**Date: 13/09/2025 Timing: 9:30 am to 4:15 pm**

**Student Name: Om Kasvala Enrollment No: 92410133005**

**Division: 3EK2**

**Long Hour Design (LHD) – Digital Circuit Layouts in Microwind**

Task 1: Design and Simulation of 4-bit Full Adder in Microwind

1. Introduction

The full adder is one of the most fundamental combinational circuits in digital design. It adds three binary inputs: **A**, **B**, and **Carry-in (Cin)** to produce two binary outputs: **Sum (S)** and **Carry-out (Cout)**.

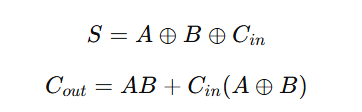
A **4-bit full adder** can be constructed by cascading four 1-bit full adders, forming a **Ripple Carry Adder**. This circuit is a crucial building block in **arithmetic logic units (ALUs)**, digital processors, and microcontrollers.

The objective of this task is to design, implement, and simulate a 4-bit full adder using **Microwind**, verifying its functionality through transistor-level layouts and timing diagrams.

2. Circuit Design

2.1 Logic Equations

For a 1-bit Full Adder:

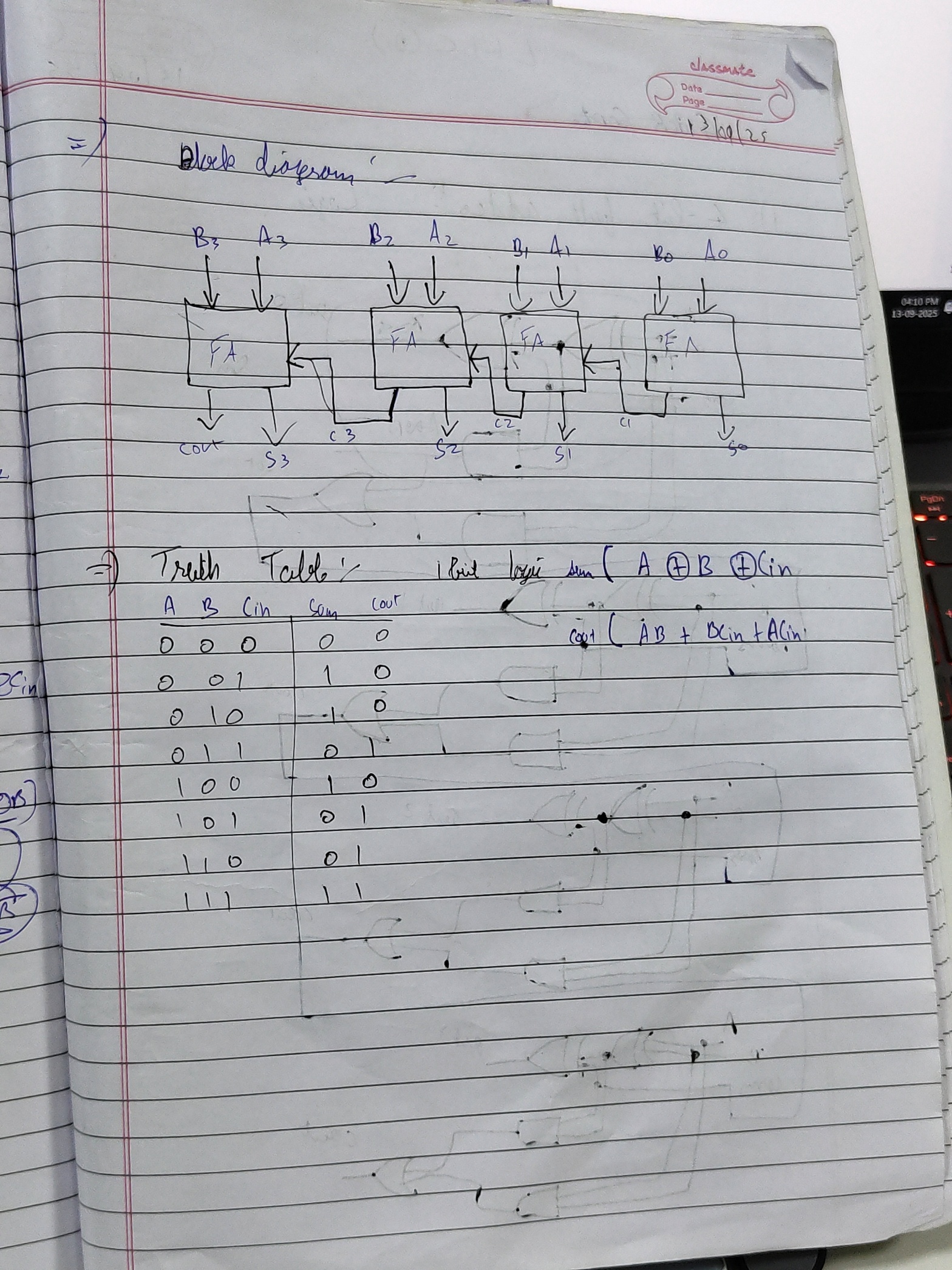


2.2 Truth Table for 1-bit Full Adder

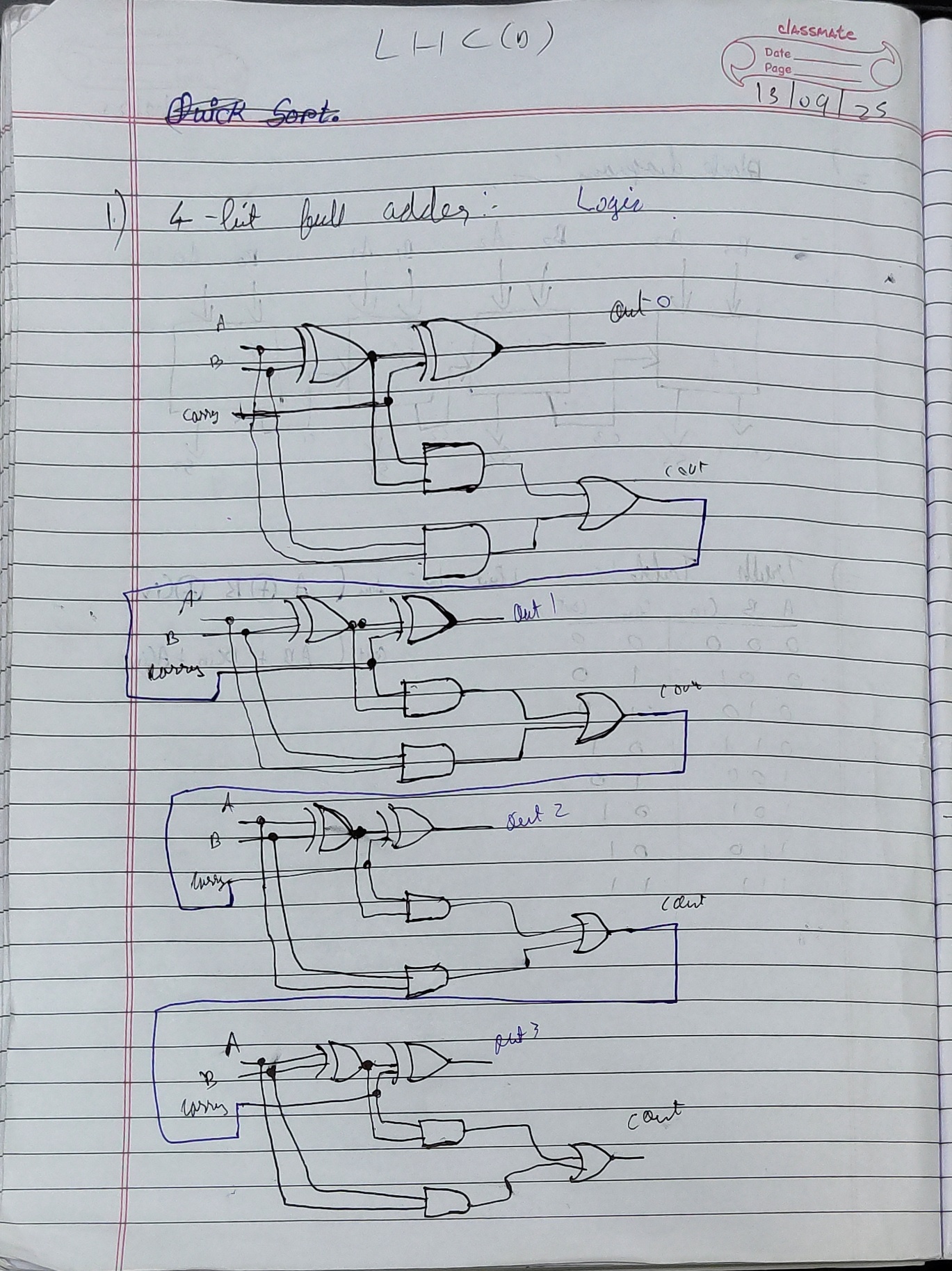
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | Sum | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**2.3 Block Diagram**

