CMOS VLSI Circuits Lab
Open-Ended Experiment

Topic:
Design and Implementation of 1-bit Comparator using fully CMOS and Pseudo NMOS logic

Team:15
Team Members:

SI.NO	Name	USN	Roll.no
1	Vishwanath.H	01FE22BEC254	512
2	Karthik.V.M	01FE22BEC270	526
3	Nirupadi	01FE22BEC277	533
4	Akash. B	01FE22BEC287	543

1-Bit Comparator

A 1-bit comparator compares two binary inputs (A and B) and produces three outputs:

A > B: Output is high if input A is greater than B

.**((A>B)=A▪B’)**

A = B: Output is high if inputs A and B are equal**.**

((A=B)=A⊕B=(A’.B’)+(A▪B))

A < B: Output is high if input A is less than B**.**

((A<B)=A’▪B))

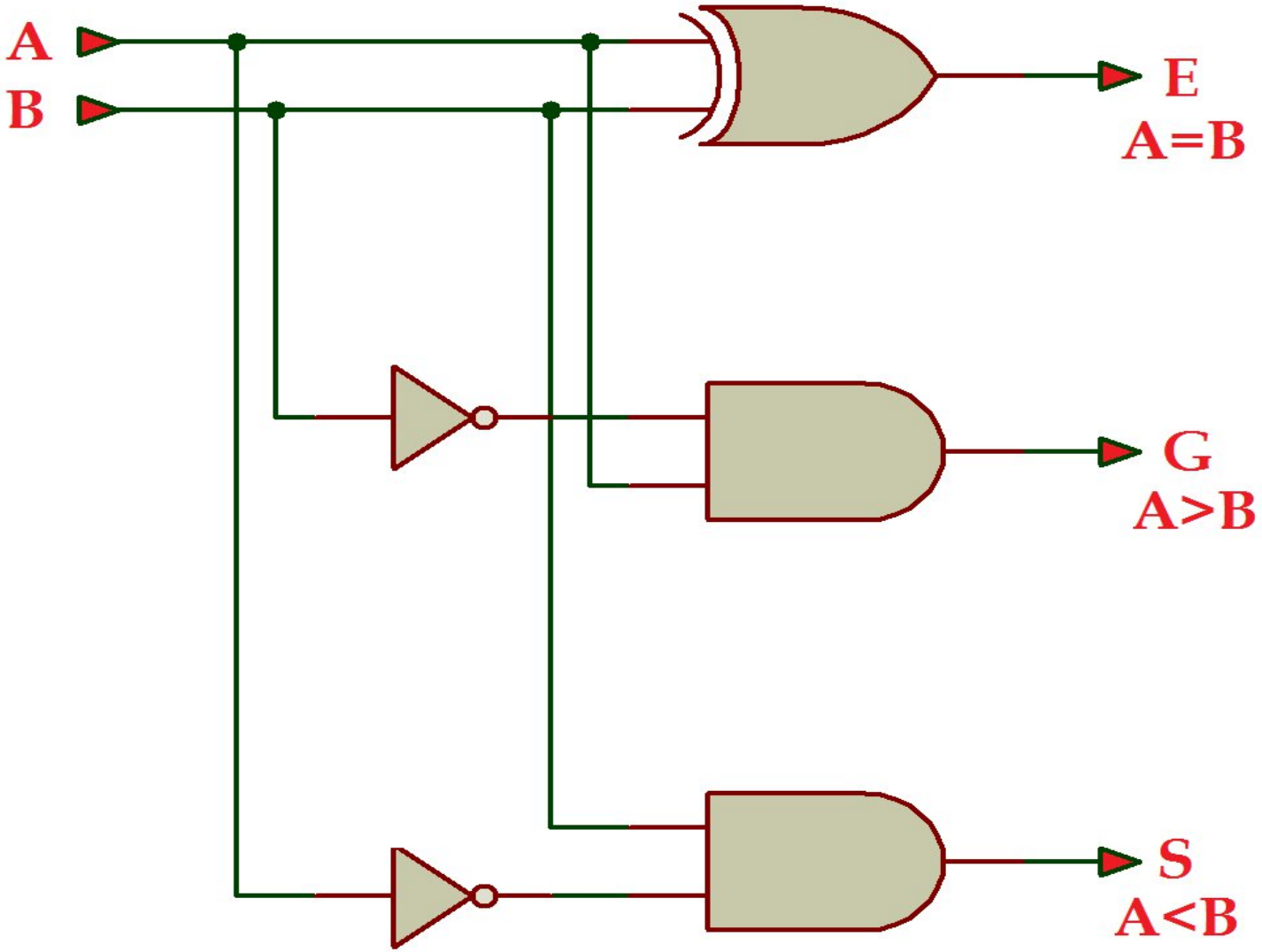
The design of a 1-bit comparator is based on logic gates, and two common logic styles used are **fully CMOS logic** and **Pseudo NMOS logic**.

