Digital VLSI Design: Project 2

32 × 32 SRAM Array

by

Abhinav S (2022102037)
Chirag Goyal (2022102041)
Himanshu Yadav (2022102012)
Dikshant (2022102038)

Course Instructor: Dr. Zia Abbas

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
H Y D E R A B A D

International Institute of Information Technology Hyderabad
500 032, India

Contents

Introduction

This project involves the design and analysis of a 32×32 SRAM (Static Random-Access Memory) array, a memory structure comprising 1024 bits organized in 32 rows and 32 columns. The primary focus of this work is the implementation and verification of the read operation, a fundamental functionality of SRAM. The design aims to optimise key performance metrics such as power consumption, access speed, and area efficiency while ensuring operational reliability.

The project examines the architectural design of the SRAM array, focusing on the arrangement of word lines, bit lines, and memory cells, along with the peripheral circuits essential for the read operation. These peripherals include a 5-to-32 decoder for row selection, a 5-bit adder for address generation, and a sense amplifier array for accurate data retrieval. The 32×32 configuration serves as a simplified yet effective framework to analyze SRAM functionality and performance, providing valuable insights into its application in embedded systems and high-performance computing. A 32×32 SRAM array or 1kb memory is to be implemented using TSMC 180 nm technology node with a supply voltage of 1.8V.
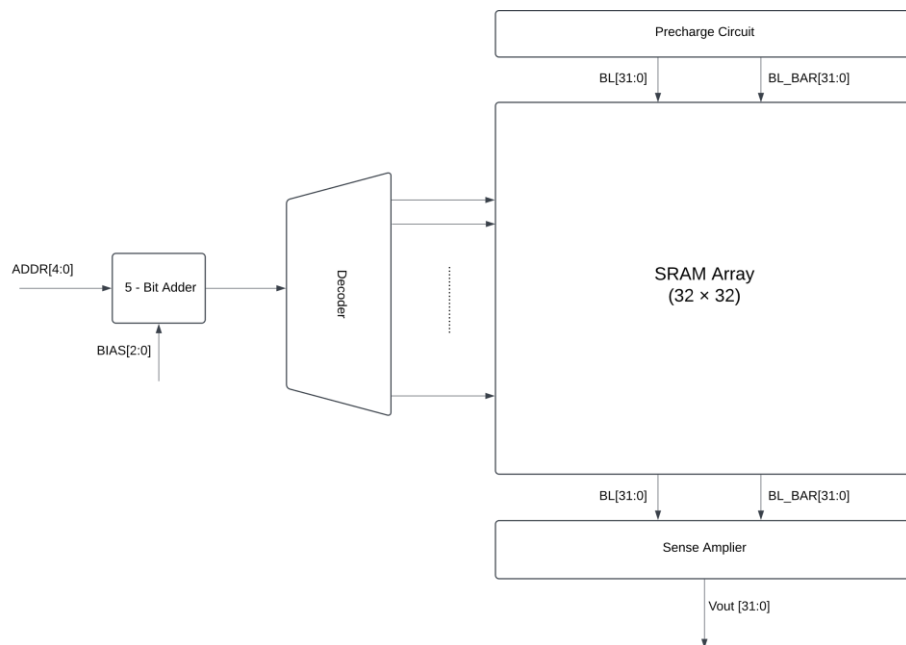
High-Level Structure

Figure 1 SRAM Block Diagram

Figure 1 shows the block diagram of the implemented SRAM. It consists of a 5-bit adder, a 5 to 32 decoder, a 32 × 32 SRAM array, a precharge circuit, and a sense amplifier. The 5-bit adder adds the 5-bit relative address and the 3-bit bias that is provided to get the required address. This 5-bit address is then provided to the 5 to 32 decoder to select the particular address. The outputs of the decoder are connected to the world line of the SRAM array and depending on the output of the decoder, the corresponding word line is selected and the word is selected. The SRAM array consists of 32 × 32 = 1024 cells. These 6T SRAM cells consist of 6 transistors. $ as a cross-coupled inverter to store the data and 2 as access transistors. These access transistors are connected to two lines – BL and $\overline{BL}$. These lines are connected to a precharge circuit to precharge them to V_{DD} before every read cycle. These lines are fed to a sense amplifier to sense the difference between both these lines and provide an output depending on the data inside the cell being read.

Design and Implementation

SRAM Cell

The 6T-SRAM cell consists of a cross-coupled inverter and two access transistors as shown in Figure 2. The cross-coupled inverter acts as a latch or the actual memory element. When the input of one inverter is "0", the output of that inverter will be "1" and the output of the other inverter will be "0". The inputs/outputs of the transistors are connected to the lines BL and $\overline{BL}$ through the access transistors. The gates of the access transistors are connected to the word line (WL). When WL is high, both the access transistors turn "ON" and BL and $\overline{BL}$ will be accessed through the access transistors by the inputs of the inverters. When WL is low, the cell is said to be in the HOLD state.
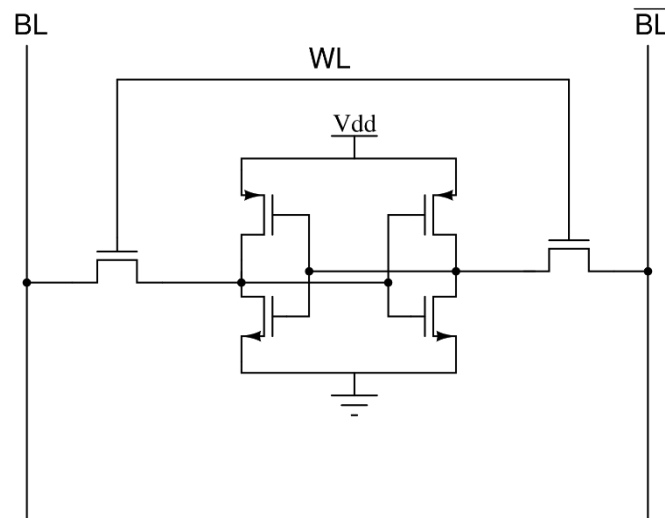

Figure 2: 6-T SRAM cell

During a read cycle, both BL and $\overline{BL}$ are precharged to V_{DD} initially using the precharge circuit. Then the WL is made high. This turns both the access transistors on. Assume that Q was 1 and $\overline{Q}$ was 0. When the access transistors are on, $\overline{BL}$ will discharge through the pull-down network of the second inverter. The sense amplifier senses the difference between the voltages at the lines BL and $\overline{BL}$ and acts basically as a comparator. It then outputs a voltage of either logic 1 or 0 depending on the difference between the lines, thereby showing the data in the SRAM cell.

The W/L ratios of the transistors have to be sized properly to ensure both read stability and writability. The NMOS pull-down transistor in the cross-coupled inverter must be the strongest, the access transistors of intermediate strength and the PMOS pull-up transistors must be weak. The following table shows the sizes of the transistors used.

Table 1: Sizing of transistors in SRAM cell

Transistor	Width (nm)	Length (nm)
Pull down NMOS	540	180
Pull up PMOS	270	180
Access Transistor	220	180

Read SNM

The stability of the data in the SRAM is usually defined by the SNM. SNM can be defined as the maximum value of DC noise voltage that can be tolerated by the SRAM cell without changing the stored bit.

Figure 3 shows the simulated circuit for the 6-T SRAM cell. For finding the feedback loop from the cross-coupled inverter is broken and the voltage transfer characteristics are made by sweeping the inverter input voltage from, 0 to V_{DD}. This is then used to make the butterfly plot. The obtained butterfly is given in Figure 4.
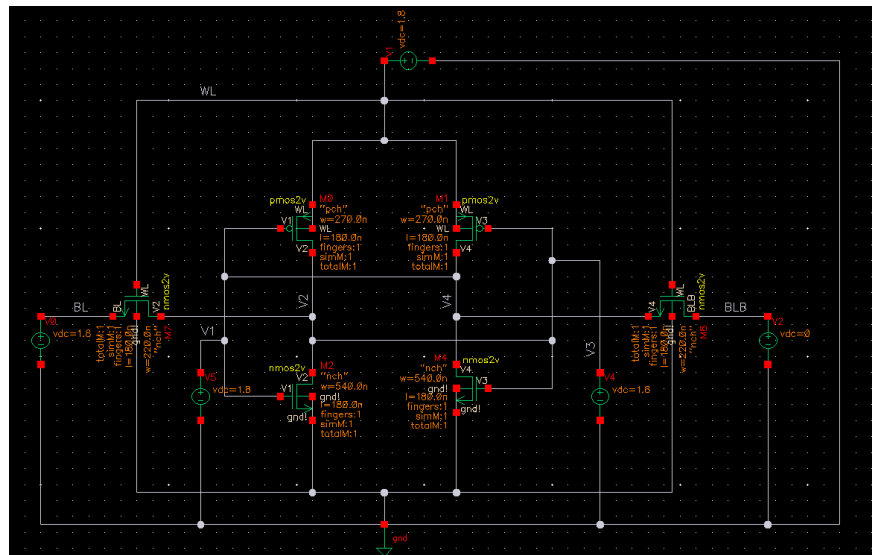
Figure 3: Simulated circuit for 6-T SRAM cell
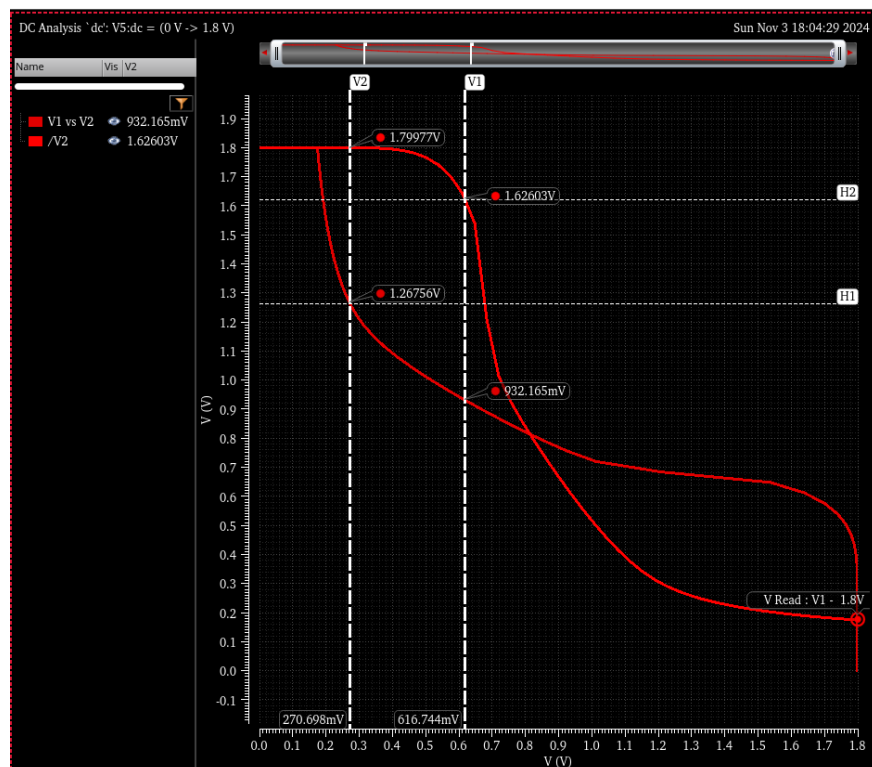
Figure 4: Obtained Butterfly plot for SNM of SRAM cell