* The 4-bit ripple carry adder is built by connecting four full adder blocks in series.
* The carry output from each stage acts as the carry input for the next stage.

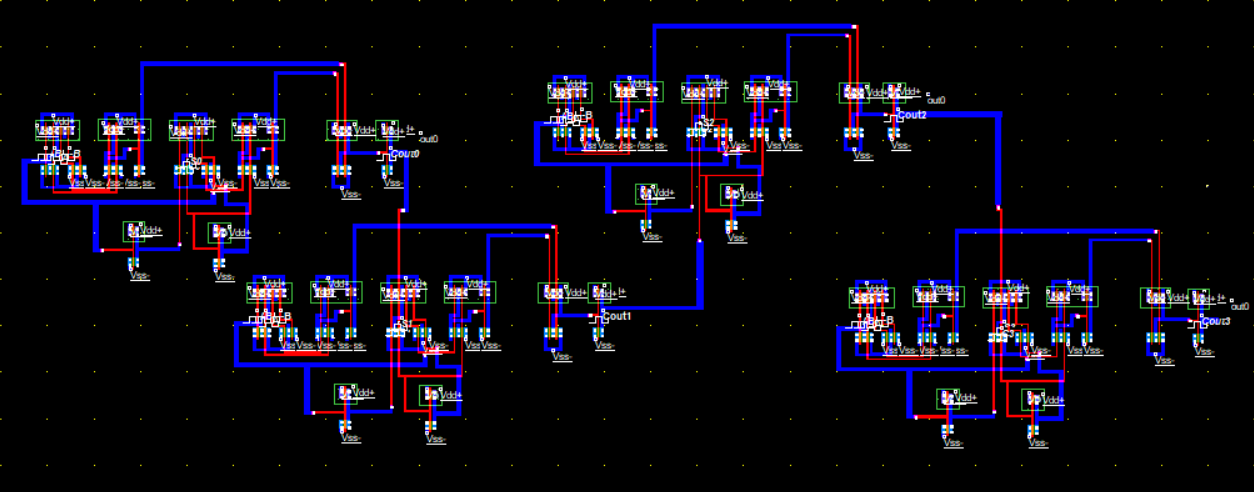


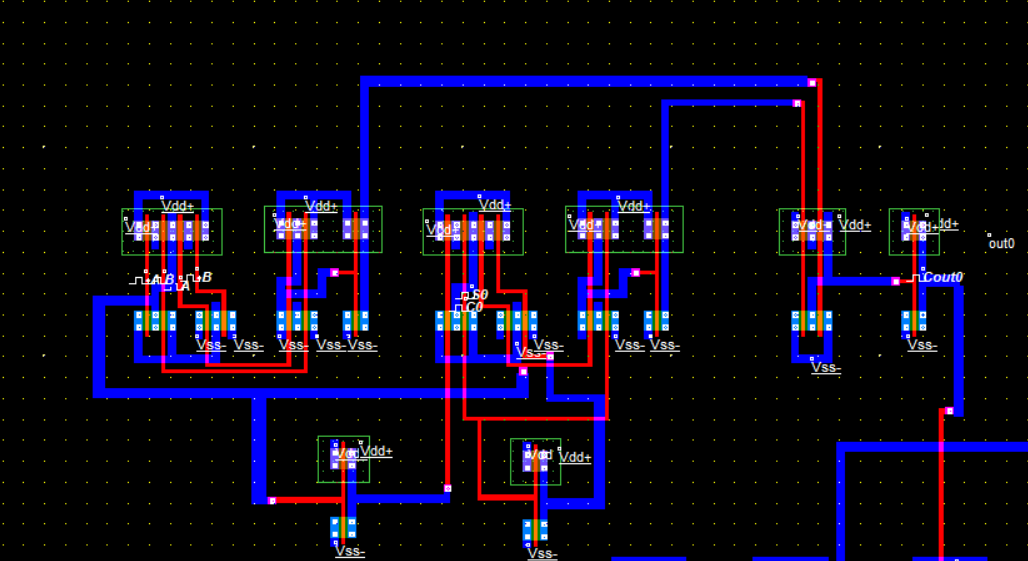
🡺Bookwork:



3. Layout Design in Microwind

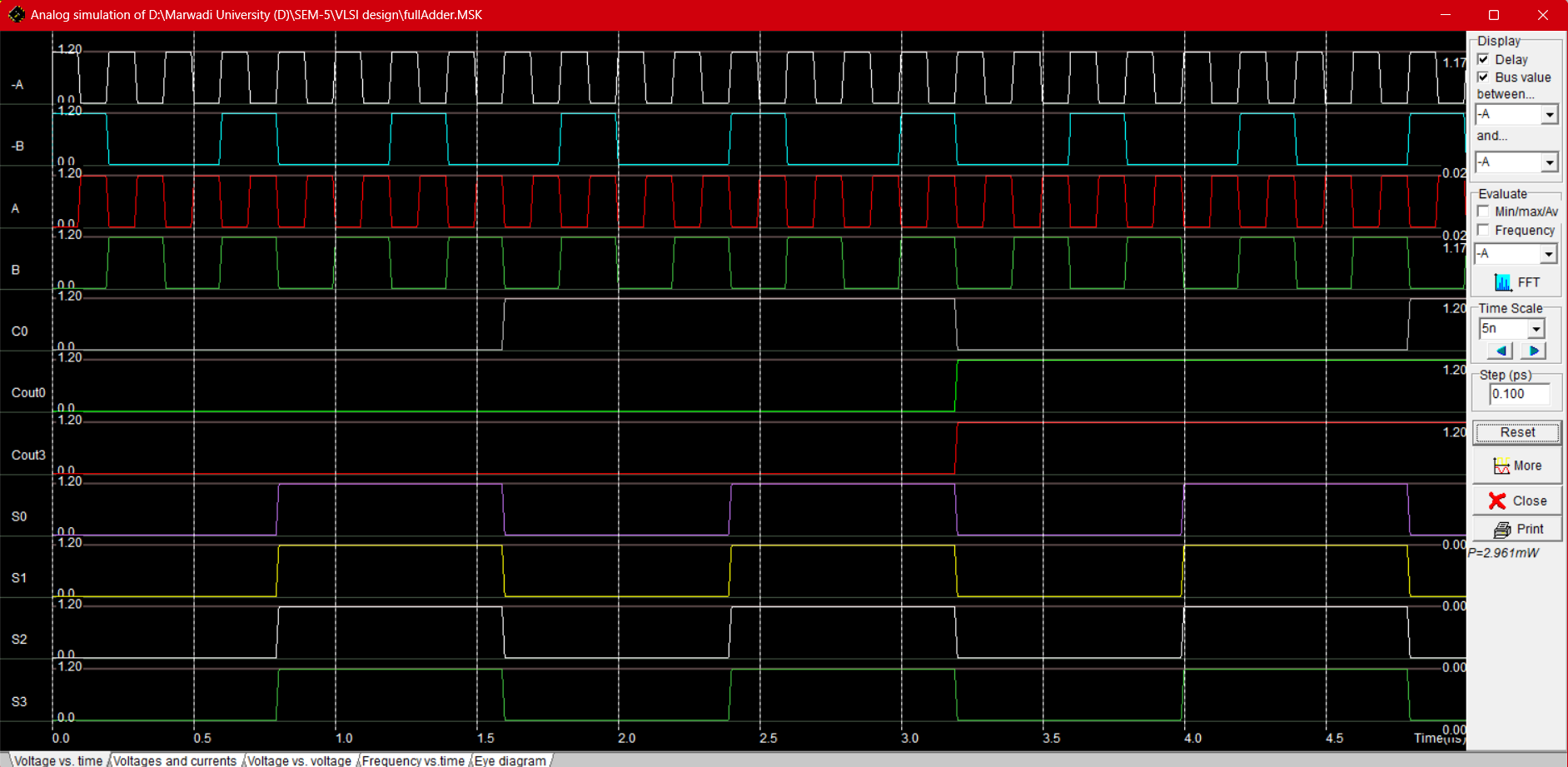
* The circuit was implemented in **Microwind** using CMOS technology.
* **PMOS transistors** (connected to Vdd) and **NMOS transistors** (connected to Vss) were used to realize XOR, AND, and OR functions.
* Four 1-bit adders were connected in cascade to form the 4-bit adder.