1-Bit Comparator:-

Truth Table:-

Inputs		Outputs		
B	A	G(A > B)	E(A = B)	L(A < B)
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

1-Bit Comparator Circuit diagram:-



1-Bit Comparator Using Fully CMOS Logic

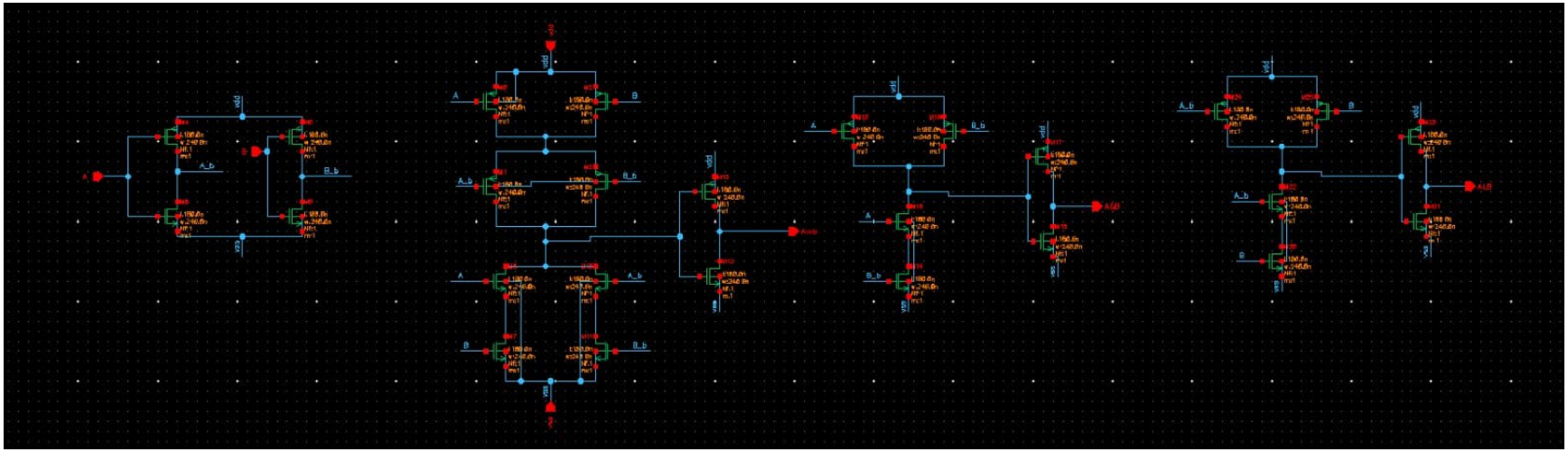
- In **fully CMOS logic**, the implementation requires a complementary PMOS pull-up network and an NMOS pull-down network for each equation.
- **Implementation for A>B:**
  - PMOS network is active when A=1and B=0 (pull-up).
  - NMOS network is active when A=1 and B=0(pull-down).
- **Implementation for A=B:**
  - Requires the XOR logic to compute A⊕B, then invert it.
  - CMOS XOR gate consists of multiple PMOS and NMOS transistors. The inverted output uses a complementary inverter circuit.
- **Implementation for A<BA < BA<B:**
  - PMOS network is active when A=0 and B=1 (pull-up).
  - NMOS network is active when A=0 and B=1 (pull-down).