Write SNM

To find the write noise margin, the feedback from the cross-coupled inverters is broken, and the VTCs of the two inverters are used to create the plot. The obtained plot for WNM is shown below in Figure 5.
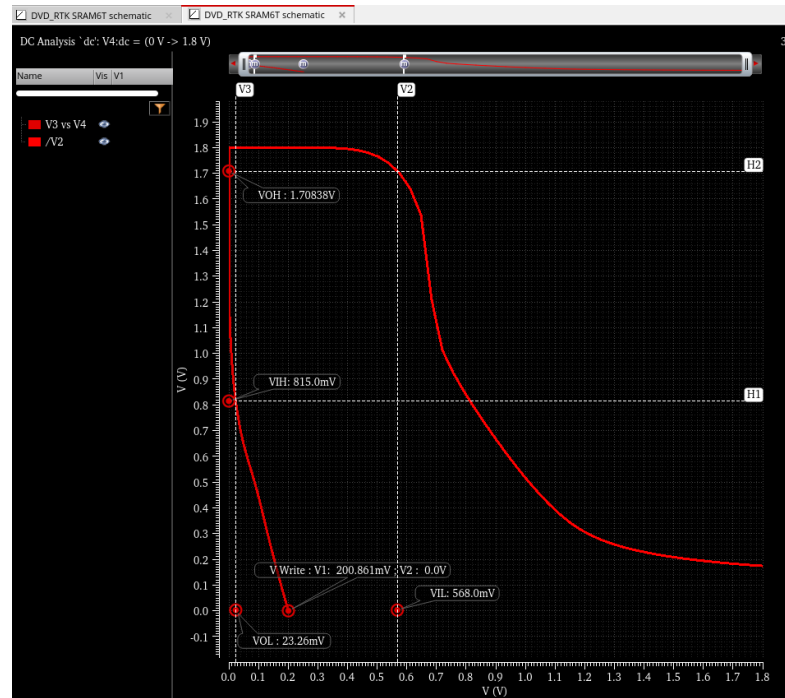

Figure 5: Obtained WNM for SRAM cell

Logic Gates

The digital logic gate serves as the fundamental building block for all digital electronic circuits and microprocessor-based systems. In digital logic design, only two voltage levels or states are used, commonly referred to as Logic "1" and Logic "0," or HIGH and LOW. Logic gates are designed in accordance with the rules of logical effort. A basic inverter gate design typically consists of PMOS and NMOS transistors sized at 2X/1X, where the 1X sizing corresponds to 2μm/180nm.

NOT Gate

The NOT gate is designed to perform the logical NOT operation ($F = \overline{A}$). It is implemented using CMOS technology and the transistors are sized such that the pull-up and pull-down network will have the same resistance and the VTC becomes symmetric. The sizing used are

Table 2: Sizing of transistors in NOT gate

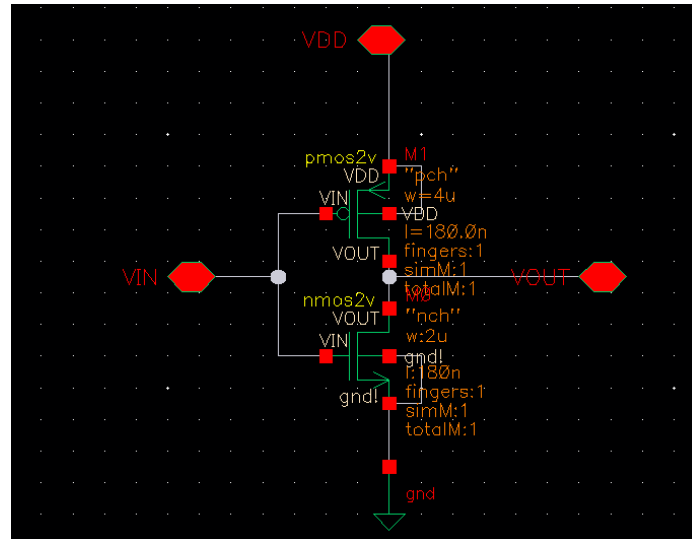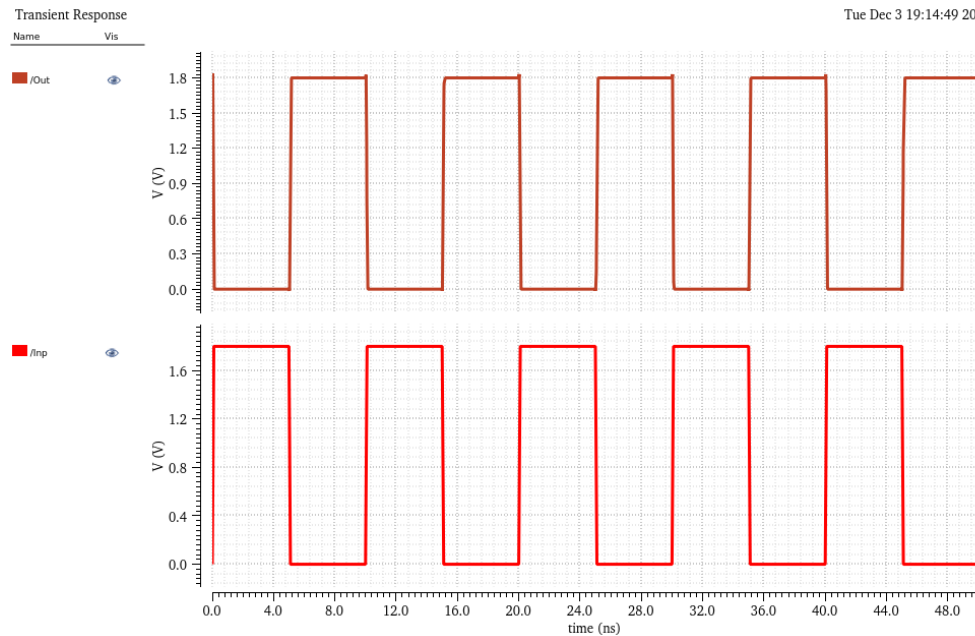| Section	W/L
Pull-up network	4μm/180nm
Pull-down network	2μm/180nm

Figure 6: Simulated Circuit Diagram for NOT gate

Figure 7: Obtained Waveforms for NOT gate

NAND Gate

NAND gate is used to implement logical NAND operation (F = $\overline{AB}$). It is implemented by having two NMOS transistors in series in the pull-down network and two PMOS in parallel in the pull-up network.

Table 3: Sizing of transistors used in NAND gate

Section	W/L
Pull-up network	4μm/180nm
Pull-down network	4μm/180nm

AND Gate

The AND gate is used to perform the logical AND operation (F = AB). It is made by cascading a NAND gate which is easily obtained by CMOS technology and a NOT gate.
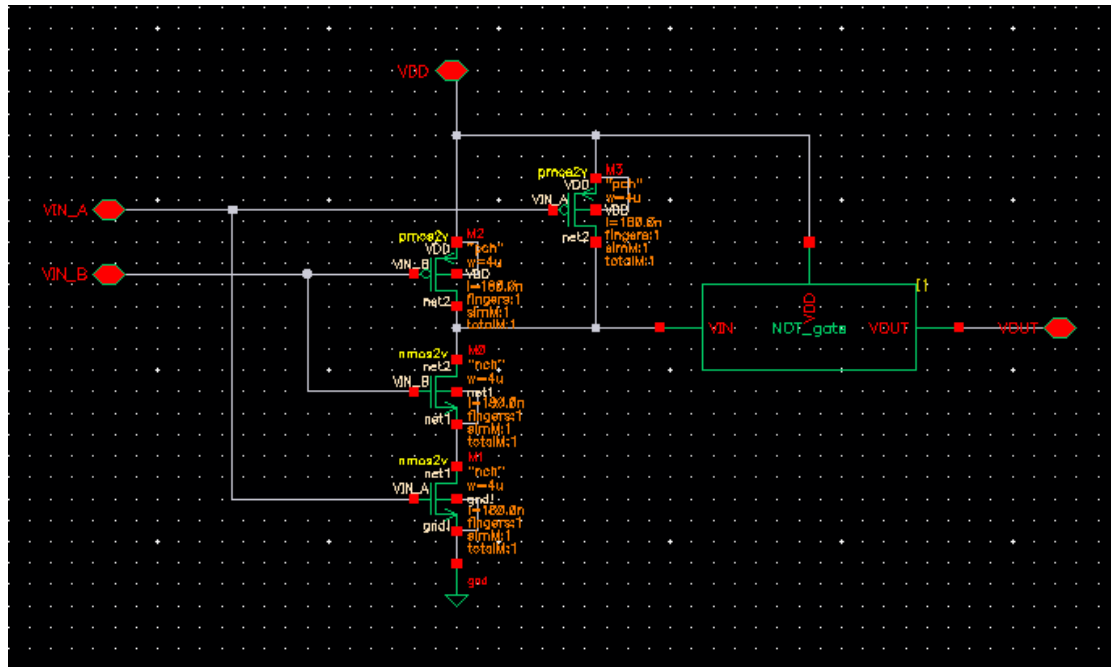