4. Simulation Results

* Simulation was carried out in Microwind with varying inputs of A, B, and Cin.
* The resulting timing diagrams clearly show the correct Sum and Carry outputs for different test cases.



5. Observations

**5.1 Case-by-Case Analysis for 1-bit Full Adder**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time (ns) | A | B | Cin | Expected Sum | Expected Cout | Observed Sum | Observed Cout | Result |
| 0–10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Correct |
| 10–20 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Correct |
| 20–30 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | Correct |
| 30–40 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | Correct |
| 40–50 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Correct |
| 50–60 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | Correct |
| 60–70 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | Correct |
| 70–80 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Correct |
|  |  |  |  |  |  |  |  |  |

**5.2 General Observations**

1. The circuit correctly implements the logic of a 1-bit and 4-bit full adder.
2. Carry propagation delay is observed, consistent with ripple carry design.
3. Simulation waveforms are stable with no glitches or undefined outputs.
4. Outputs match the theoretical truth table for all tested cases.
5. Power consumption during simulation was measured at approximately **2.96 mW**.

6. Conclusion

* A **4-bit Full Adder** was successfully designed, implemented, and simulated using Microwind.
* Layout design followed CMOS design rules with proper PMOS and NMOS placement.
* Simulation results verified the correct functionality of the adder, with Sum and Carry outputs matching the theoretical truth table.
* Carry propagation delay is inherent to the ripple carry adder, but the design remains effective for small-scale applications.
* This full adder can be extended for larger bit-widths and used as a building block in an **Arithmetic Logic Unit (ALU)**.

Task 3: Design and Simulation of 1-bit Magnitude Comparator

1. Introduction

A **magnitude comparator** is a fundamental combinational logic circuit that compares two binary numbers (A and B) and determines their relative magnitude. It produces three outputs to indicate whether A is greater than B, A is equal to B, or A is less than B.

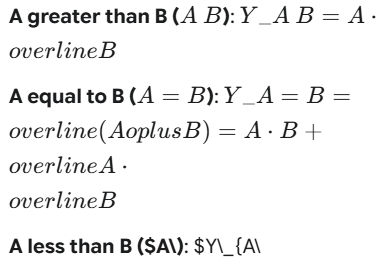
The 1-bit comparator is the basic building block from which larger, n-bit comparators can be constructed. These circuits are crucial in digital systems for decision-making processes, particularly within the Arithmetic Logic Units (ALUs) of central processing units (CPUs).

The objective of this task is to design, create a CMOS layout, and simulate a 1-bit magnitude comparator using Microwind to verify its logical functionality and performance.

2. Circuit Design

**2.1 Logic Equations**

For a 1-bit comparator with inputs A and B, the three outputs are defined by the following Boolean expressions:

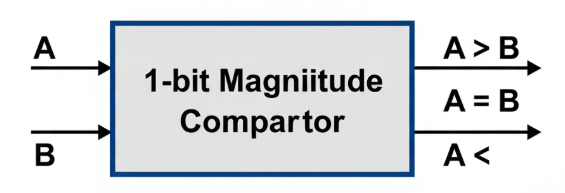
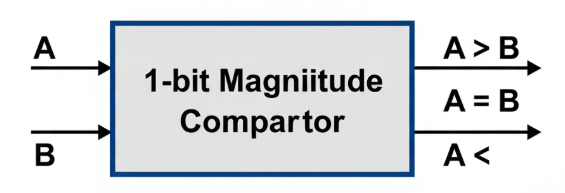
****

**2.2 Truth Table for 1-bit Magnitude Comparator**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | A > B | A = B | A < B |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

