Abstract

Static Random Access Memory (SRAM) serves as a fundamental building block in modern digital systems, finding widespread use in embedded applications, cache memories, and microprocessors. Large SRAM arrays, particularly those employed as cache in high-performance processors and application-specific integrated circuits (ASICs), occupy a substantial portion of the chip area. Industry projections suggest that SRAM may account for more than 90% of the total System-on-Chip (SoC) area within the next decade.

To enhance system performance, designers rely on dense arrays of high-speed SRAM. However, the integration of such large memory blocks significantly impacts chip area and cost, creating a critical need to minimize the footprint of individual SRAM cells. Consequently, millions of minimum-sized cells are tightly packed, making SRAM arrays the most area-intensive structures on a chip.

This work presents the design and analysis of a 16×16 SRAM memory array implemented using Cadence Virtuoso toolset. The design targets high-speed memory applications such as cache, with emphasis on three key parameters: access time, operating speed, and power consumption

Introduction

The rapid growth of modern communication systems, signal processing technologies, and handheld consumer electronics has created a strong demand for compact and efficient computing solutions. System-on-Chip (SoC) designs have emerged as a key enabler of cost and size reduction by integrating memory, digital computing, and signal processing circuits onto a single chip. In these designs, memory occupies nearly 70% of the die area, with static random-access memory (SRAM) being the dominant component. Consequently, technology selection and design choices in SoCs are largely driven by digital circuit requirements.

SRAM plays a critical role in high-performance VLSI circuits and mobile products. It is widely used for cache memory in microprocessors, mainframe computers, engineering workstations, and handheld devices due to its high speed and low power consumption. The increasing reliance on SRAM has made efficient memory design a central challenge in modern SoC development.

Design Methodology

This project employs the Cadence Virtuoso tool for SRAM design. The proposed design focuses on a 16*16 SRAM array, constructed using the basic 6T SRAM cell. To support memory operations, essential peripheral circuits are implemented, including:

- Row Decoder
- SRAM
- Pre-charge Circuit
- Write Driver
- Sense Amplifier

The objective is to demonstrate successful read and write operations while achieving low power consumption and reduced access time, thereby validating the feasibility of low-power SRAM design.

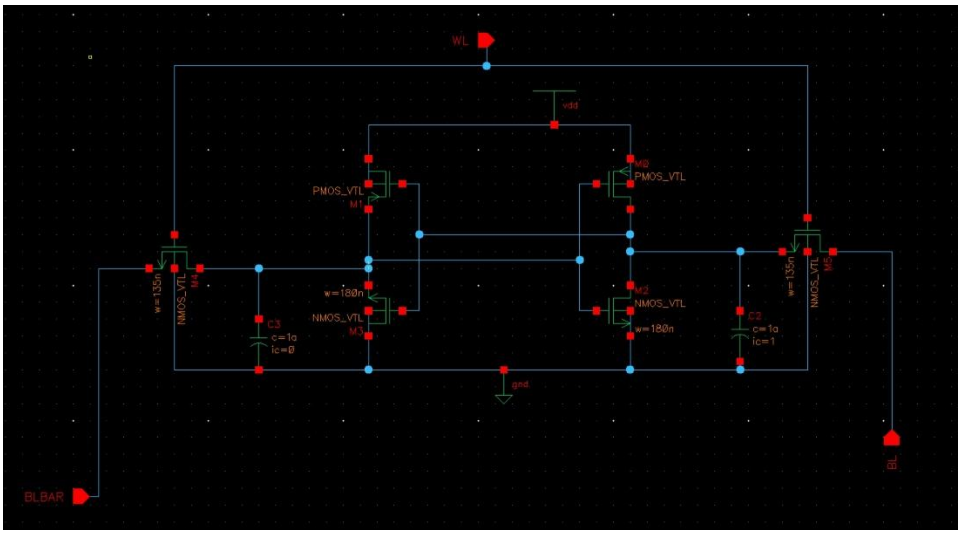
Background

Previous research, such as *"A Single-Ended 6T SRAM Cell Design for Ultra-Low Voltage Applications"* (IEICE, 2008), highlights how advances in transistor scaling have enabled faster circuits, smaller die areas, and higher clock rates. These developments have paved the way for compact, high-speed SoCs with increased integration levels.

Semiconductor memory arrays are fundamental to digital systems, storing large volumes of information essential for computation. The number of transistors dedicated to memory storage far exceeds those used for logic operations. Driven by the demand for higher capacity, fabrication technologies have continually evolved toward denser designs. As a result, single-chip memory capacity has reached 1 gigabit (1 Gb), with storage density approximately doubling every two years. This trend is expected to continue, pushing the boundaries of digital system design.

SRAM architecture

The main SRAM building blocks are as follows • SRAM cell. • Pre-Charge Circuit. • Write Driver Circuit. • Sense Amplifier. • Row decoder.

