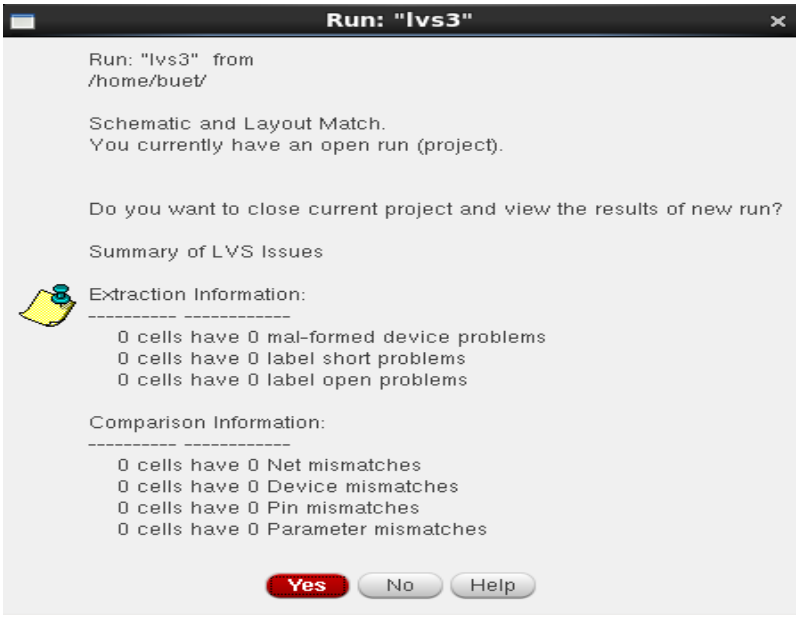

Figure 06: LVS Completion Window.

3.7 RCX Verification:

- After the final error-free LVS run, parasitic resistance and capacitance extraction (RCX) was performed to create an extracted view of the layout, which included parasitic elements.
- Upon completion, the extracted view was opened using File $>$ Open.

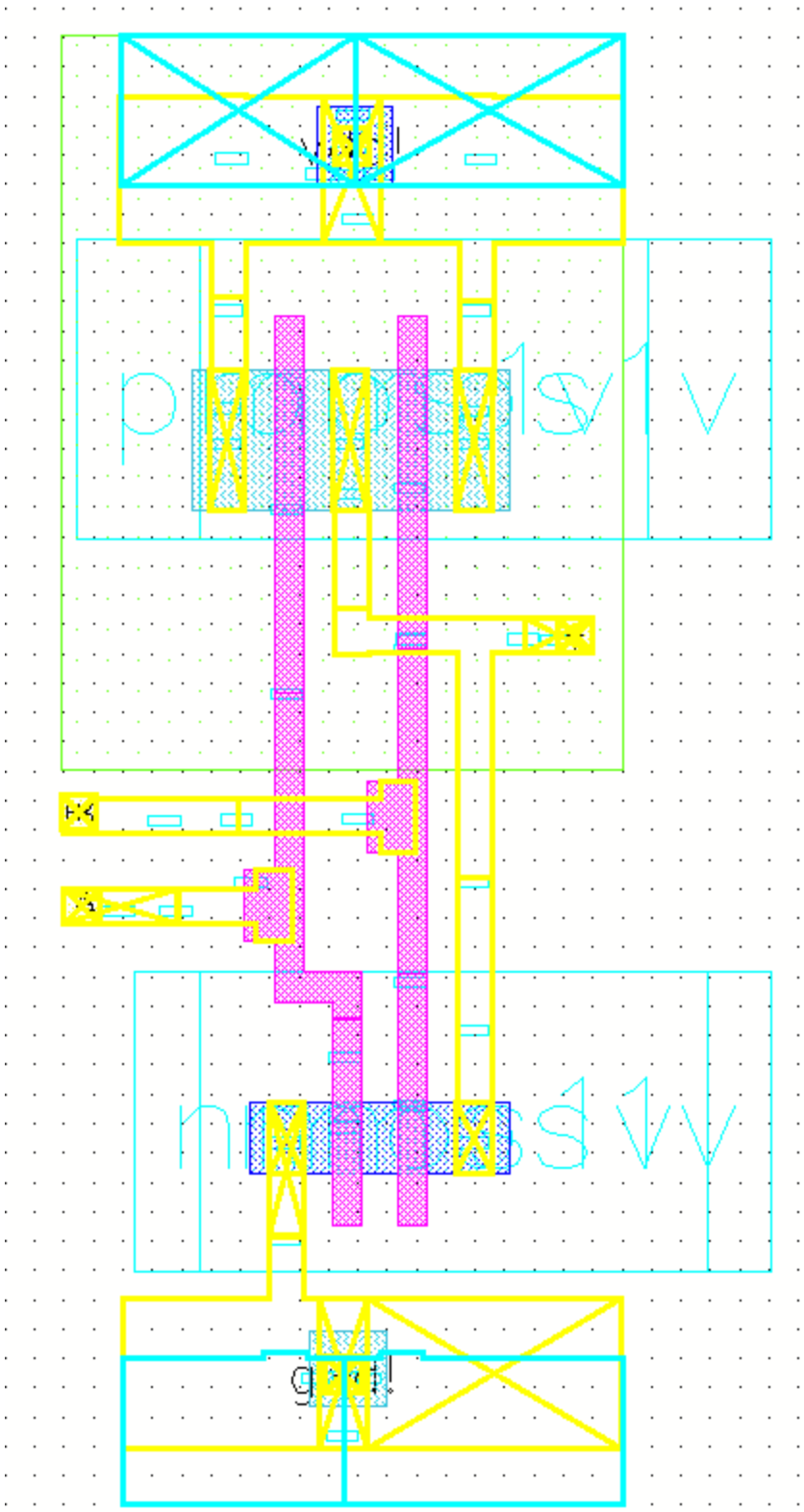

Figure 08:Extracted view of the NAND gate.

