Design and Implementation of SRAM and SDRAM Using Cadence Virtuoso

Asst Prof. Prashant G S
Dept of ECE
JNNCE
Shivamogga, India

Sowmya M
Dept of ECE
JNNCE
Shivamogga, India

Spoorthi R Shelke
Dept of ECE
JNNCE
Shivamogga, India

Vijayalakshmi R
Dept of ECE
JNNCE
Shivamogga, India

Lasya N S
Dept of ECE
JNNCE
Shivamogga, India

Abstract

This paper presents the design and implementation of a 64-bit SRAM and SDRAM memory architecture using Cadence Virtuoso .The main objective is to understand how static and dynamic memories are built at the transistor level and to analyze their performance .The SRAM is designed using a 6T cell ,while the SDRAM includes basic blocks such as decoders ,control logic, and refresh circuitry .All circuits are created and tested through schematic design and simulation. Important parameters like access time, power consumption, proper read/write operation are observed. The results show the both memory designs work correctly and meet the basic requirements for small VLSI applications.

Keywords: SRAM, SDRAM, Cadance, Read-Write operation.

I. INTRODUCTION

Memory units form the core of digital systems, as they hold data that is continuously accessed and updated during operation. With every generation of technology, devices become smaller and faster, creating the need for memory designs that can operate efficiently at lower dimensions. Among the commonly used memory types SRAM offers quick access through a stable cell structure, while SDRAM provides larger storage by using periodic refresh cycles. This project mainly focusses on internal structures of both memory and analysis of basic operations such as read, write and refresh. SRAM uses the conventional 6T cell, SDRAM includes basic blocks such as decoders, control logic and refresh circuitry. These both designs are stimulated using spectre to study their timing behavior, power consumption, and functional accuracy. As the technology scales, understanding the internal operation of these memory structures becomes essential for designing. As mentioned this is transistor level design and verified through simulation to understand the overall performance. This gives us the practical knowledge of memory implementation of VLSI design.

II. Literature Survey

[1] Bhaskar, Akshay – Low-Power SRAM Techniques (2017) This paper discusses how traditional 6T SRAM cells consume more power and highlights different low-power design approaches. The study shows that using MTCMOS technology greatly cuts down the power usage and also improves the operating speed when compared to the basic 6T cell. Another technique, Gated-VDD SRAM, further brings down leakage power and provides faster operation.

[2] Kiran, PN Vamsi & Nikhil Saxena – Comparison of SRAM Topologies (2015) This work compares various SRAM architectures including 6T, 7T, 8T, 9T, and 10T cells. The study highlights that 6T, 7T, and 8T structures are compact because they use fewer transistors, but they face issues like high leakage and poor read stability. The 9T cell improves data stability but cannot reduce leakage effectively. The proposed 10T cell performs better by reducing both leakage power and leakage current, while also enhancing read stability compared to earlier designs.

[3] Gupta, Vasudha & Mohab Anis – Statistical Design of 6T SRAM (2009) This paper introduces a statistical approach for designing 6T SRAM cells to achieve high yield while balancing performance, stability, area, and leakage requirements. The method focuses on selecting transistor dimensions that can tolerate process variations and threshold shifts. It emphasizes moving away from traditional sizing strategies to meet the demands of nanometer technologies. The design also considers real-world variations such as manufacturing inconsistencies, random dopant fluctuations, and important cell parameters like SNM, read current, leakage, and area.

[4] Lakis & Martin Schoeberl – SDRAM Controller for Real-Time Systems (2013) This study explains the importance of worst-case execution time (WCET) analysis in real-time systems. Conventional DRAM controllers mainly aim for high throughput, but their unpredictable delays make them unsuitable for time-critical applications. The paper presents an SDRAM controller specifically built for constant and predictable latency. It has been verified on the Java processor JOP and the T-CREST platform, ensuring that it supports deterministic memory access, which is essential for real-time operations.