- The SRAM cell
- 

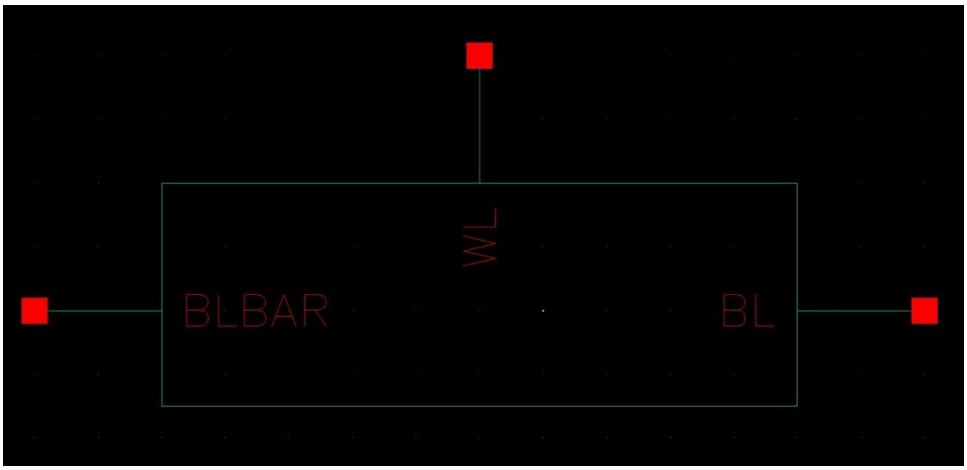
- Fig. 3: Schematic of SRAM circuit.
- 

Fig. 3: Schematic of SRAM symbol

The design of an SRAM cell is a critical aspect of modern memory architecture, as it directly influences the stability, speed, power efficiency, and overall yield of the system. A typical SRAM cell consists of two cross-coupled inverters forming a latch, along with access transistors that control read and write operations. When the word line is low, the access transistors remain off, isolating the cell; when the word line is high, the transistors switch on to enable data transfer. This structure ensures non-destructive read access, reliable write capability, and continuous data retention as long as power is supplied. Careful transistor sizing is essential to balance performance and robustness, with PMOS, NMOS, and access transistors dimensioned to optimize packing density while maintaining operational stability. These considerations make SRAM cell design a cornerstone of efficient System-on-Chip (SoC) implementations, where memory occupies the majority of chip area and directly impacts overall system performance.

The Pre-Charge Circuit

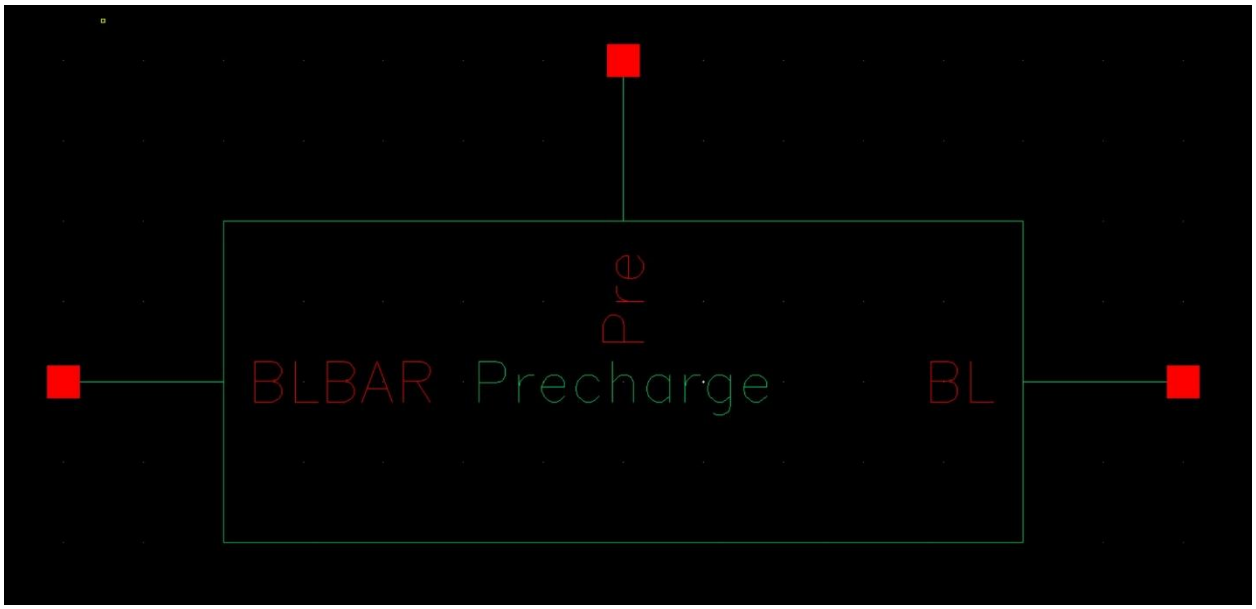


Fig. 3: Schematic of Pre-Charge symbol.