3.8 Post-Layout Simulation:

- The av_extracted view was opened for post-layout simulation. Inputs were applied to the NAND gate, and the output waveform was analyzed to ensure that the circuit's behavior matched the expected functionality.

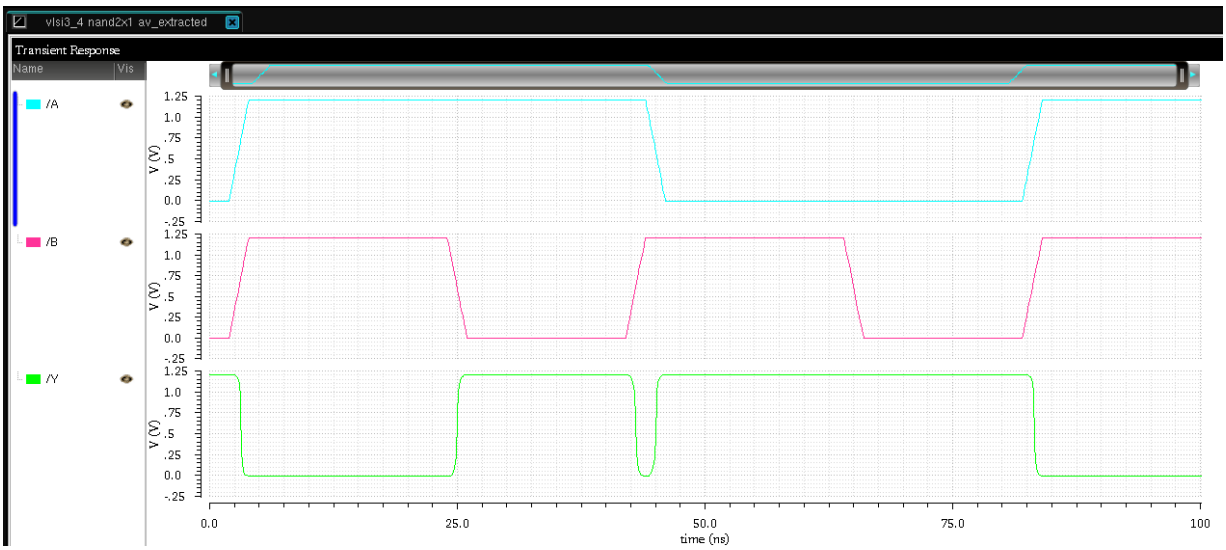

Figure 09:Output waveform of an NAND gate.

- A small glitch appears at the output of the NAND gate due to unequal delays in the two input paths. When input A transitions from high to low, input B takes some time to switch to high because of the inverter delay. As input A falls but remains above the NMOS threshold of 1V, the NMOS transistor connected to A stays on. Meanwhile, input B has already risen above 1V, turning the NMOS transistor connected to B on as well. During this overlap, both NMOS transistors conduct briefly, causing a glitch at the AND gate's output.

4 Home Task:

Difference between Stacked and Unstacked Transistors in Post-layout Simulation:

- Area Efficiency: Stacked transistors take up less space in the layout compared to unstacked transistors.
- Switching Speed: Unstacked transistors tend to switch faster due to lower resistance, while stacked transistors are slower because of higher series resistance.
- Power Consumption: Stacked transistors may have lower leakage currents, making them more power-efficient in low-power applications.
- Parasitic Elements: Stacking reduces parasitic capacitances between the transistors, but the increased resistance can slow down the circuit's performance.
- Use Cases: Stacked transistors are more suitable for power-sensitive designs, while unstacked transistors are ideal for high-performance, speed-critical circuits.

Why we should stack transistors with common drain/source terminals?

Stacked transistors with shared source/drain terminals offer several advantages. Stacking reduces parasitic capacitance and minimizes area by eliminating unnecessary connections. This improves the overall performance in terms of reduced delay and lower power consumption compared to unstacked transistors. Additionally, stacked transistors help in routing optimization as fewer vias are required, which is crucial in dense layouts like standard cells.

5 Discussion:

- In this experiment the schematic-driven layout design of the NAND gate was successfully completed using the Virtuoso Layout Editor XL.
- Schematic-level verification ensured that the logic of the NAND gate was correct.
- Stacking transistors with common source/drain terminals helped in optimizing the layout area and improving performance metrics.
- The DRC and LVS checks helped to identify and resolve any layout inconsistencies or rule violations.
- After correcting minor errors in the layout, the design passed all verification steps.
- Parasitic elements were extracted, providing insights into their impact on the NAND gate's performance.
- The post-layout simulation results confirmed that the extracted layout met the required functional specifications of a NAND gate.
- Overall, this experiment demonstrated the importance of transistor stacking and proper routing techniques to minimize parasitic effects and improve design efficiency in VLSI circuits.