1-Bit Comparator Using Pseudo NMOS Logic

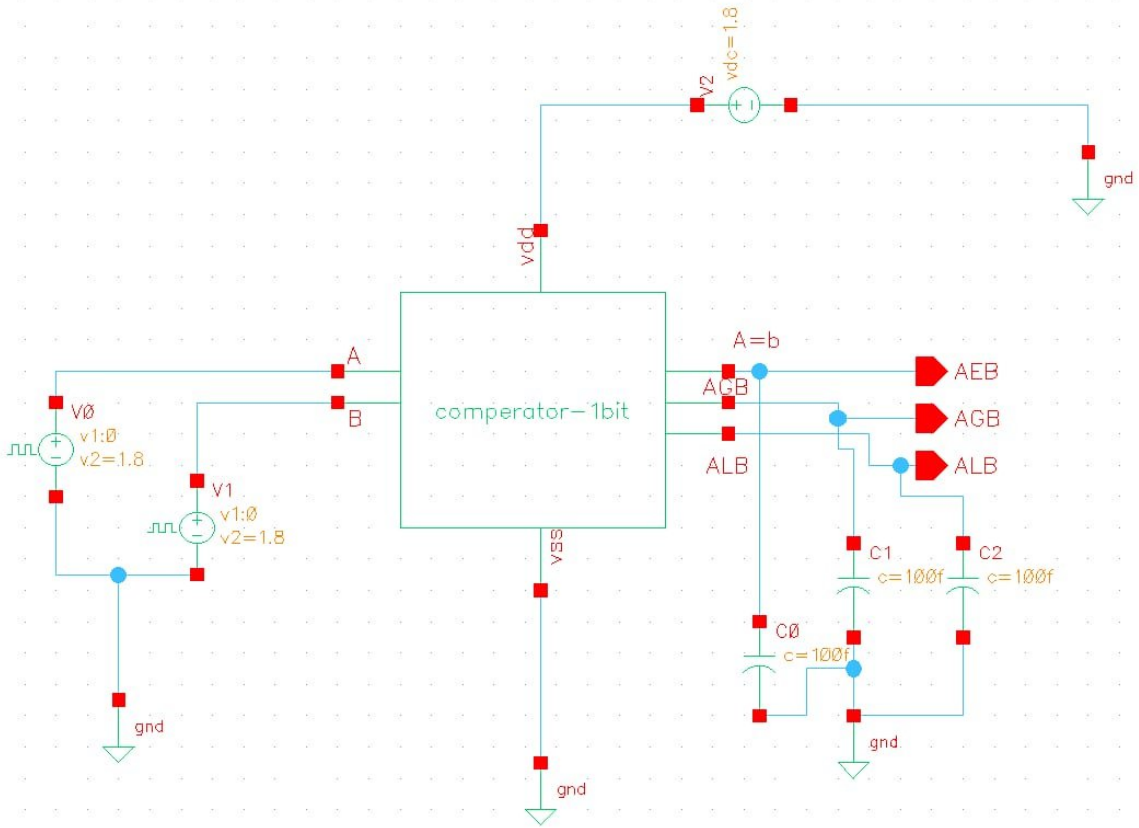
In **Pseudo NMOS logic**, the pull-up network is replaced by a single PMOS transistor that is always ON, while the NMOS network handles the logic operations.

- **Implementation for A>B:**
  - The NMOS network implements A·B’
  - The PMOS transistor acts as the pull-up resistor.
- **Implementation for A=B:**
  - The NMOS network implements (A’·B’)+(A·B)
  - The PMOS transistor pulls the output high when no path to ground exists.
- **Implementation for A<B:**
  - The NMOS network implements A’·B
  - The PMOS transistor pulls the output high when the NMOS path is OFF.