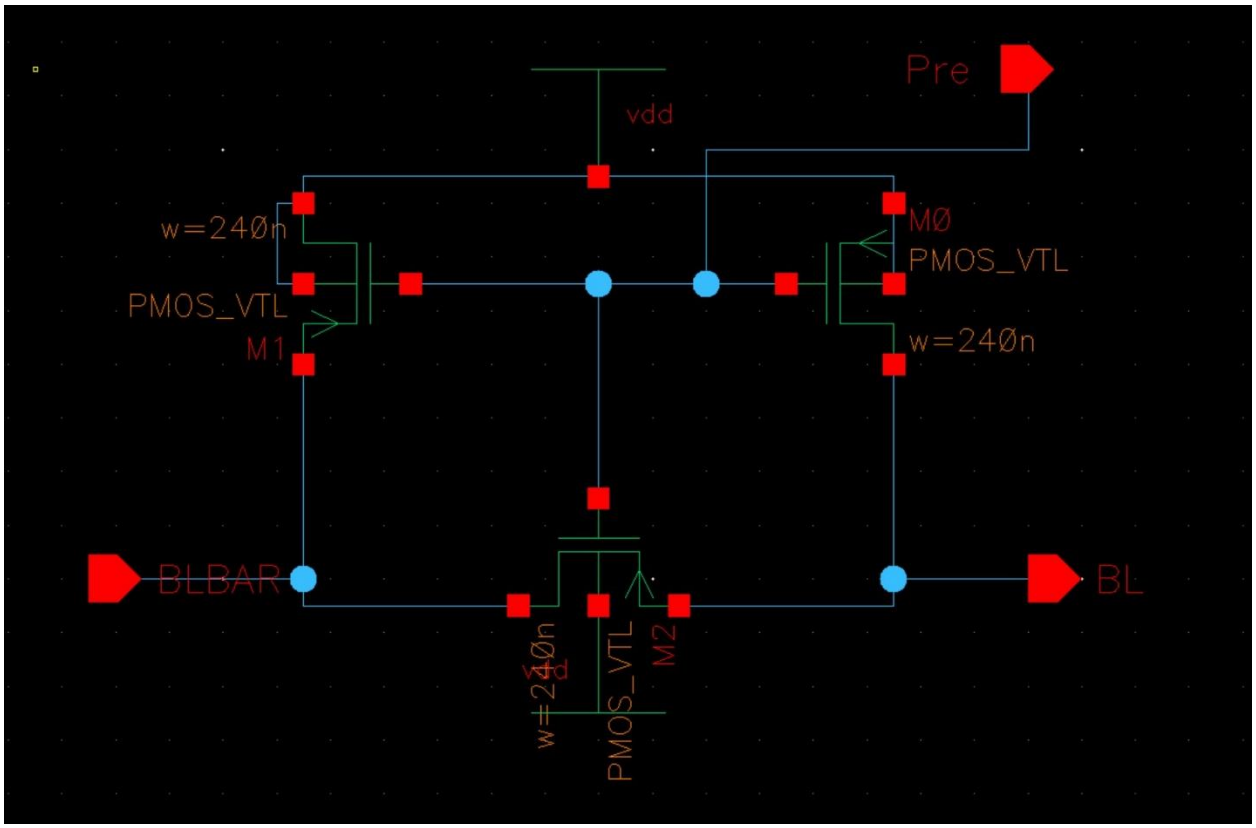


Fig. 3: Schematic of Pre-Charge circuit.

The pre-charge circuit is an essential element in SRAM design, as it ensures that the bit line and bit-bar lines are initialized to the supply voltage before any read or write operation. In the proposed circuit, the bit lines are charged to Vdd = 1 V, maintaining them at a high level during idle states. This setup allows faster access and reliable operation, since the pre-charge is disabled only during active read and write cycles. The PMOS transistor used in the pre-charge circuit is designed with a minimum width of 240 nm providing efficient charging characteristics.

Write Driver Circuit

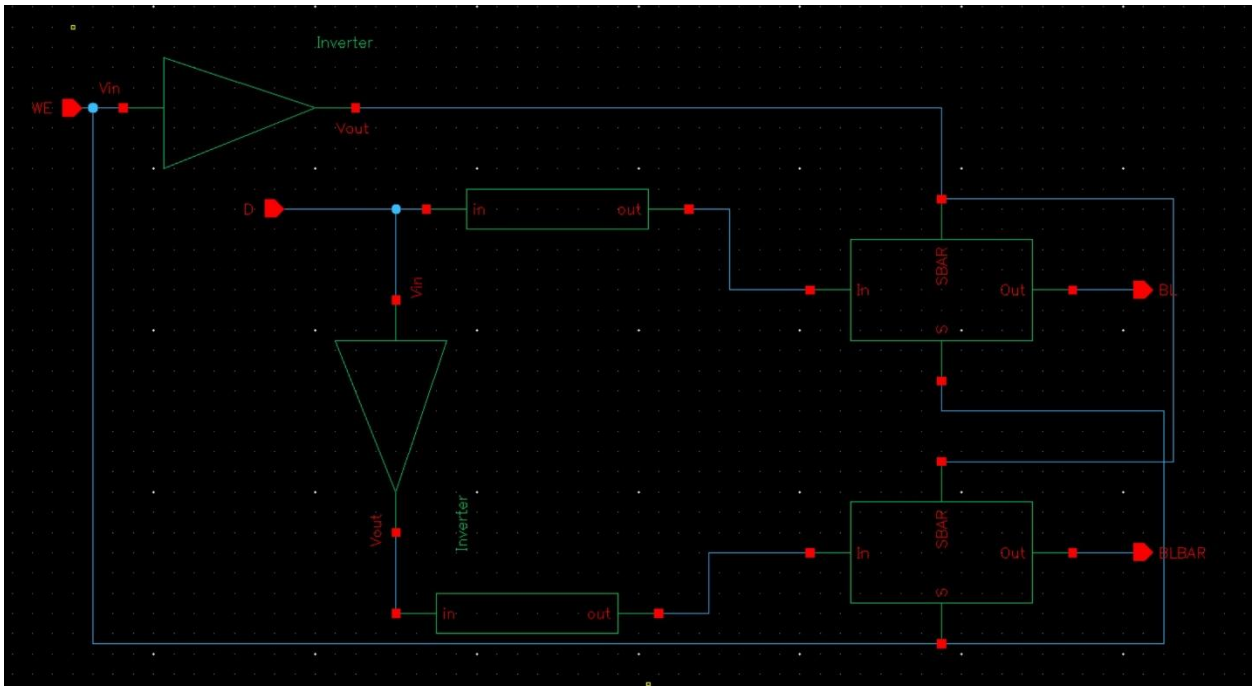


Fig. 4: Schematic of write driver circuit.