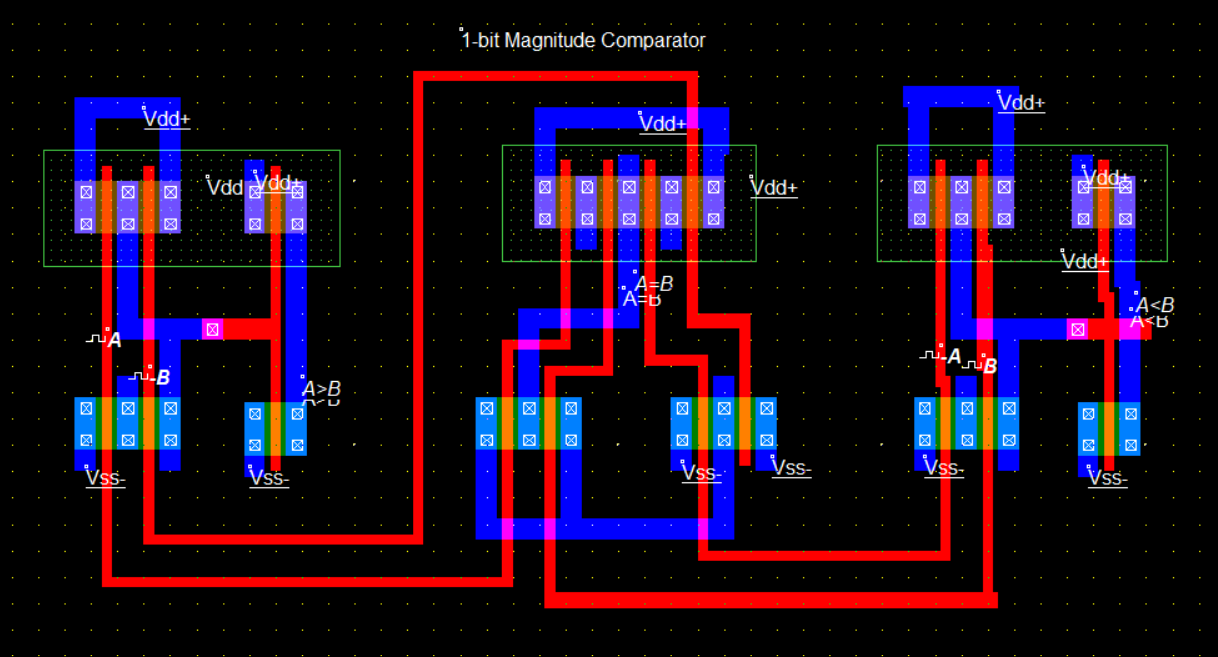
Export to Sheets

**2.3 Block Diagram**

The 1-bit comparator can be represented as a single logic block with two inputs and three outputs. 