[5] Rugma et al. – 64-bit SRAM and SDRAM Controller in SCL 180nm. This paper proposes the design and implementation of 64-bit SRAM and SDRAM controllers developed using the SCL 180nm process technology. The authors use Cadence Genus for synthesis and Cadence Innovus for physical design to meet timing, area, and power targets. The SRAM controller is designed for fast access and strong data reliability, while the SDRAM controller includes features like burst transfer, auto-refresh, and error correction for stable high-speed operation. The work also covers essential physical-design steps such as floorplanning, placement, routing, clock tree synthesis, and post-layout checks.

III. MOTIVATION

The motivation of this work is to design and analyze fundamental SRAM circuits—such as the bitcell, precharge unit, write driver, sense amplifier, and word-line decoder—and to verify read and write operations in Cadence Virtuoso. A 16-bit SRAM architecture is implemented at the schematic level to study cell stability, timing behavior, and memory reliability. In parallel, an SDRAM controller is developed using Verilog and FSM-based design to understand command sequencing, burst operations, and refresh control. This combined approach provides both device-level and system-level insight into modern memory design. The project strengthens overall understanding of VLSI memory subsystems and real-time digital implementation.

IV. Methodology

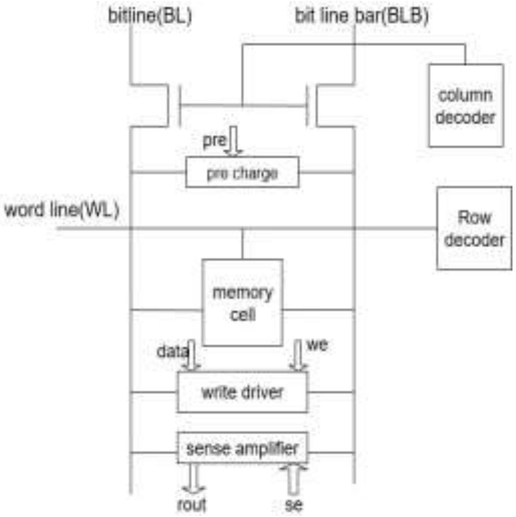

Fig1: Block diagram

The block diagram illustrates the internal data path and control flow of the memory subsystem. The row decoder activates the required word line (WL), while the column decoder selects the corresponding bit-line pair (BL/BLB). Prior to any operation, the precharge circuitry equalizes BL and BLB to a stable reference level to ensure high-speed differential sensing. During a write cycle, the write driver drives the programmed logic value onto the selected bit lines under the control of the WE signal, forcing the state of the 6T memory cell. During a read cycle, the selected cell introduces a differential voltage on BL/BLB, which is detected and amplified by the sense amplifier, enabled through SE. The routed output (r_out) forms the final read data. This architecture supports reliable high-speed data access with controlled power and improved signal integrity.

The two most important parameters which decide the Transistor size of the SRAM cell are:

1. Cell Ratio: The data-operation should not destroy the stored information in the SRAM cell. i.e, Read upset. The cell ratio (beta) should be less than 1.5 for the proper functioning of SRAM cell.

The cell ratio, for the read operation which is given by, **β = (W/L) pull-down / (W/L) access**

2. Pull-Up Ratio: The cell should allow the modification of the stored information during the data-write phase. PR should be small as possible. (meaning access transistor M6 should be strong compared M4).

The pull-up ratio, P.R for the write operation is given by, **P.R = (W/L) access / (W/L) pull-up**