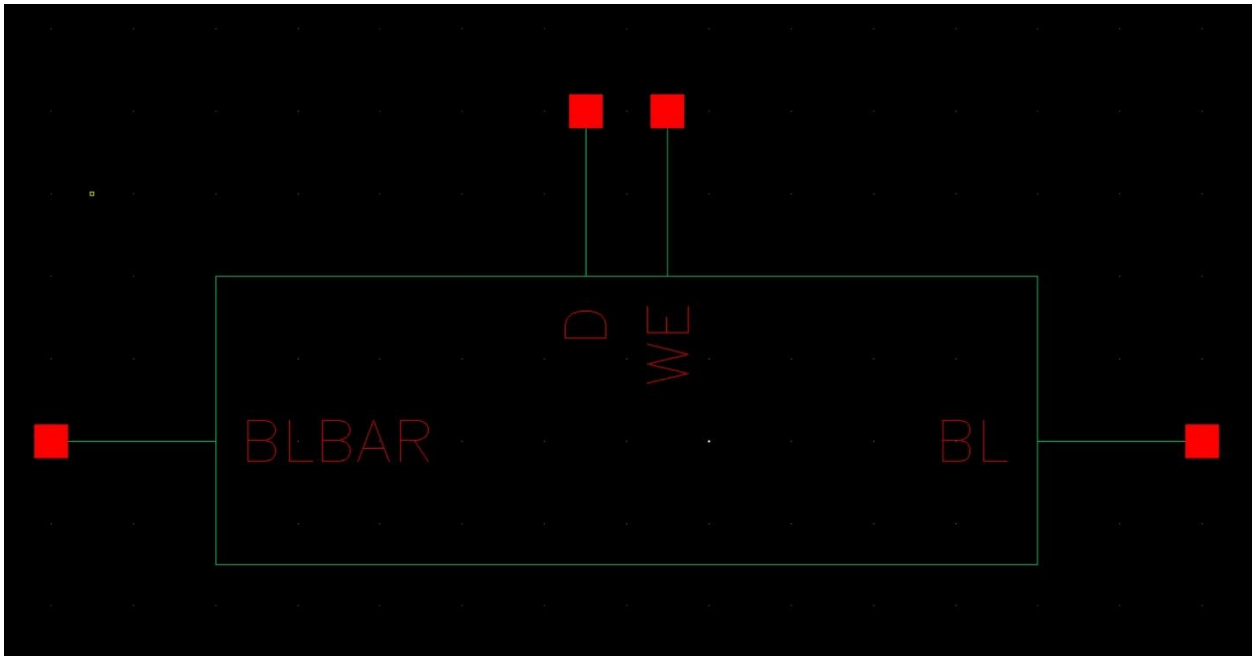


Fig. 4: Schematic of write driver symbol

The write driver circuit is responsible for forcing data into the SRAM cell by discharging one of the bit lines from the precharged state down to ground, thereby crossing the write margin threshold. Activation occurs through the Write Enable (WE) signal, which triggers a full-swing discharge of either BL or BLB. The timing relationship between word line activation and write driver enablement is flexible and does not affect correctness. Structurally, the driver uses two stacked NMOS transistors arranged as pass-transistor (Q1–Q3 and

Q2–Q4). When WE is asserted, the input data processed through inverters activates either Q1 or Q2, resulting in a strong logic “0” on the selected bit line.