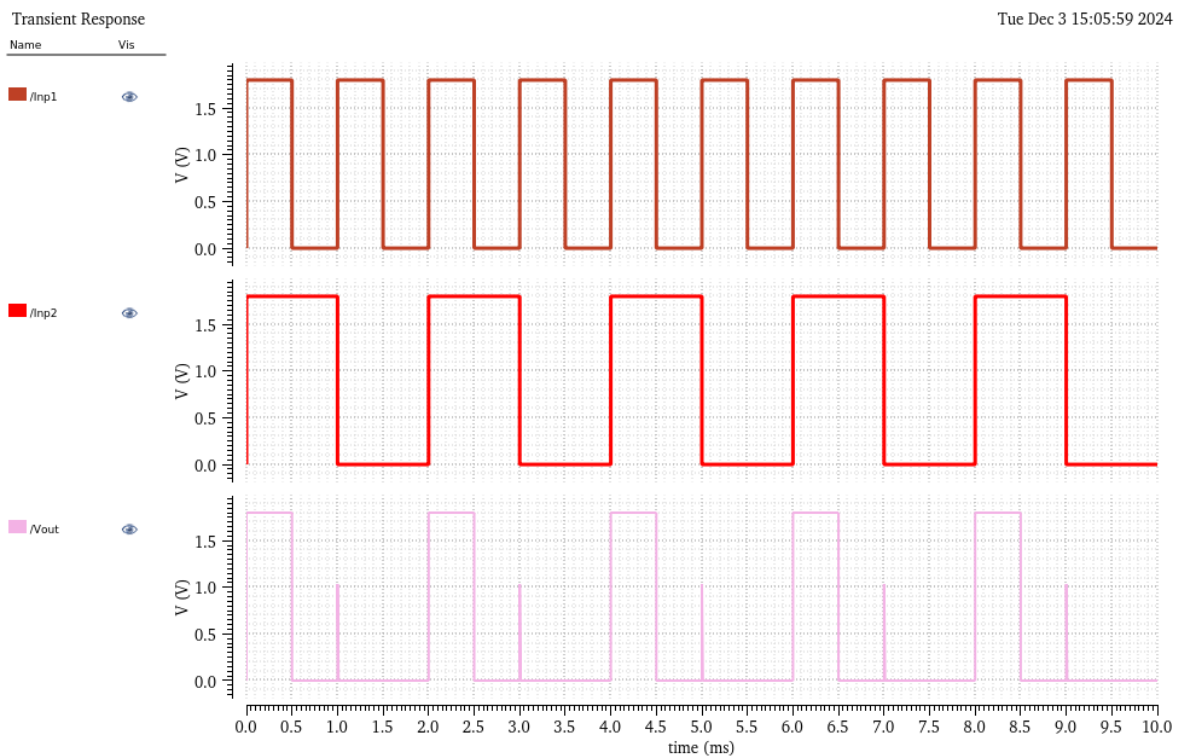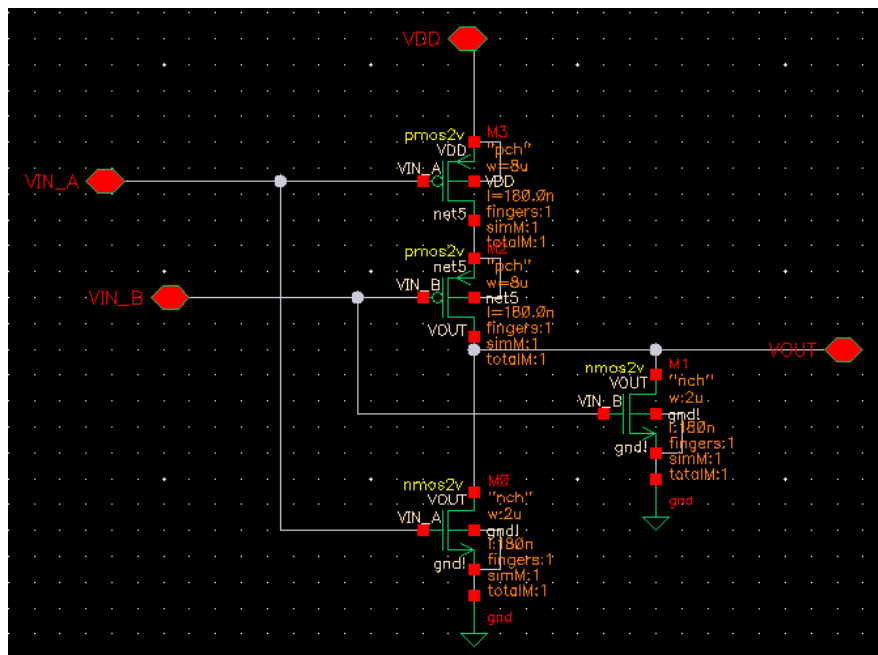*Figure 8: Simulated Circuit Diagram for AND gate*

Figure 9: Obtained waveforms for AND gate

NOR Gate

The NOR gate is used to perform the logical NOR operation (F $= \overline{A + B}$). It is made by having two NMOS in parallel in the pull-down network and 2 PMOS in series in the pull-up network. The sizes of the transistors are shown below:

Table 4: Sizing of transistors used in NOR gate

Section	W/L
Pull-up network	8μm/180nm
Pull-down network	2μm/180nm

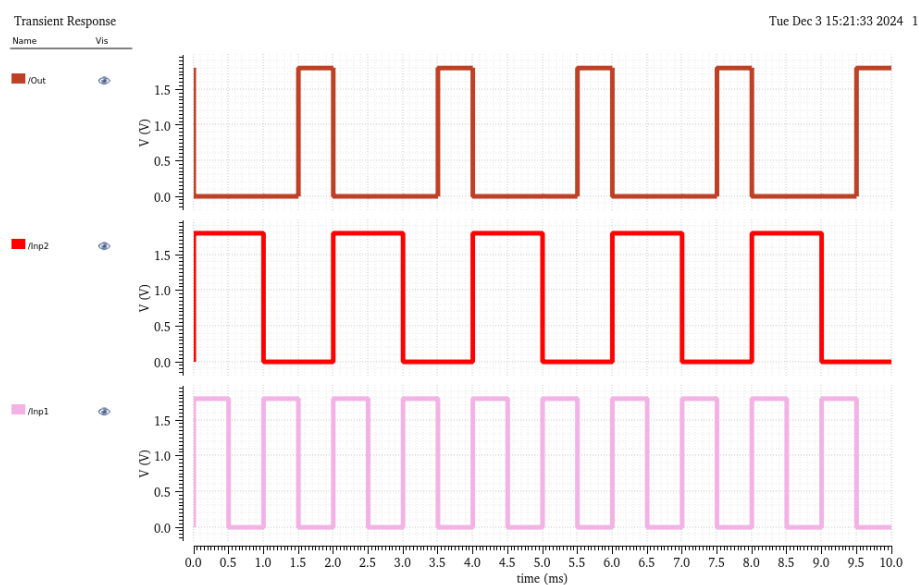*Figure 10: Simulated Circuit Diagram for NOR gate*

Figure 11: Obtained waveforms for NOR gate

OR Gate

OR gate is used to perform logical OR operation (F = A +B). It is implemented by cascading a NOR gate and a NOT gate.
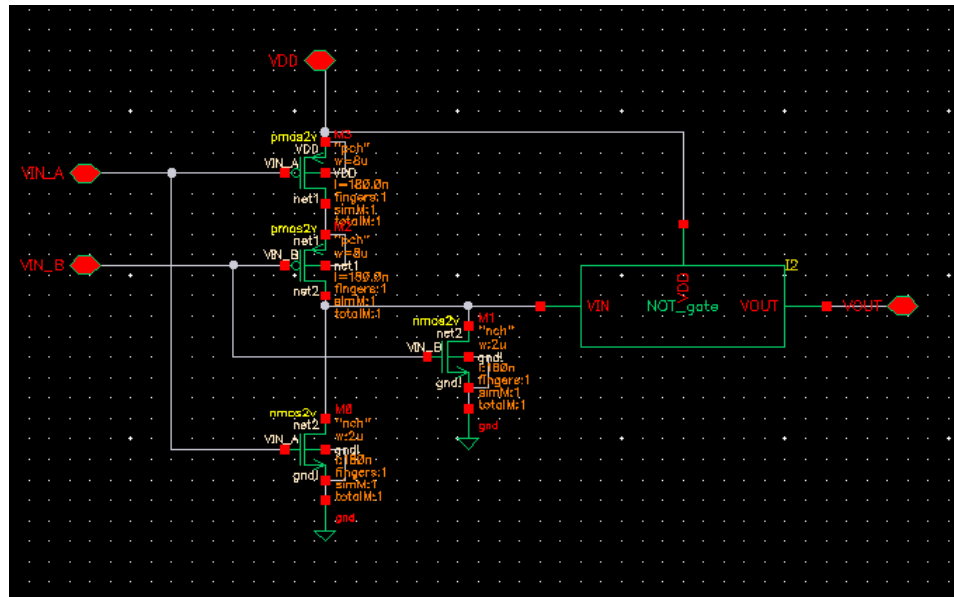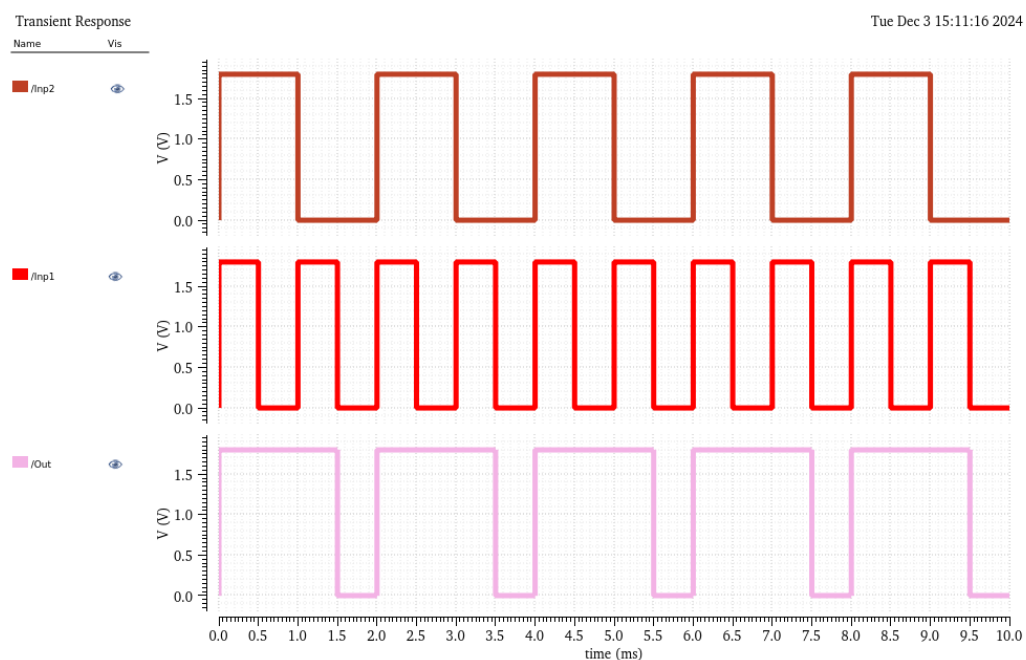