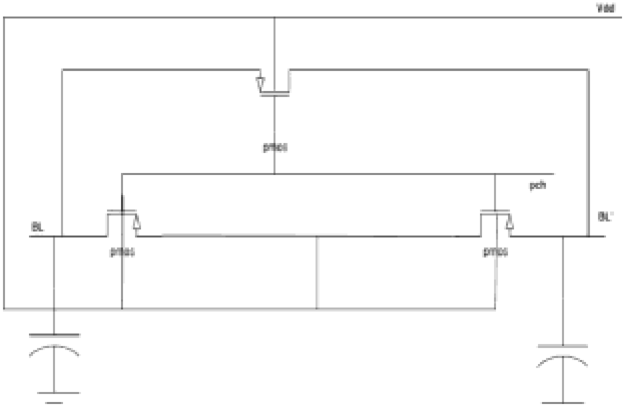

Fig 2: Precharge circuit

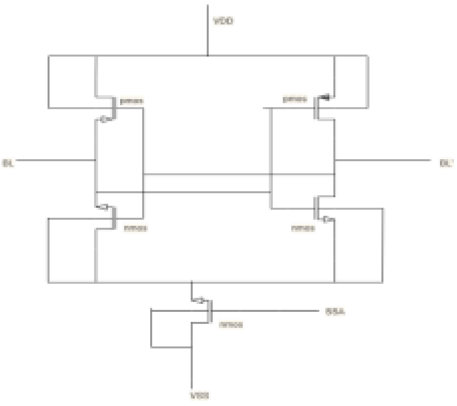

Fig 3: Sense amplifier circuit

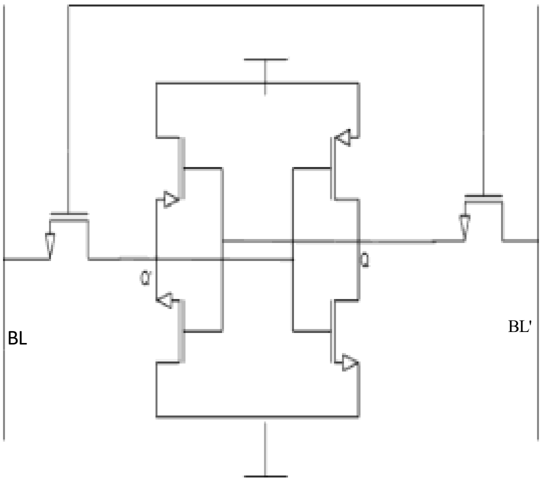

Fig 4 :6T SRAM

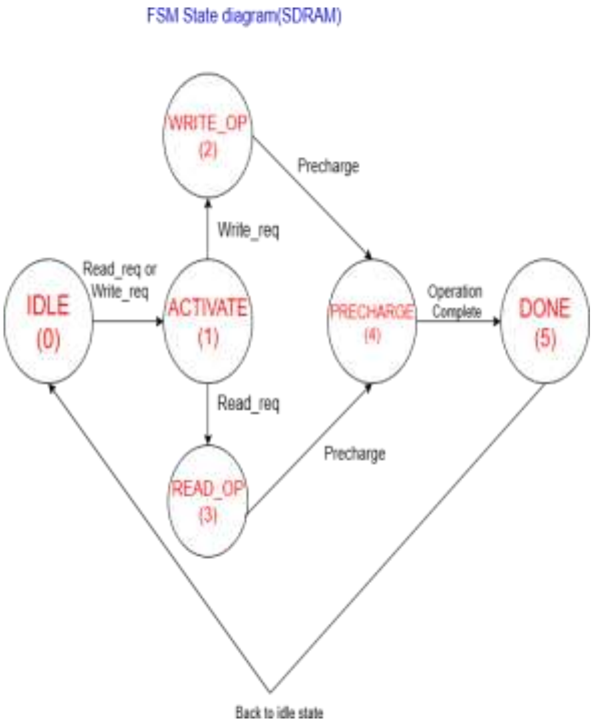

Fig 5: FSM State diagram

The FSM controls the operation of the SDRAM by sequencing through predefined states to ensure correct timing and data integrity. The IDLE state represents the standby condition where no operation is performed. Upon receiving a valid command, the controller enters the ACTIVATE state, which opens the required row by enabling the word line. The READ or WRITE state follows, allowing data to be accessed through the bit lines based on the command issued. After completion of the operation, the FSM transitions to the PRECHARGE state to close the active row and restore the bit lines to a stable condition. The REFRESH state is periodically invoked to retain the stored charge in the memory cells. Finally, the DONE state indicates the successful completion of the memory operation before returning to the IDLE state.