Sense Amplifier

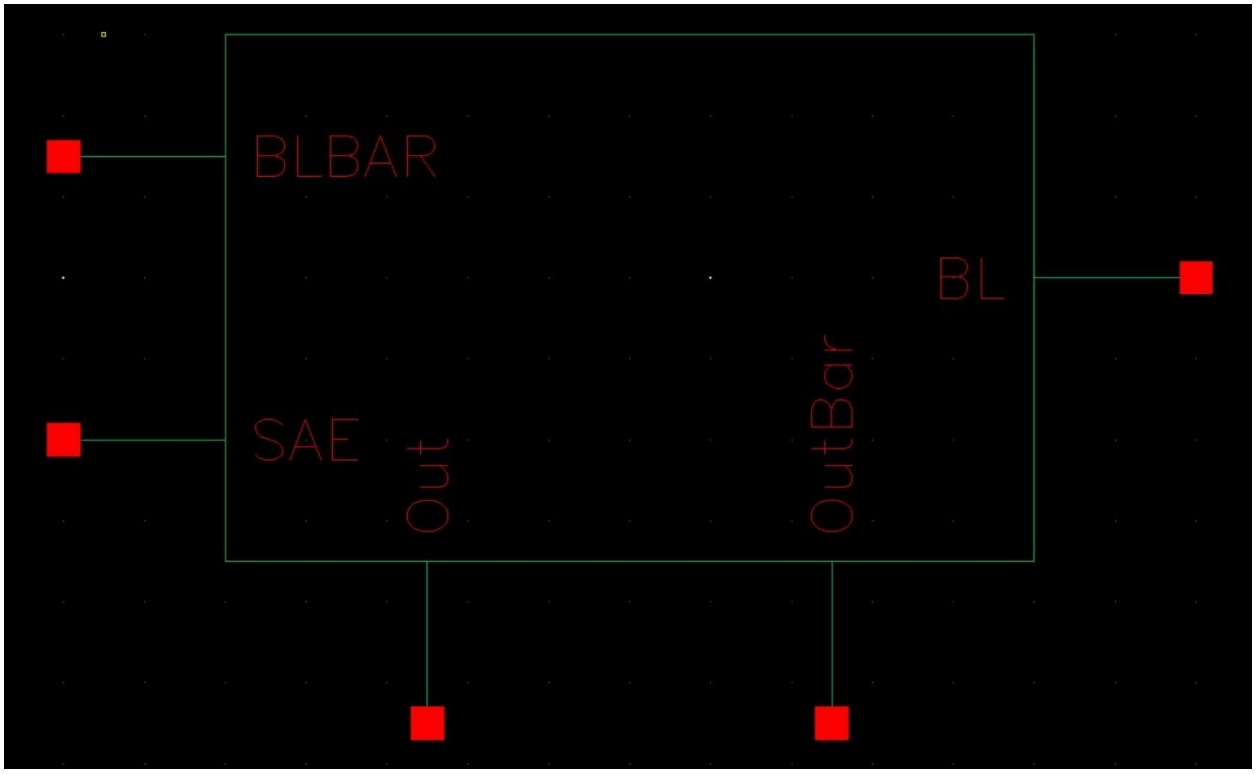


Fig. 4: Schematic of Sense Amplifier Symbol

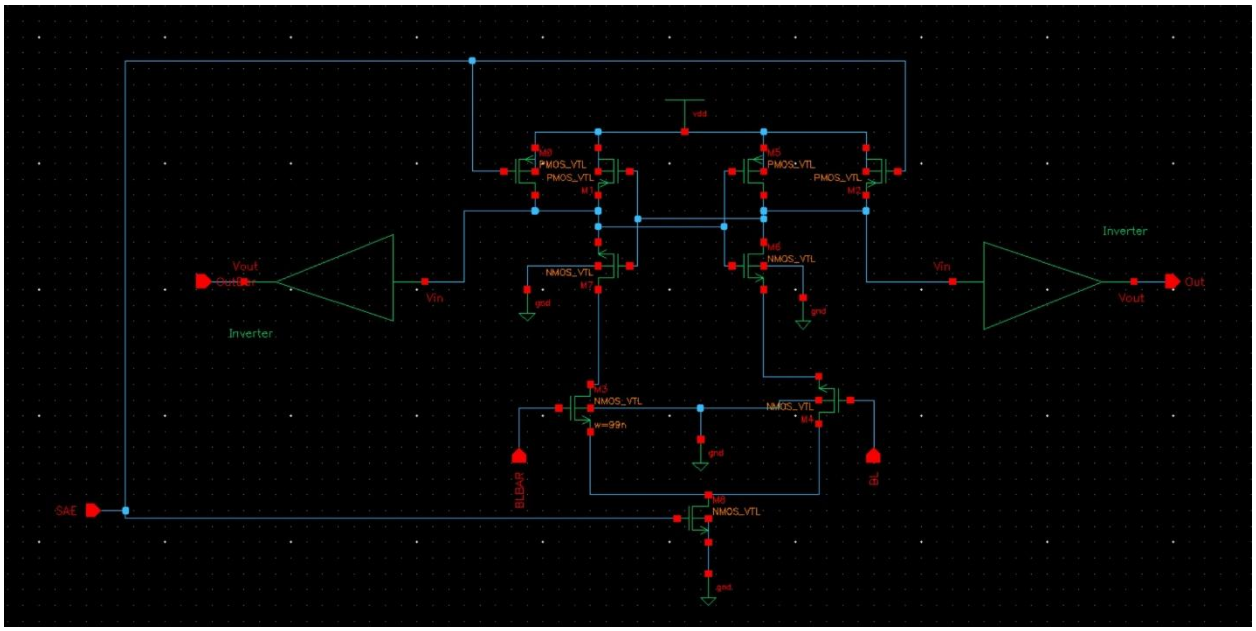


Fig. 4: Schematic of Sense Amplifier

The sense amplifier (SA) is a crucial circuit in SRAM design, as it determines the speed, reliability, and power efficiency of read operations. Its primary role is to detect the small voltage difference that develops between the bit line (BL) and bit bar (BLB) when a cell is accessed and then amplify this differential signal into a full-swing digital output. This ensures fast and nondestructive sensing, which is essential since SRAM does not require refresh cycles after a read.