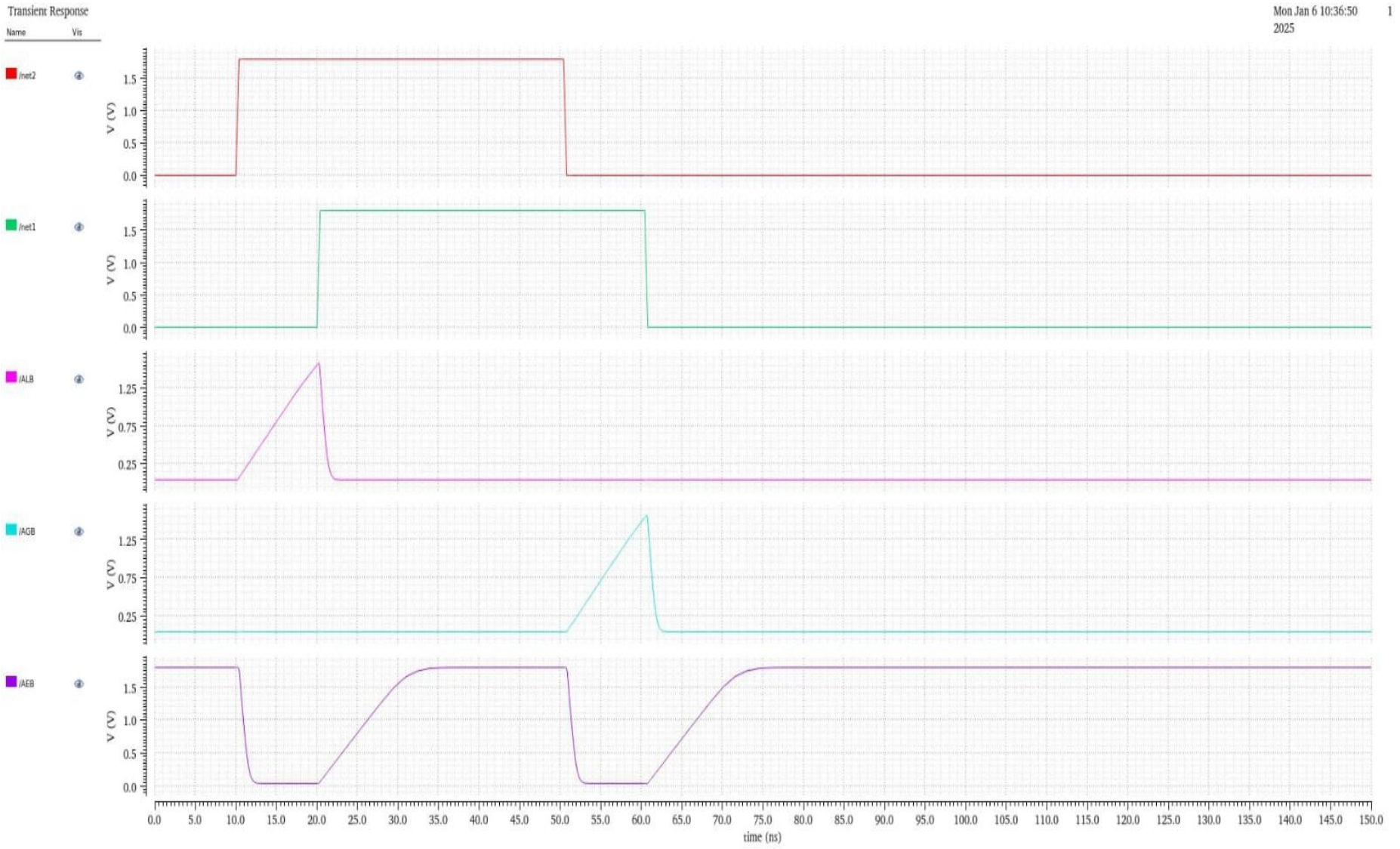
CMOS Logic:Schematic



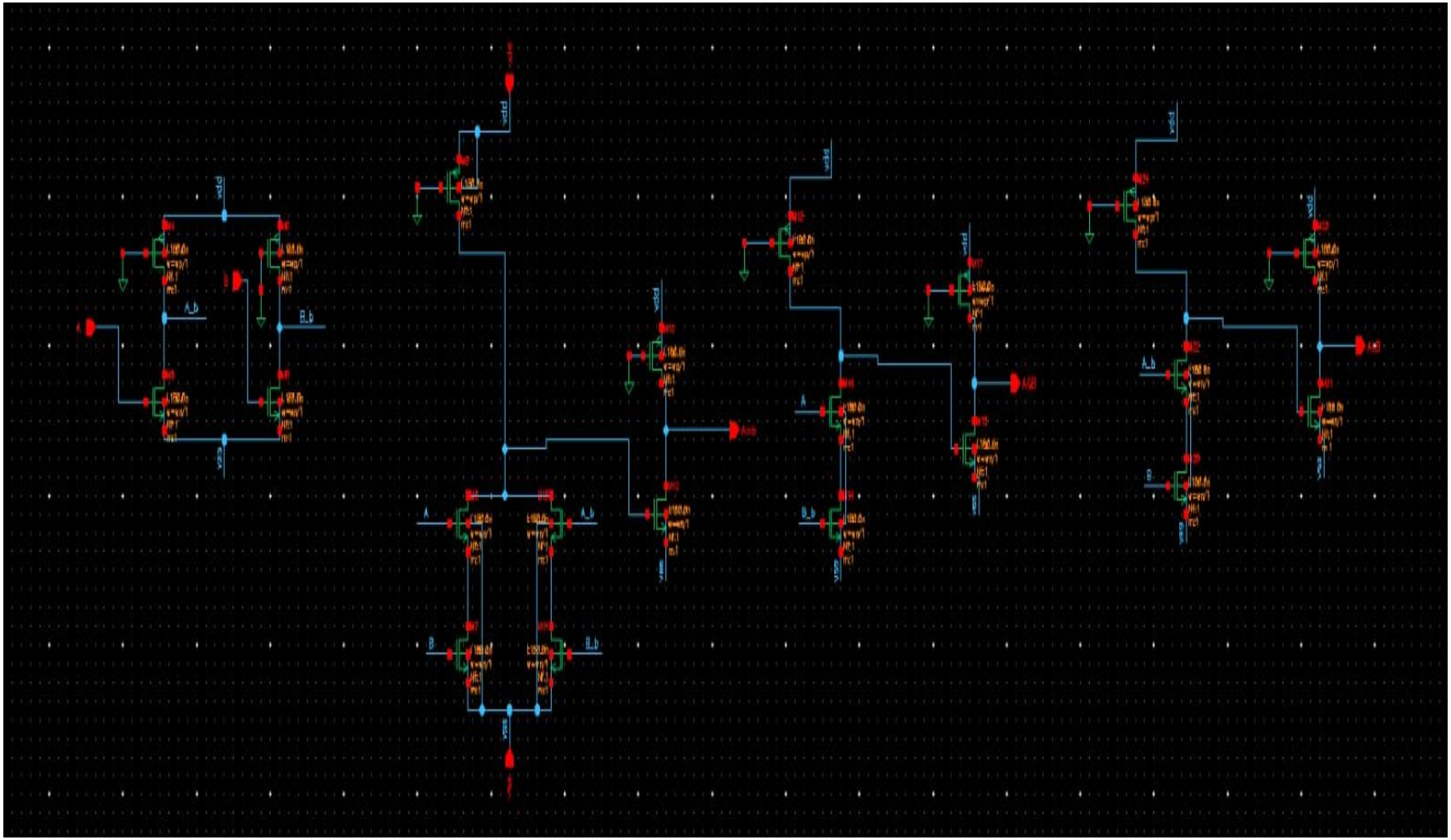
CMOS Logic:Test circuit



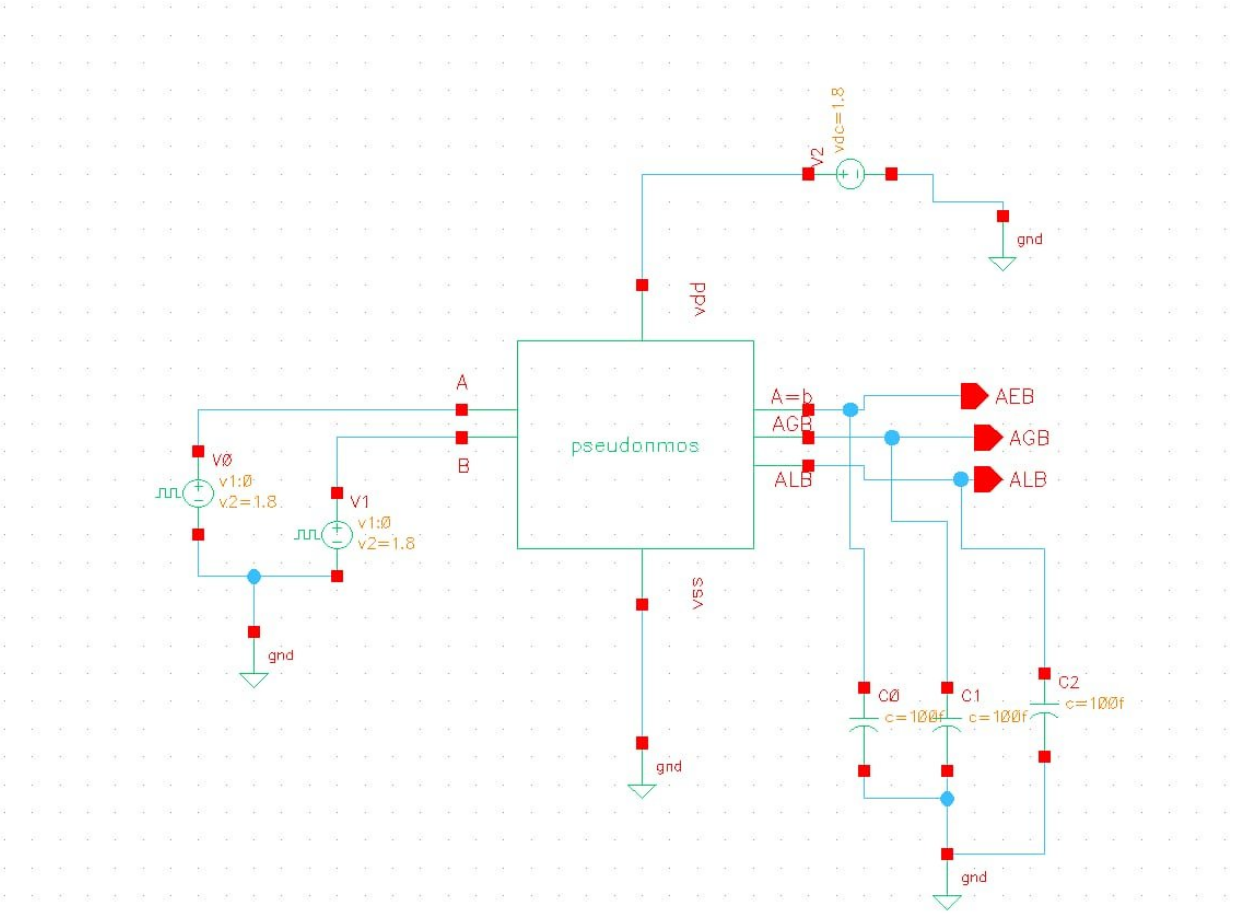
CMOS Logic:Output Waveform



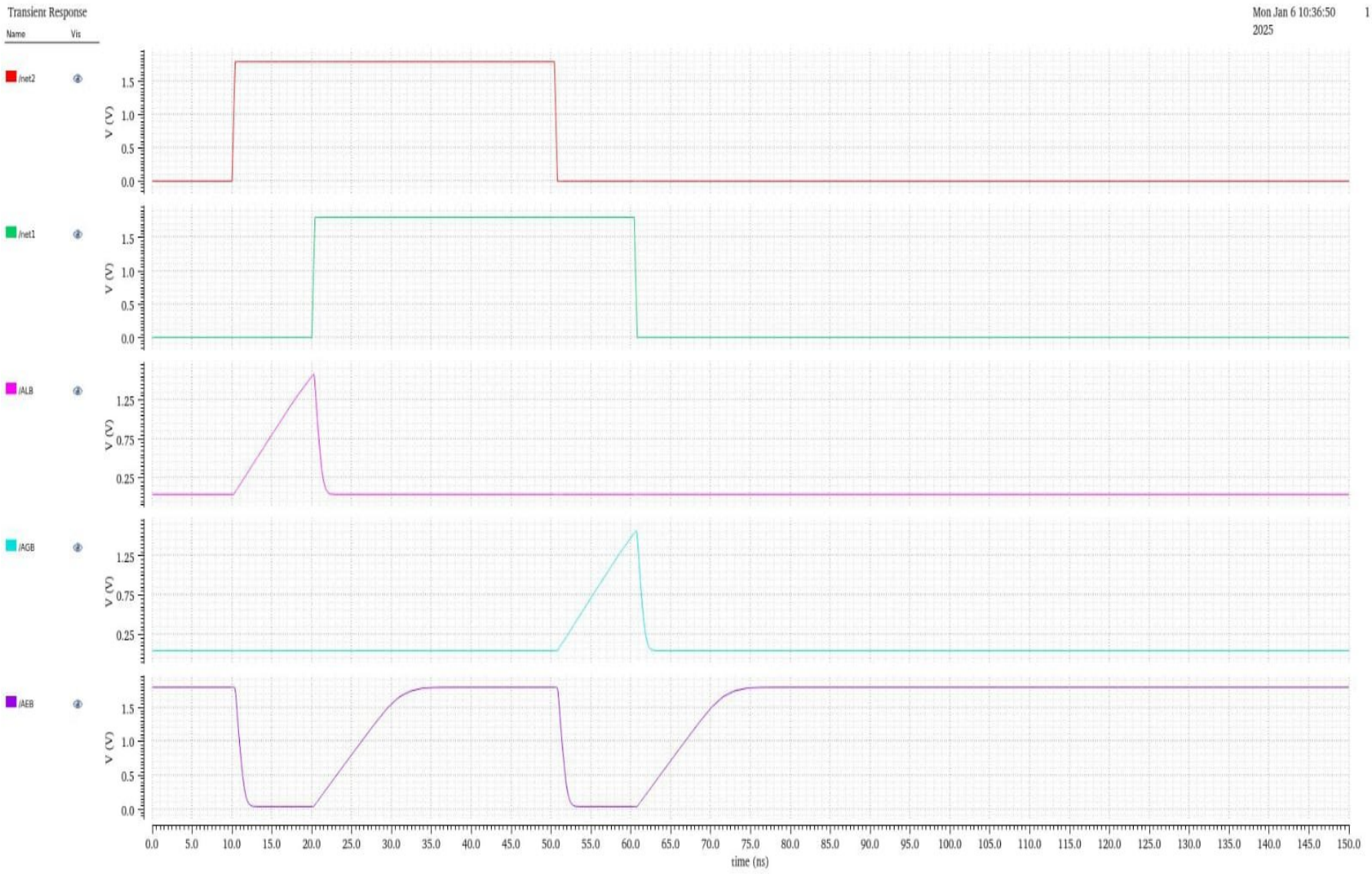
Pseudo NMOS:Schematic



Pseudo NMOS:Test circuit



Pseudo NMOS:Output Waveform



Comparison of Output Values Obtained