Figure 12: Simulated Circuit Diagram for OR gate

Figure 13: Obtained waveforms for OR gate

XOR Gate

The XOR gate implements the EXCLUSIVE OR function. It is made to implement the expression F = $A\overline{B} + \overline{A}B$. The sizing of the transistors are shown below:

Table 5: Sizing of transistors used in NAND gate

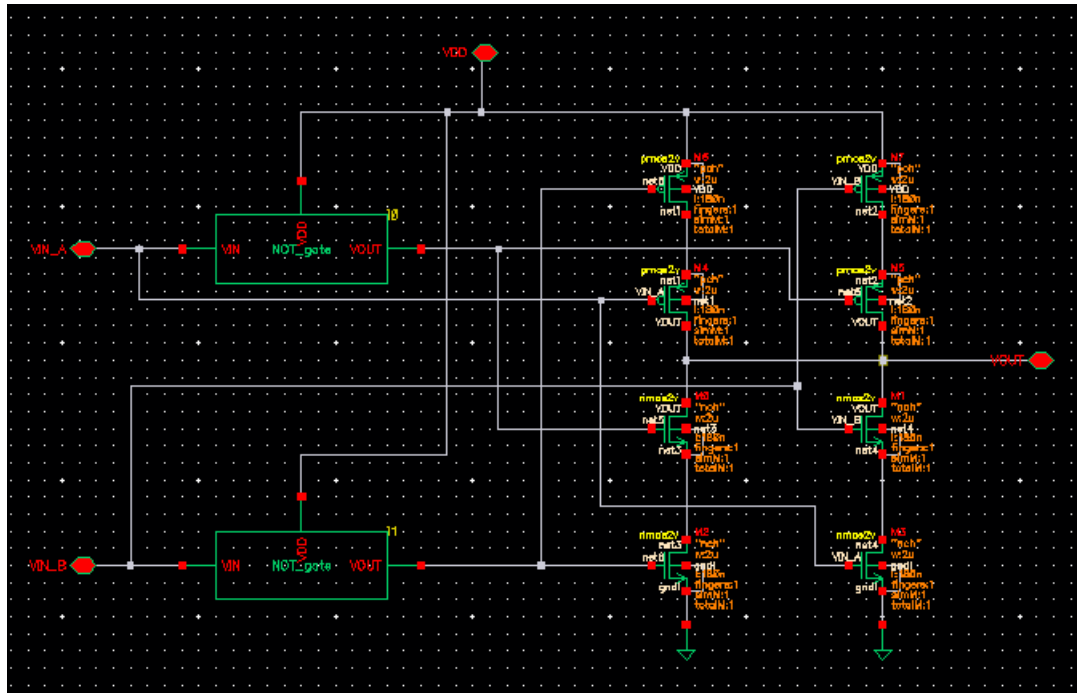| Section	W/L
Pull-up network	8µm/180nm
Pull-down network	4µm/180nm

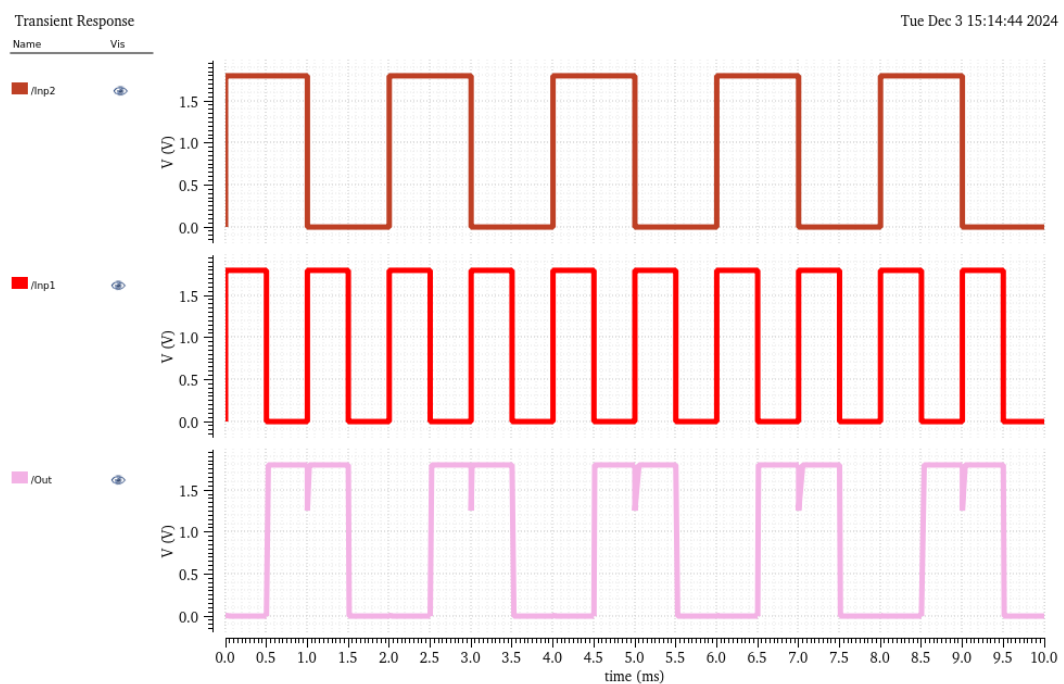*Figure 14: Simulated Circuit Diagram for XOR gate*

Figure 15: Obtained Waveforms for XOR gate

Adder Circuits

The 5-Bit Adder adds a 3-bit offset address to a 5-bit relative address to calculate the effective address.

Half-Adder

A half-adder is used to add 2-bit binary numbers. The Half adder is implemented using an XOR gate to generate the sum and an AND gate to generate the carry.
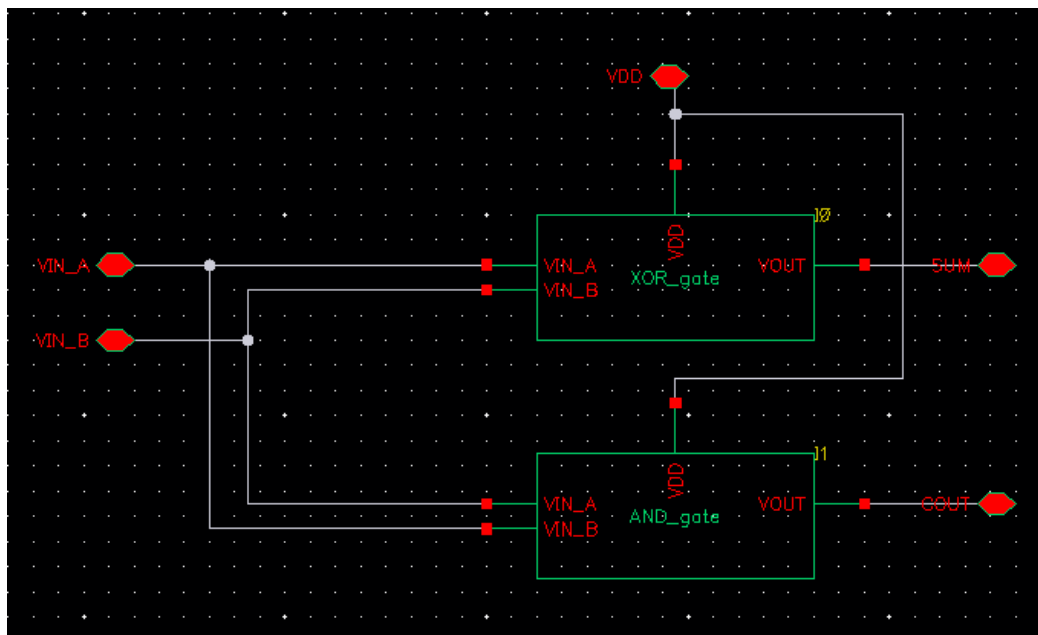

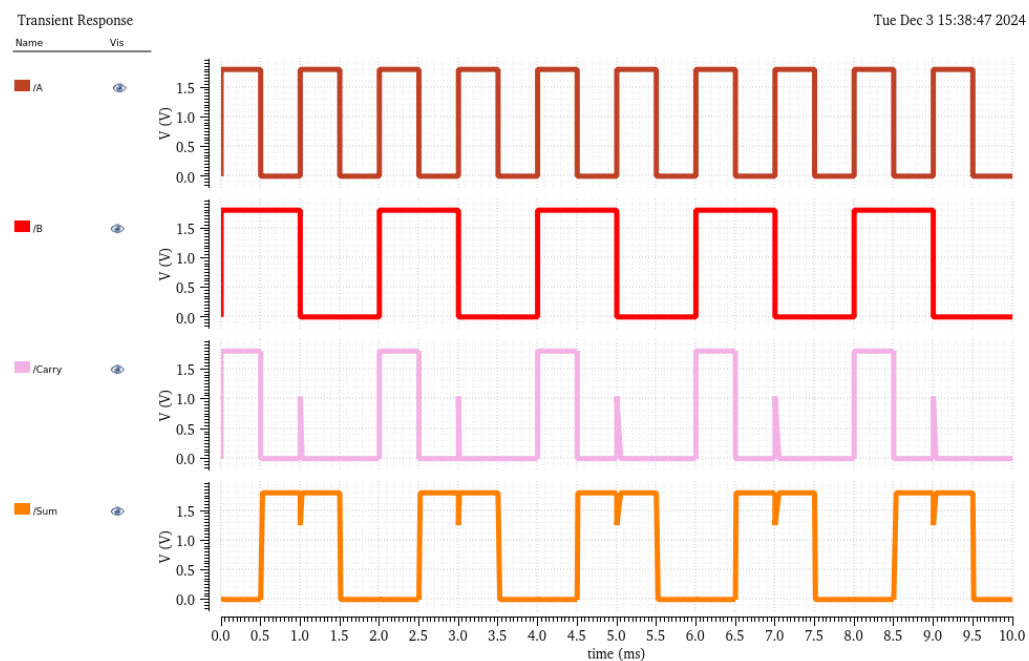*Figure 16: Simulated Circuit Diagram for Half Adder*