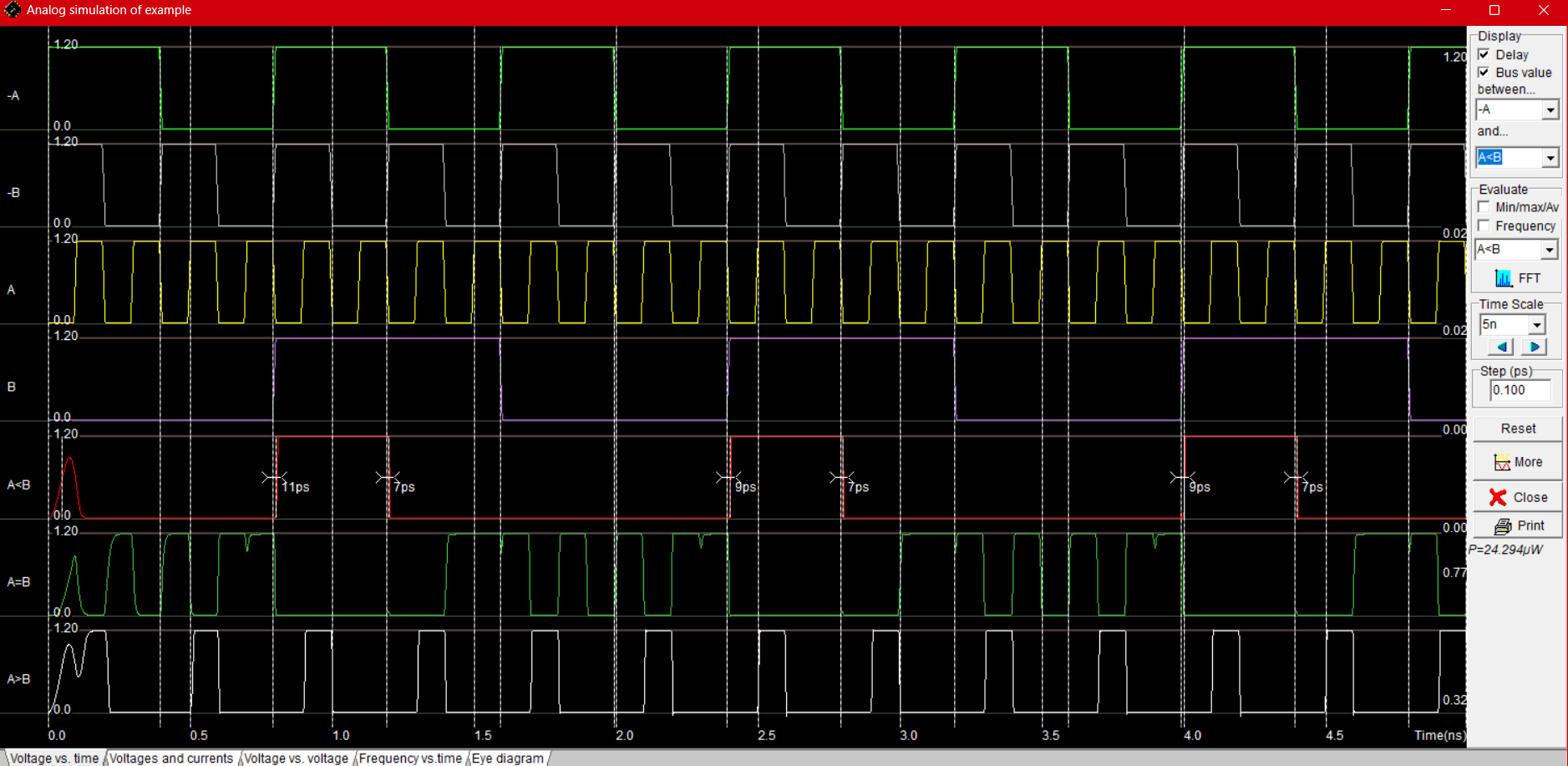
3. Layout Design in Microwind

The circuit was implemented in Microwind using CMOS technology. The layout consists of three separate logic gates constructed from **PMOS transistors** (connected to Vdd) and **NMOS transistors** (connected to Vss). Each gate corresponds to one of the comparator's outputs (AB, A=B, and $A\). The inputs A and B are routed to these three blocks to generate the required comparison results.



4. Simulation Results

The layout was simulated in Microwind by applying clock signals to inputs A and B to cycle through all four possible input combinations (00, 01, 10, 11). The resulting timing diagram confirms the correct logical behavior of the comparator, with the outputs switching to high or low states according to the truth table.



**5. Observations**

**5.1 Case-by-Case Analysis**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Approx. Time (ns) | A | B | Expected A>B | Expected A=B | Expected A<B | Observed A>B | Observed A=B | Observed A<B | Result |
| 0.0–0.25 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Correct |
| 0.25–0.50 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | Correct |
| 0.50–0.75 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Correct |
| 0.75–1.00 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Correct |

**5.2 General Observations**

* The circuit successfully implements the logic of a 1-bit magnitude comparator.
* The simulation waveforms are stable and clean, with outputs matching the theoretical truth table for all tested cases.
* Minor **propagation delays** (measured in picoseconds, e.g., 7ps, 9ps) are observed between input transitions and the corresponding output responses, which is characteristic of transistor switching times.
* The three outputs (AB, A=B, $A\) are **mutually exclusive**; only one output is high at any given time, as expected.
* Power consumption during simulation was measured at approximately **24.28 µW**.

**6. Conclusion**

A **1-bit Magnitude Comparator** was successfully designed, implemented at the transistor level, and simulated using the Microwind tool. The CMOS layout was created following standard design rules, and the subsequent simulation verified its correct logical functionality. The observed outputs for all possible input combinations perfectly matched the comparator's truth table.

This design demonstrates the fundamental principles of creating combinational logic circuits using CMOS technology. The verified 1-bit comparator module can be reliably used as a foundational cell to build larger, multi-bit comparators for more complex digital systems.

Git hub link: https://github.com/ompatelok/VLSI-Long-Hour-Designing/