CMOS logic

A=B

- Avg. Rise Time, tr =3.6 ns
- Avg. Fall Time, tf = 1.18 ns
- Avg. Power = 1.255W

Pseudo NMOS logic

A=B

- Avg. Rise Time, tr =3.59 ns
- Avg. Fall Time, tf = 2.132ns
- Avg. Power =1.341W

Comparison of Output Values Obtained

CMOS logic

A>B

- Avg. Rise Time, tr =3.614 ns
- Avg. Fall Time, tf = 1.105 ns
- Avg. Power = 0.347W

Pseudo NMOS logic

A>B

- Avg. Rise Time, tr =3.593 ns
- Avg. Fall Time, tf = 2.128 ns
- Avg. Power = 0.338W

Comparison of Output Values Obtained

CMOS logic

A<B

- Avg. Rise Time, tr =3.581 ns
- Avg. Fall Time, tf = 1.105 ns
- Avg. Power = 0.1056W

Pseudo NMOS logic

A<B

- Avg. Rise Time, tr =3.591 ns
- Avg. Fall Time, tf = 2.116 ns
- Avg. Power = 0.340W

Comparison Between Fully CMOS and Pseudo NMOS Logic

Fully CMOS Logic:	Pseudo NMOS Logic:
Accurate and power-efficient.	Simpler and faster due to fewer transistors.
More transistors required (higher area and complexity).	Consumes more static power.
Area will be more due to more transistors.	Area will be less due to less transistors.

THANK YOU