Figure 17 Obtained waveforms for Half Adder

Full-Adder

A full adder is used to account for a carry-in when adding two binary numbers. So a full adder adds 3 bits. It is implemented by using two half-adders. Two of the inputs are given to a half adder and the sum is taken as one input for the next half adder. The sum of the second half adder will give the final sum. To obtain the final carry-out, the carry-outs from both half-adders are given to an OR gate.
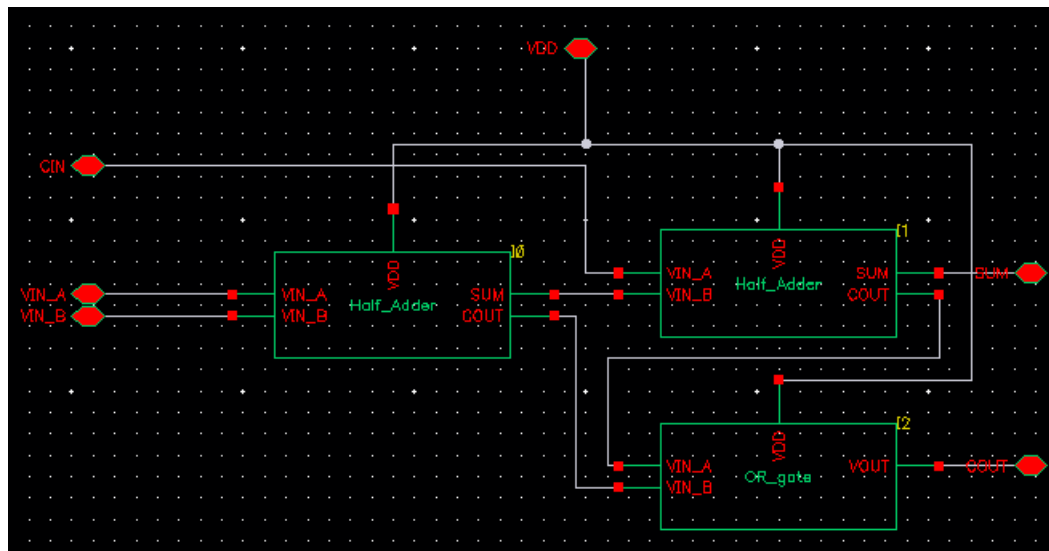

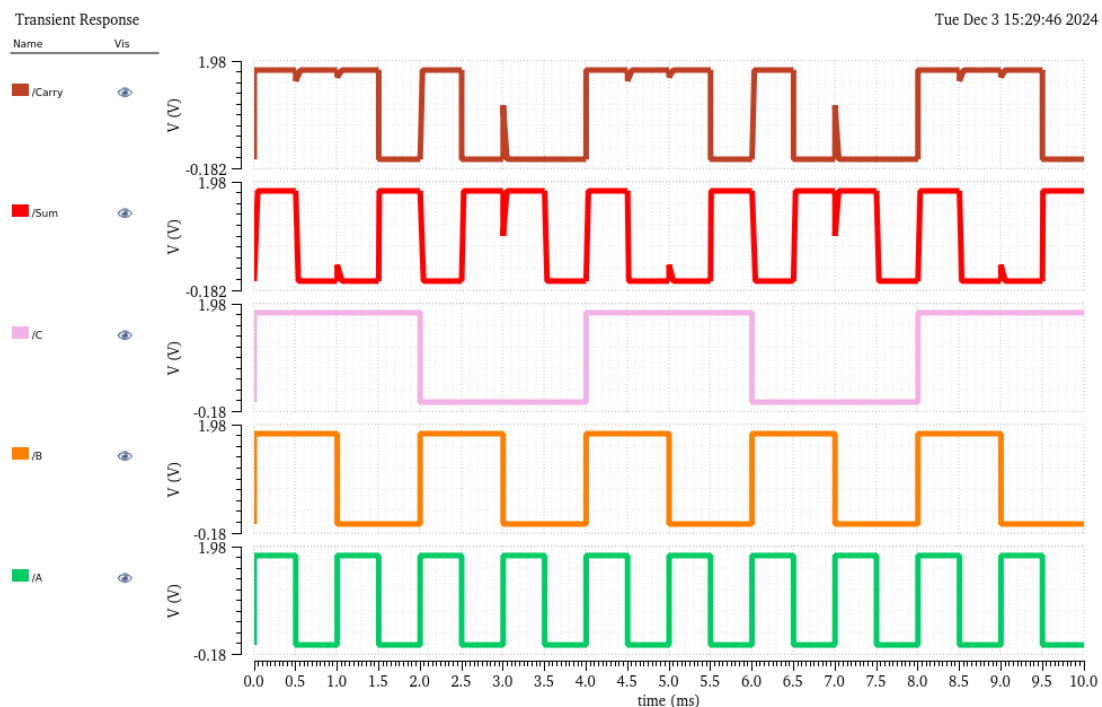*Figure 18: Simulated Circuit Diagram for Full Adder*

Figure 19: Obtained Waveforms for Full Adder

5-Bit Adder

A 5-bit adder is required for adding the 5-bit relative address and the 3-bit offset to get the effective address. The 5-bit adder is implemented using a combination of half adders, full adders and an XOR gate. From the observation that the relative address varies from 0 to 24 and the offset varies from 0 to 7, it can be noted that the LSB and the second bit from the left only add 2 bits. Hence a half adder is sufficient in those places. For the MSB, the carry-out need not be generated as it is not required for the operation of any downstream circuits, so only the sum needs to be generated. Hence an XOR is sufficient in the MSB position.
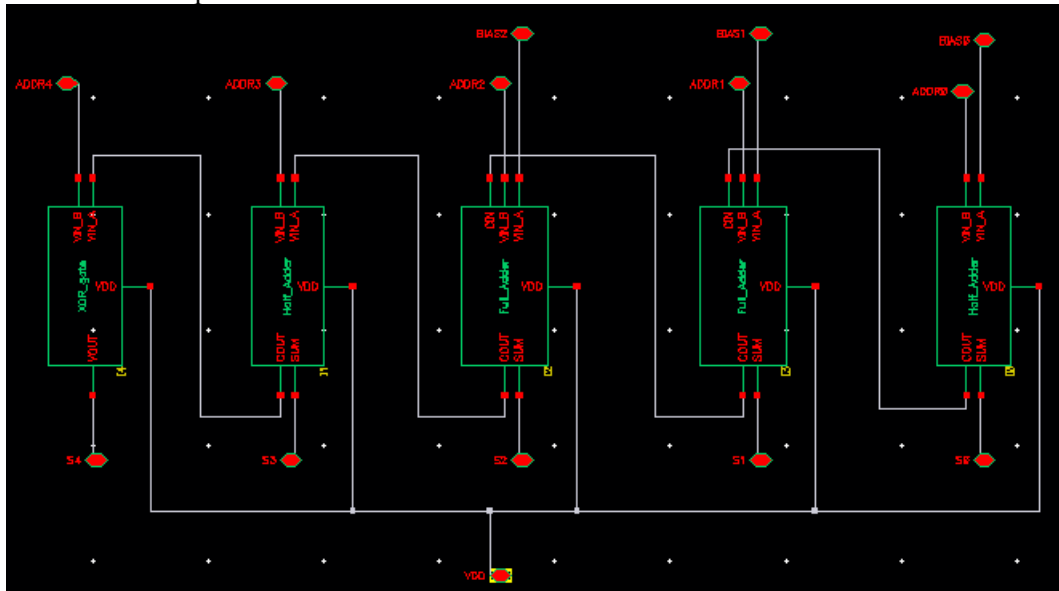

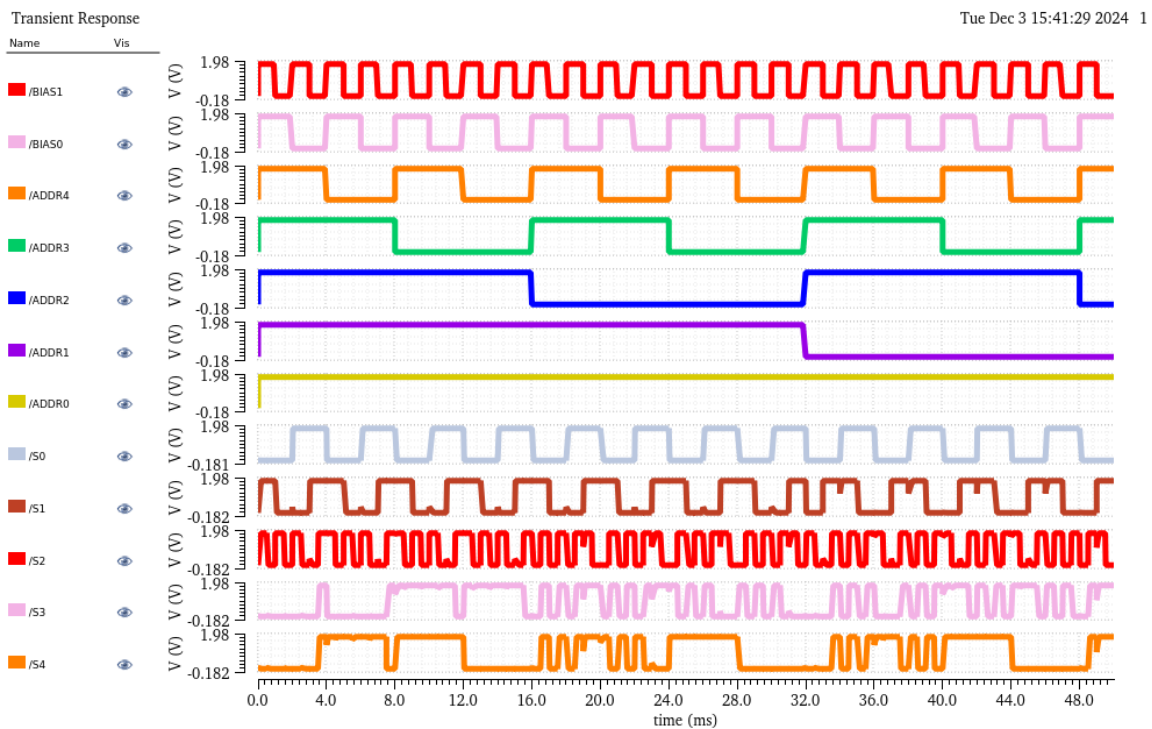*Figure 20: Simulated Circuit Diagram for 5-bit Adder*

Figure 21: Obtained waveforms for 5-bit Adder

Decoder Circuits

Decoders are used to select the appropriate word based on the address provided by the adder. Since the decoding required pre-decoding, the 5 to 32 decoder is implemented using a 2 to 4 decoder and four 3 to 8 decoders.