Decoder

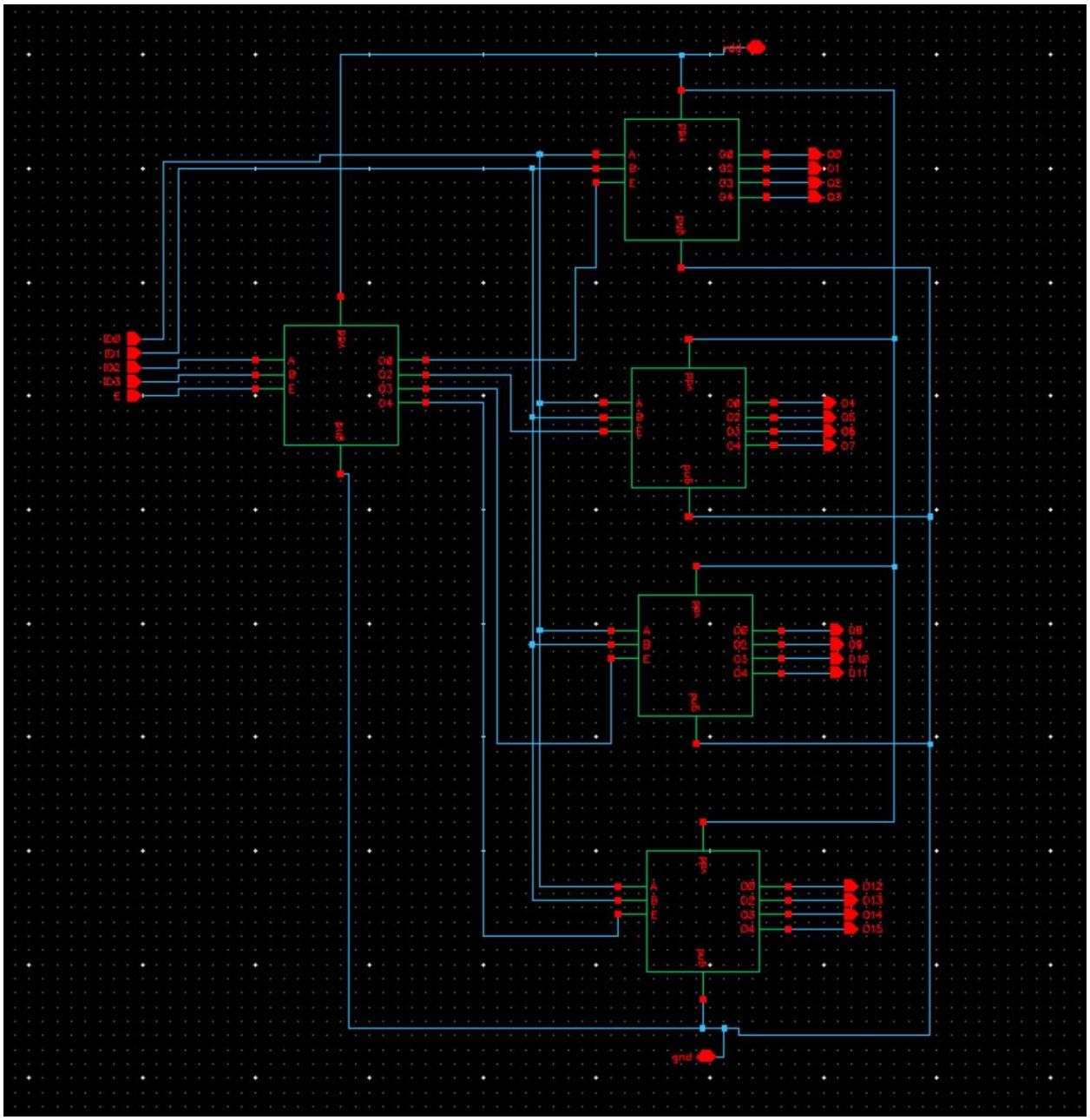


Fig. 7: Schematic of 4:16 decoder

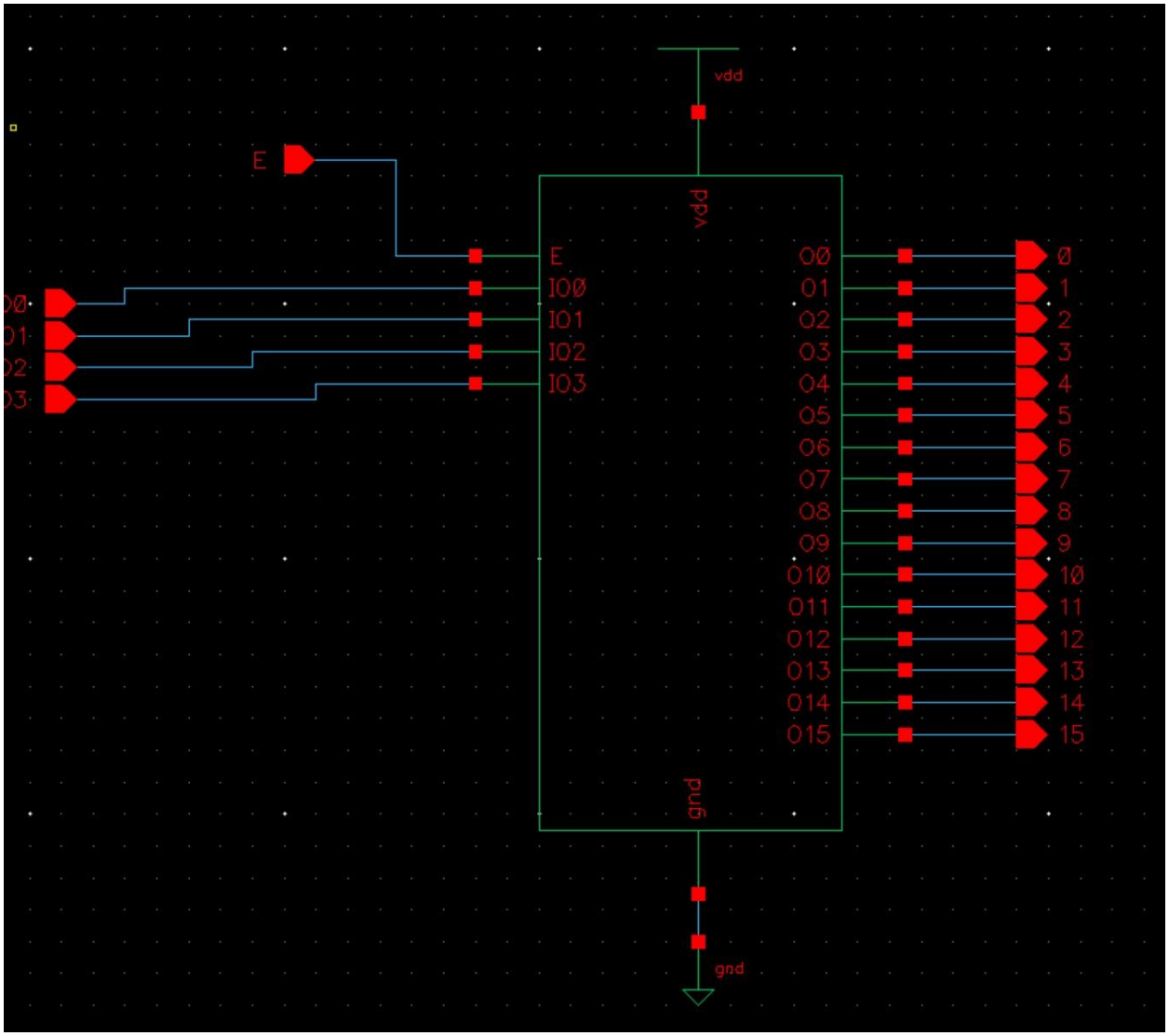


Fig. 7: Schematic of 4:16 decoder symbol

The decoder in the SRAM array is responsible for selecting the appropriate word line based on the input address signals. In this design, the decoder is constructed using NAND gates, which are preferred due to their lower transistor count and better area efficiency compared to pure AND-gate implementations.