2 to 4 Decoder

A 2 to 4 decoder was implemented using NOR gates and NOT gates rather than the conventional NOT gates and AND gates on area and transistor count considerations. This implementation of the 2 to 4 decoder will have a transistor count of $(2×2) + (4×4) = 20$, while a conventional implementation will contain $(2×2) + (4×4) + (4×2) = 28$ transistors.
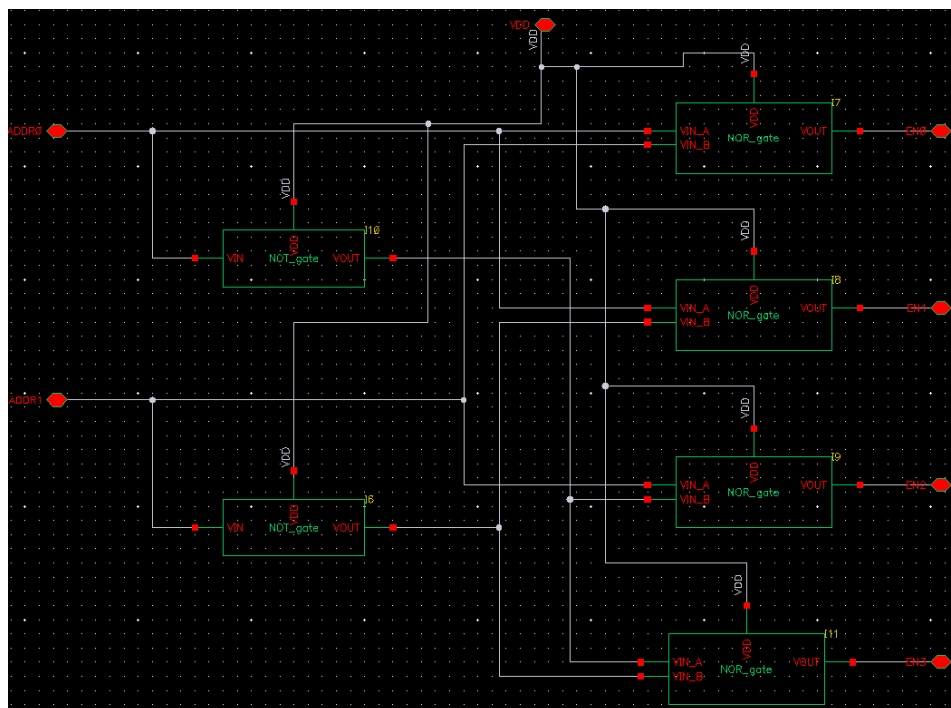

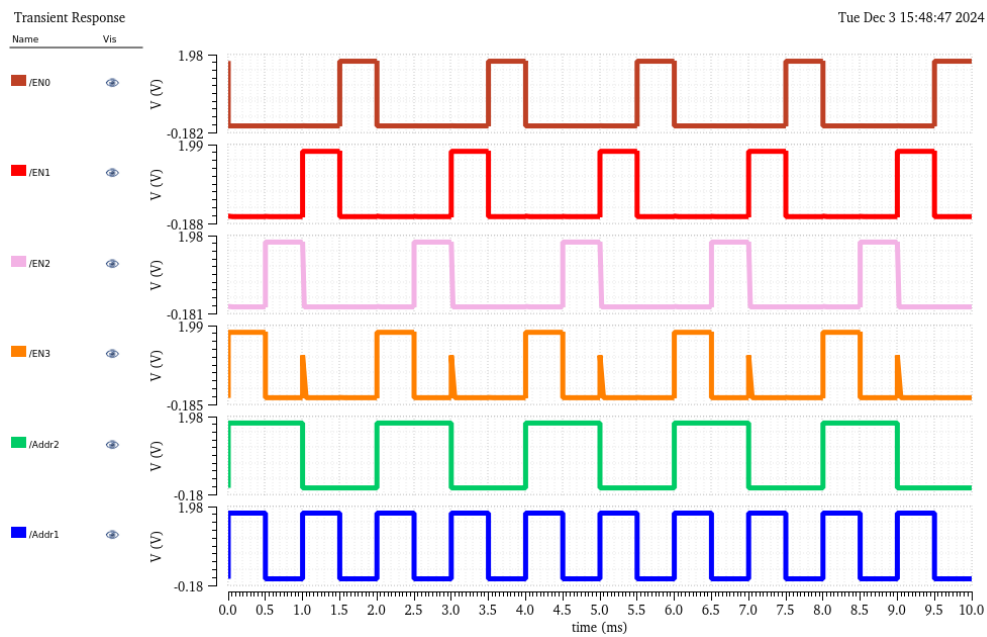*Figure 22: Simulated Circuit Diagram for 2 to 4 Decoder*

Figure 23: Obtained Waveforms for 2 to 4 Decoder

3 to 8 Decoder

A 3 to 8 decoder is also implemented using NOT gates and NOR gates. The circuit diagram and the obtained waveforms are shown below.
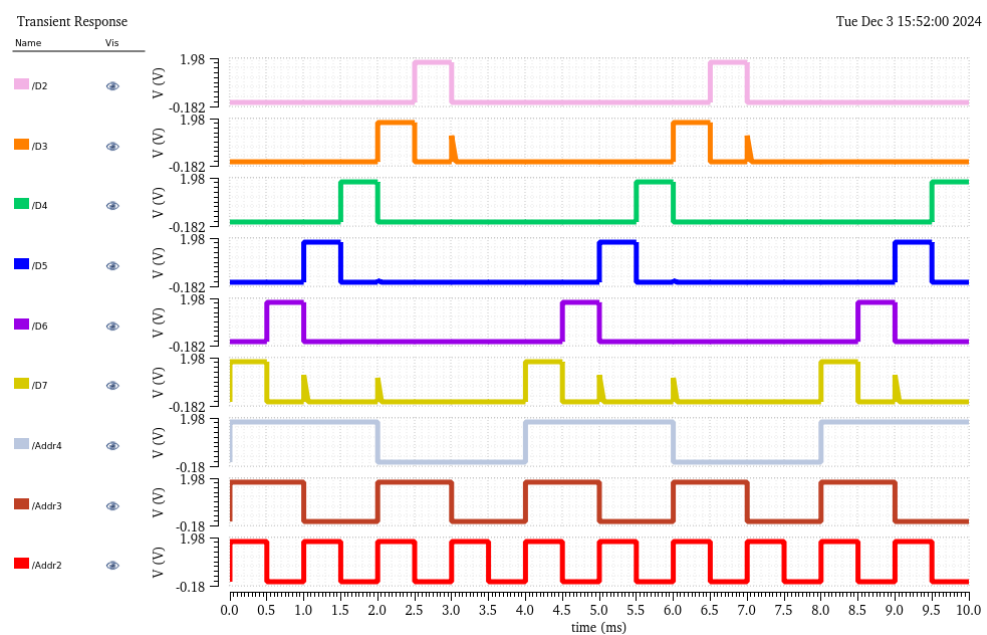

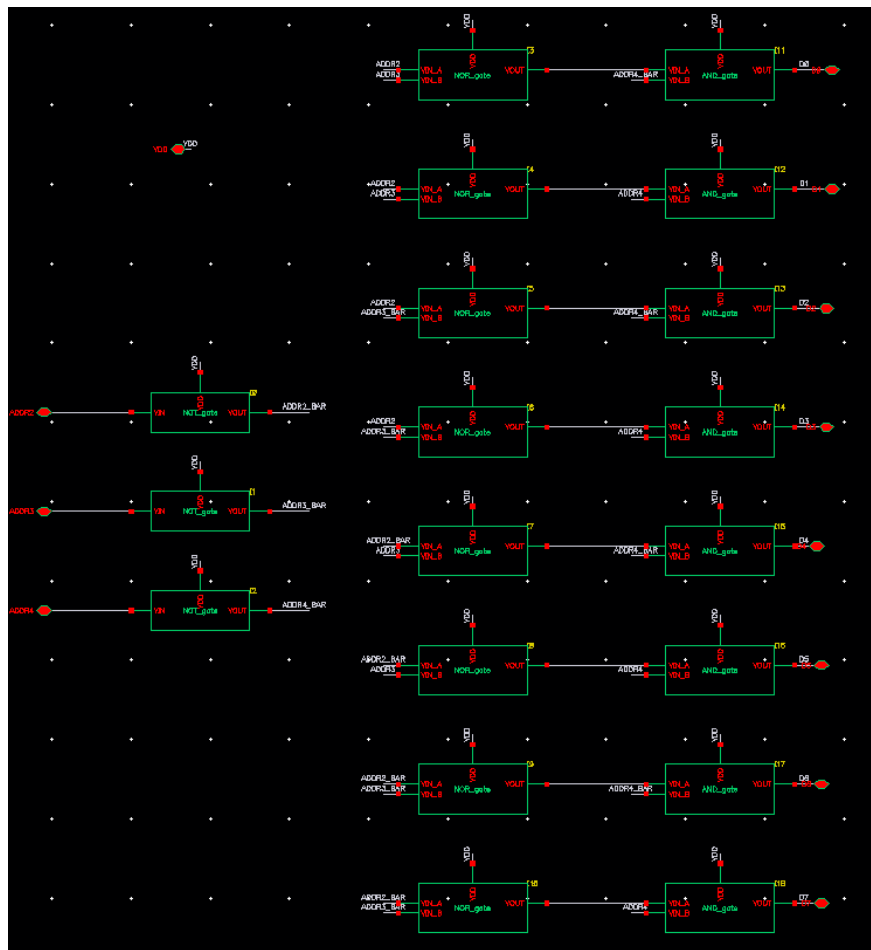*Figure 24: Obtained Waveforms for 3 to 8 Decoder*

Figure 25: Simulated Circuit Diagram for 3 to 8 Decoder

5 to 32 Decoder

A 5 to 32 decoder was implemented using one 2 to 4 and four 4 to 8 decoders. This is used to select the proper word line in the SRAM array using the output of the 5-bit adder.
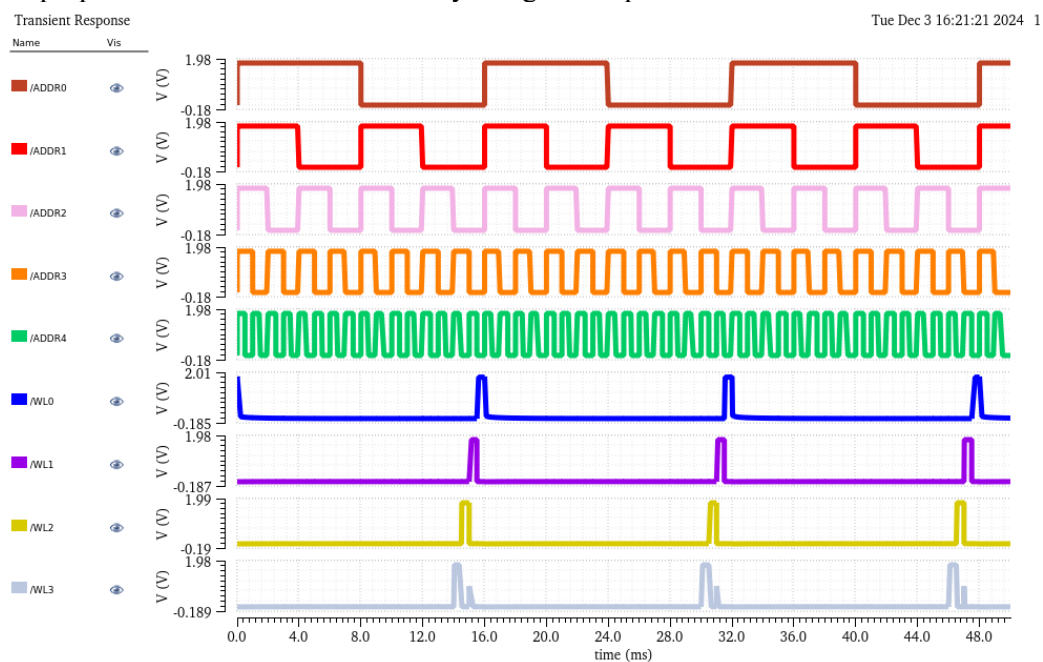

Figure 26: Obtained Output Waveforms for 5 to 32 Decoder
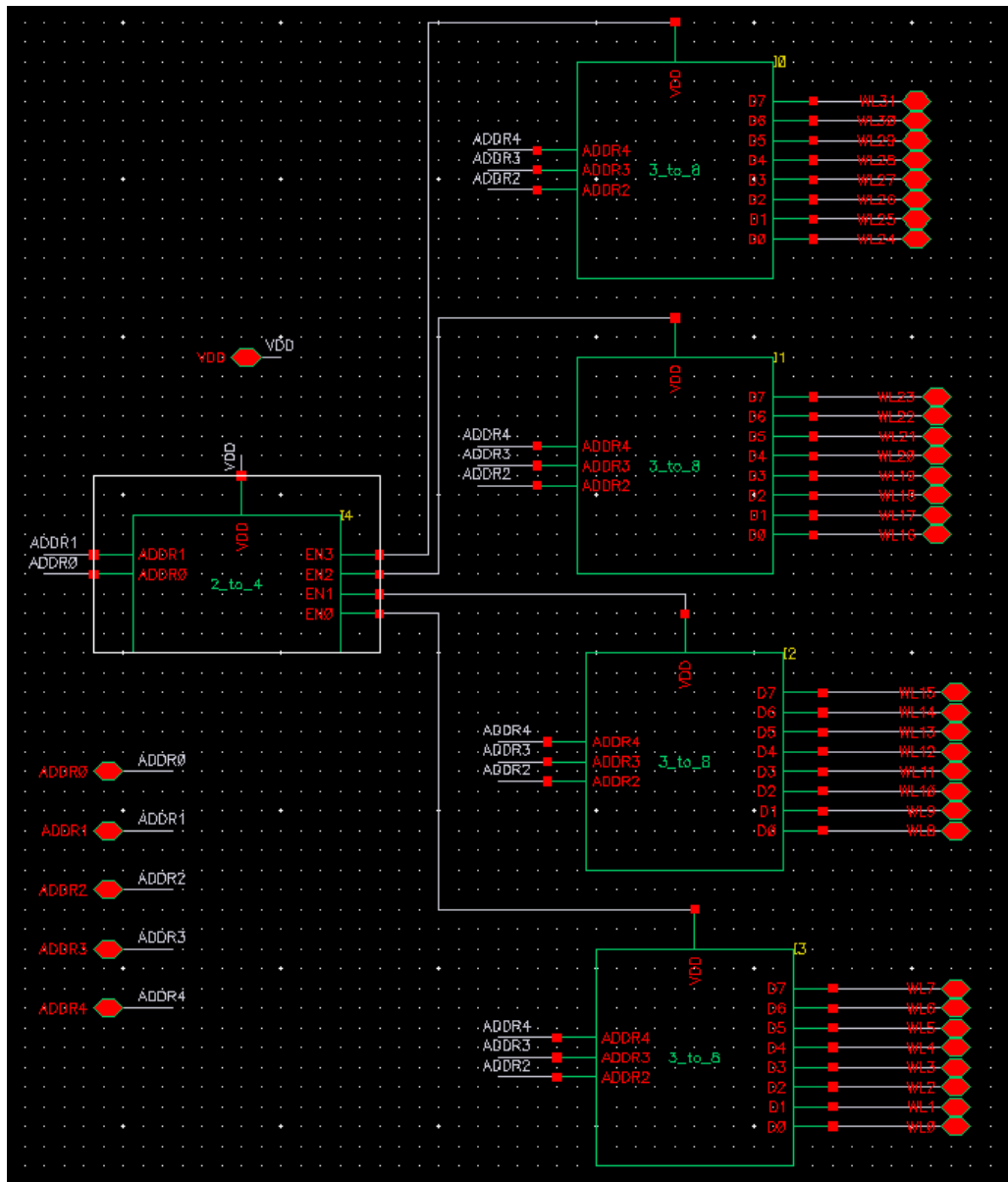
Figure 27: Simulated Circuit for 5 to 32 Decoder

Precharge Circuit

The lines BL and $\overline{\text{BL}}$ need to be pre-charged to V_{DD} before each ready cycle. A dedicated precharge circuit is required to implement this. The pre-charging circuit used in this project is a simple PMOS transistor connected to V_{DD}. The gate of this PMOS is connected to a pin labelled PE which stands for Precharge Enable. This is an active low pin, when this pin is given 0, the PMOS turns on and the line capacitance gets charged to V_{DD}. Having a single PMOS as a pre-charging circuit will help in reducing the transistor count and area.
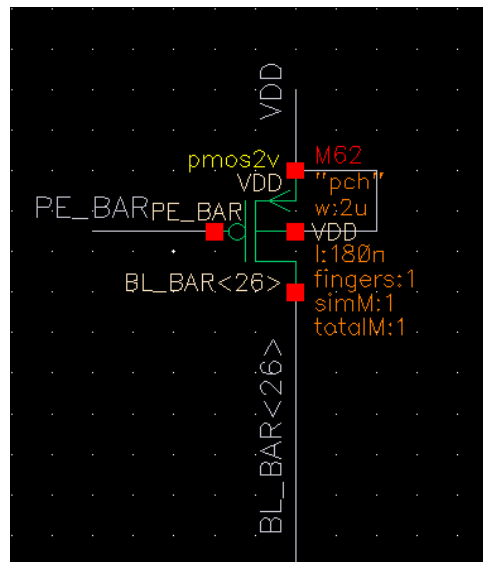
Figure 28: Pre-charge Circuit

Sense Amplifier

The sense amplifier defines the robustness of the bit-line sensing, impacting the read speed. This amplifies the differential output voltage between BL and $\overline{\text{BL}}$. A 5-T OTA is implemented as a differential amplifier for this purpose. Having a sense amplifier will help in reducing the size of the storage cell, as each individual cell need not completely discharge the bit line.

Figure 29: Simulated Circuit Diagram

Figure 30 Obtained Waveforms for Sense Amplifier

Integration and Evaluation

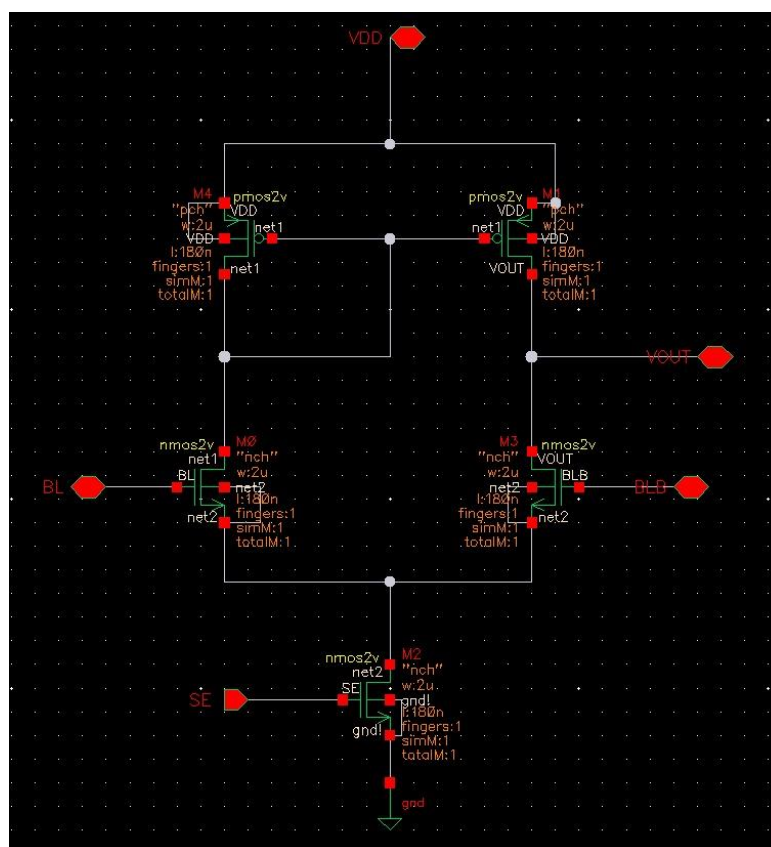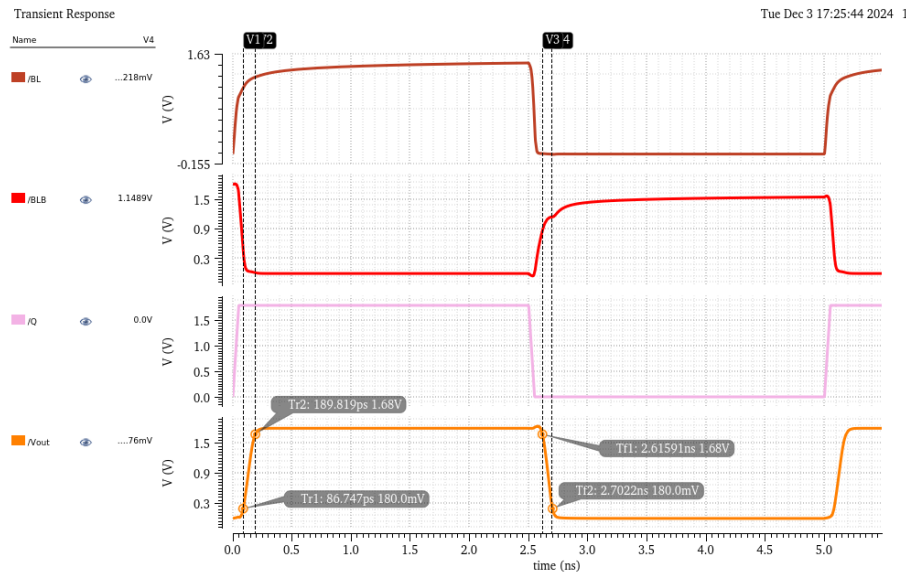All individual blocks namely- the 5-bit adder, 5 to 32 decoder, 32×32 SRAM a, the pre-charge circuit, and the sense amplifier were integrated together to make a working SRAM memory unit. During the read operation in the 6-T SRAM, both the bit lines are precharged to V_{DD} using the precharge circuit. This is done by setting PE to a low voltage. The relative address [ADDRB0:ADDRB4] and the 3-bit bias [BIAS0:BIAS2] to get the 5-bit effective address. The 5-bit [ADDR0:ADDR4] effective address is given as input to a 5:32 decoder. The decoder output determines the row to be selected. The output line that goes high corresponds to the word line (WL), which will be high. Once the bit lines are precharged, the word line is set to high. Sense enable(SE) is also set to high.