For a simple case, a 2-to-4 decoder uses two input lines to generate four distinct outputs, each corresponding to a word line. Extending this concept, a 4-to-16 decoder is employed in the SRAM design, with inputs A, B, C, and D producing 16 outputs (WL1–WL16). Each output drives one word line in the memory array.

For example, when the inputs are A=1, B=0, C=0, and D=0, the ninth word line (WL1) is activated. The waveform confirms that WL1 is asserted, enabling the corresponding row for read and write operations.

By using NAND gates, the decoder achieves reliable word line selection while maintaining compact circuit design, which is critical for dense SRAM arrays.

16×16 Memory Array

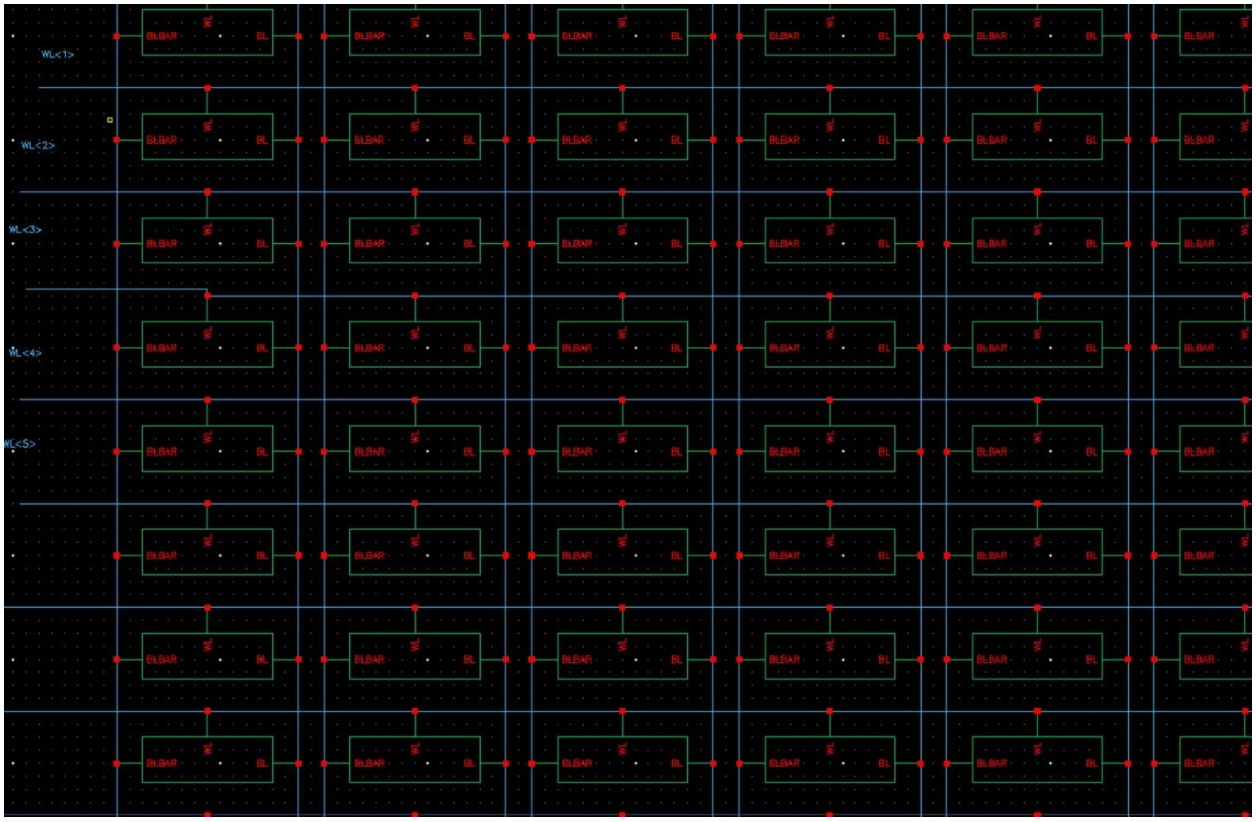


Fig. 7: Schematic of 16X16 memory array.

The designed memory array has a capacity of 256 bits, organized as a 16×16 structure containing 256 individual SRAM cells. Each column in the array is supported by a dedicated sense amplifier, write driver, and precharge circuit, resulting in a total of 16 of each peripheral component.

As the storage capacity scales upward, transistor sizing must be adjusted to account for the loading effect, ensuring stable operation and reliable performance. The array generates 16 readout signals along with their complementary readout-bar signals, which together provide differential outputs for improved robustness.

Row selection is managed by a NAND-based row decoder with four inputs (A, B, C, and D). Depending on the input combination, the decoder activates the corresponding word line, thereby enabling the associated row of cells for read or write operations. This systematic organization allows efficient memory access while maintaining low power consumption and fast response times.