Let us assume that Q is set to "0". On the left part of the bit cell, we have $V_{DS} = V_{DD}$, i.e., there is some current flow in that part. The NMOS circuitry of the SRAM cell acts as a path for the Bit line (BL) capacitance to discharge. Hence the voltage on BL decreases little by little. The bit lines are then passed to an output buffer to drive capacitance loading the output of the memory cell. Both bit lines are taken and sent to the sense amplifier; here, the sense amplifier acts as the comparator, and the output of SA is "0". If Q is set to be "1", then the voltage in the right section (BL) of the cell decreases, and the output of SA shown will be "1". Here, the sense amplifier amplifies the output by restricting all the noise caused.
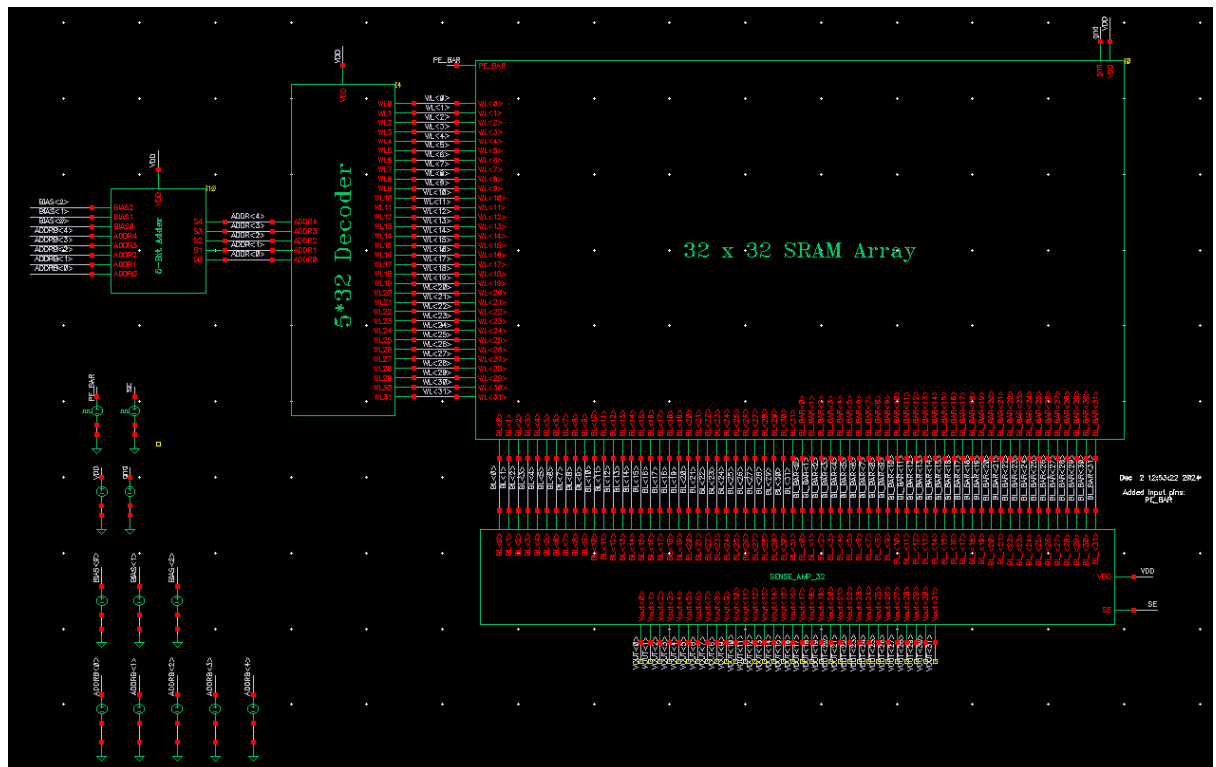
Figure 31: Final Circuit

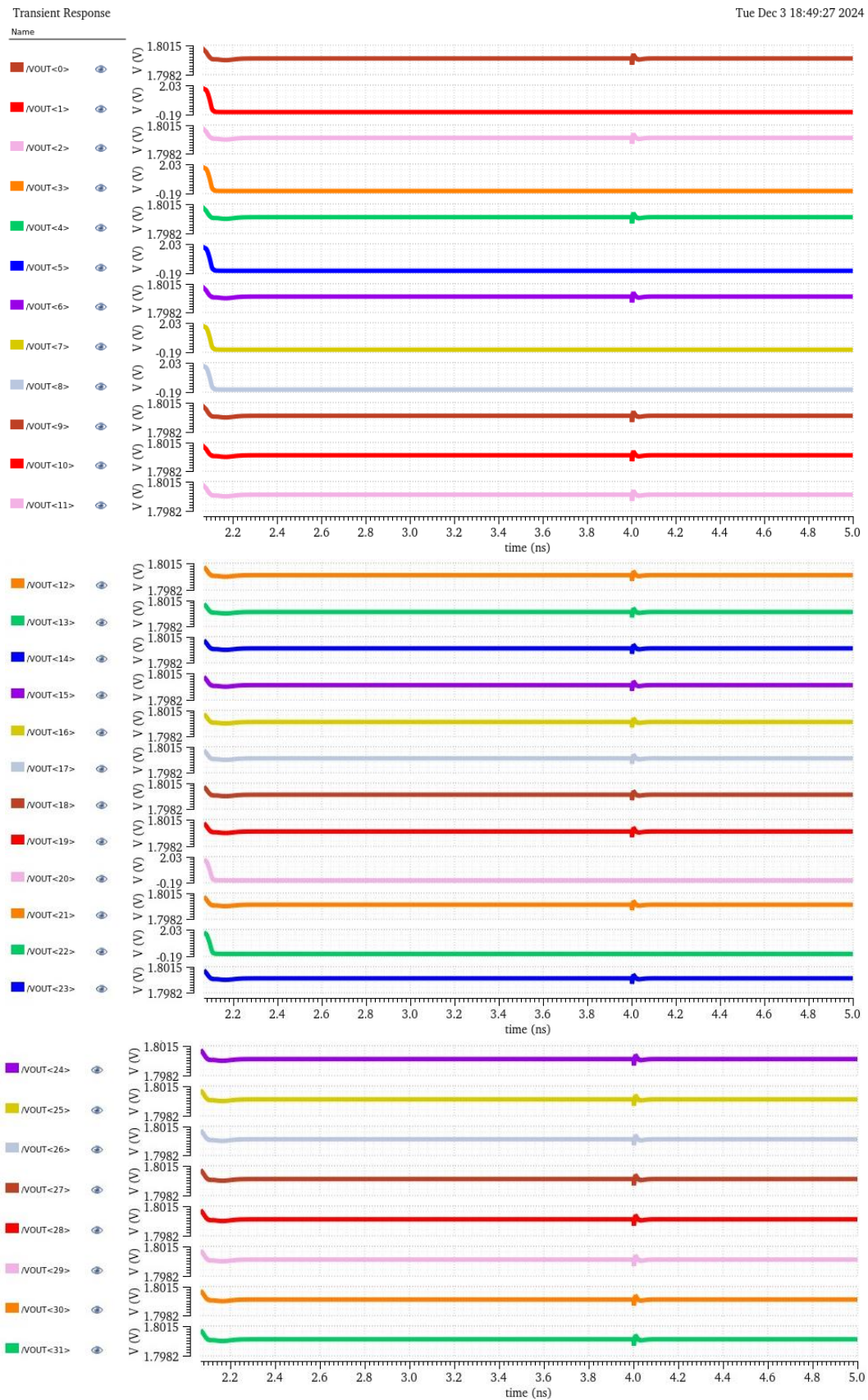*Figure 32: Data Initialised in the Word*

Figure 33: Obtained Waveforms from SRAM Read Operation

Table 6: Initialised data in word 0 before the read operation

Cell name	Q0	Q1	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
Value	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31
1	1	1	1	1	1	1	1	1	1	1	1	1

Table 7: Data read from word after read operation

Cell name	Q0	Q1	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
Value	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31
1	1	1	1	1	1	1	1	1	1	1	1	1

Results

A 32×32 SRAM array was implemented capable of successfully completing read operations. The following parameters of the SRAM array were measured:

Table 8: Obtained Results

Parameters	Value
Delay	239ps
Rise time	103ps
Fall time	86ps
Power during Read operation	19.79mW

References

1. Digital Integrated Circuits- A Design Perspective, Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikolic, 2e, Pearson Publications, 2003
2. N. Weste and D. Harris, CMOS VLSI Design: A circuits and systems perspective. Pearson India, 2015
3. Digital IC design Lectures by B. Mazhari