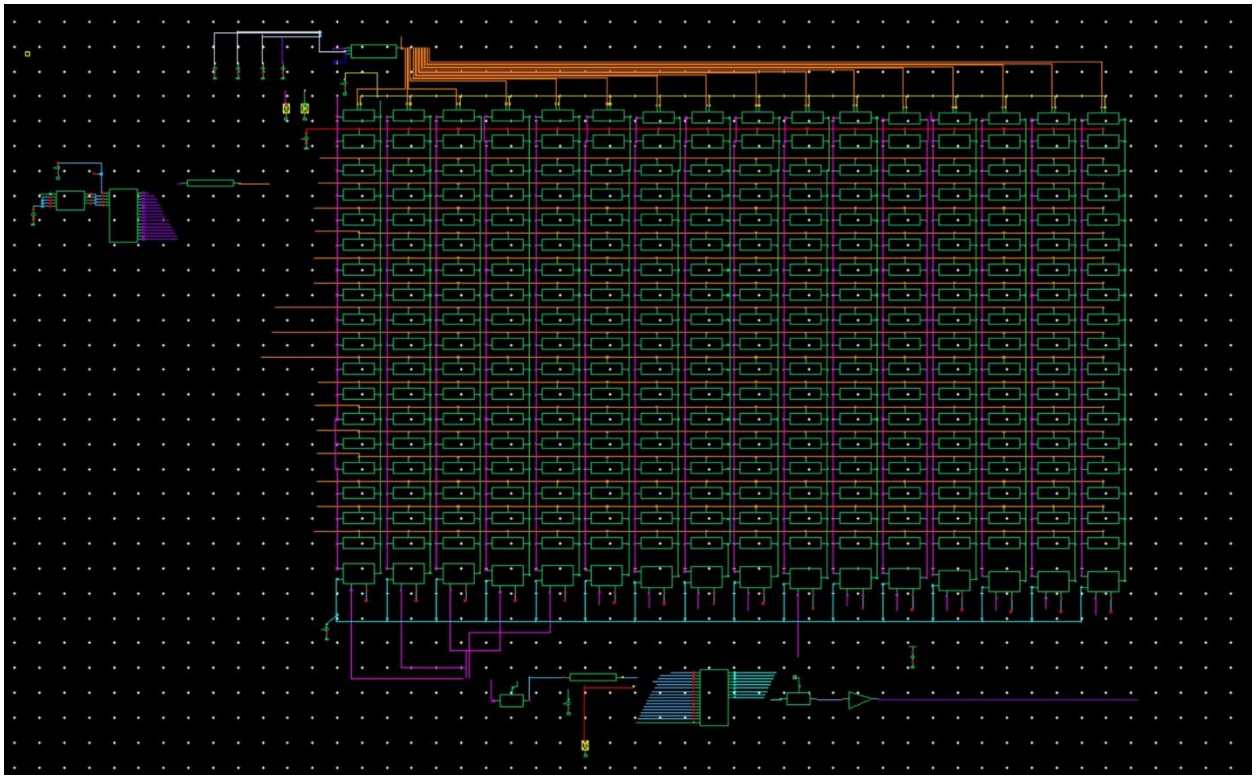


Fig. 7: Schematic of complete system.

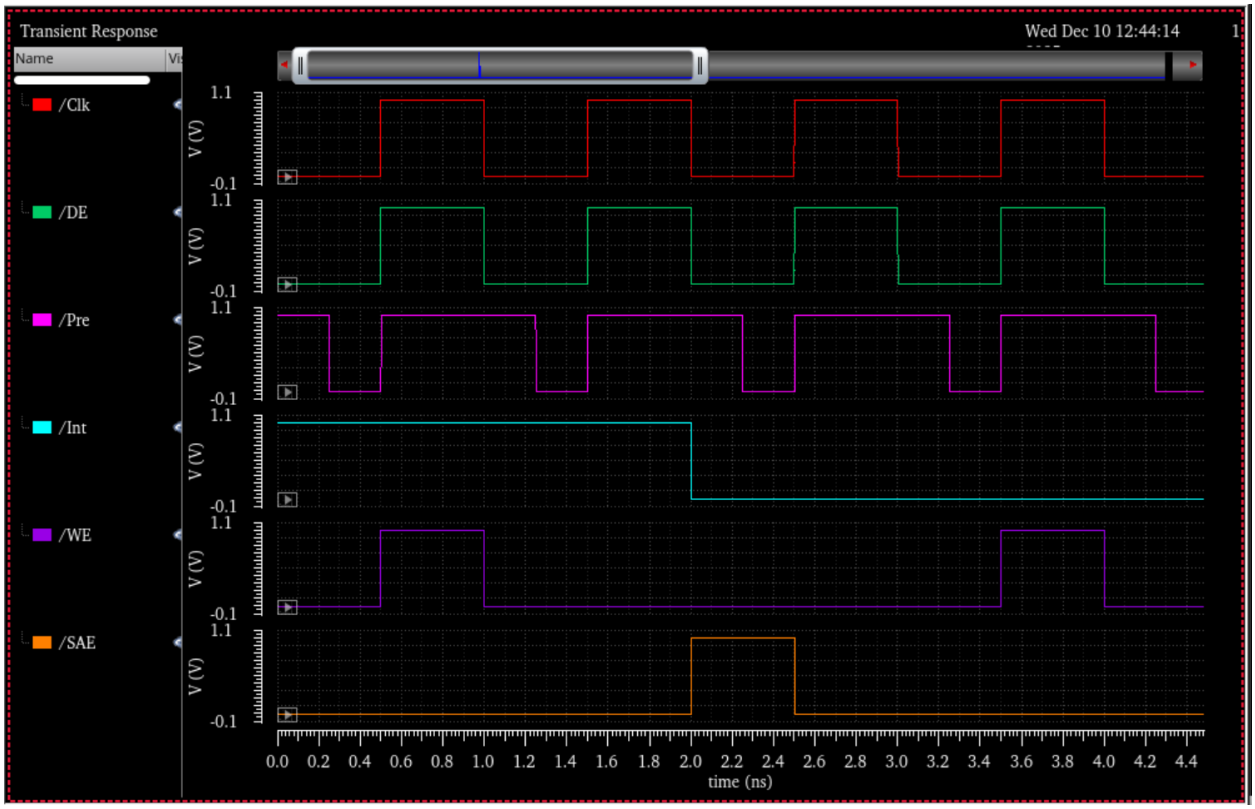


Fig. 7: Control and Data Input Signals