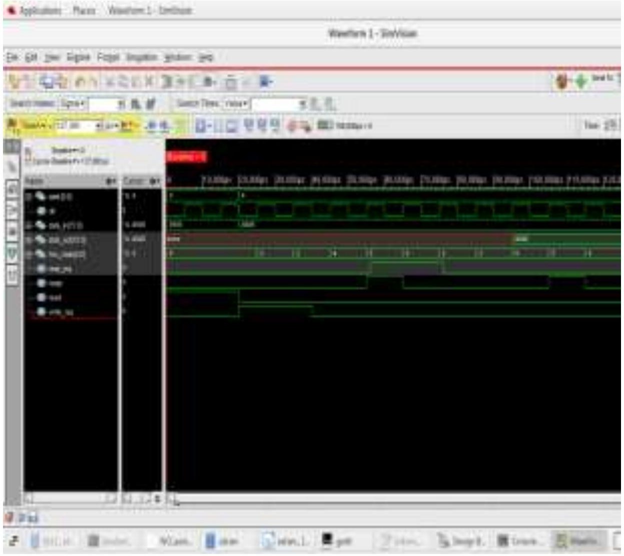

Fig 6: SDRAM Waveform

The simulated SDRAM waveform demonstrates correct synchronous operation with all commands referenced to the rising edge of the clock. When the chip select signal is active, the memory accepts control commands and address inputs. Assertion of the row address strobe activates the selected row by latching the row address and enabling the corresponding word line. After the required row-to-column delay, the column address strobe initiates the read or write operation based on the write enable signal. During write cycles, valid input data is driven on the data bus and sampled correctly, confirming proper write timing. During read cycles, the data bus remains in a high-impedance state until valid data appears after the CAS latency. The read data matches the previously written values, verifying correct storage and retrieval. The wait request signal indicates internal busy periods and prevents overlapping commands. After data transfer, the precharge operation restores the bit lines to their reference levels. The waveform confirms correct command sequencing, timing compliance, and reliable SDRAM functionality.

V. CONCLUSION

The design and analysis of SRAM and SDRAM were successfully carried out using Cadence Virtuoso at the transistor level. The 6T SRAM cell demonstrated reliable read and write operations, confirming correct static memory functionality. The SDRAM design effectively supported dynamic data storage with appropriate control and refresh behavior. Transient simulations verified the functional accuracy of both memory architectures. The results showed that technology parameters significantly influence memory performance characteristics. Transistor sizing was observed to impact access delay, signal integrity, and overall memory stability. Bit-line behavior during read and write operations highlighted the importance of load capacitance. The SDRAM control circuitry showed proper synchronization with the system clock and correct command sequencing. Overall, the project strengthens practical understanding of memory architecture implementation in VLSI design.

VI. REFERENCES

1. **Jawar Singh, Saraju P Mohanty, Dhiraj K Pradhan** “Robust SRAM Designs and Analysis”, 1-29, 2012

2. **Mulani Mumtaj Raju, Pawar Swapnali, and Patole**, “Smart Waste Management System Using ESP32,” International Research Journal Innovation in Engineering and Technology, 2022.

3. **Priyanka Thirumugam and Damninda Hiremath**, “Smart Dustbin and Garbage Monitoring System,” IEEE, 2023.

4. **Joni Welman Simaptaung and Vincent**, “IoT-Based Smart Parking System Management Using ESP32,” IEEE, 2022.

5. **Md. Hasanul Kabir, A.A. Nayeem, A.A. Akash, R.H. Hasan, and A. Rahman**, “Design and Implementation of IoT-Based Intelligent Management Smart Street Light System,” IEEE, 2023.

6. **Smitaraj. R. Shastri, Pallavi B. Chitale, and Gaikwad,** “IoT-Based Smart City.”

7. **AS Rugma, G Sreelatha, VR Adersh** 2024 International Conference on Smart Electronics and Communication Systems (ISENSE), 1-6, 2024

8. Verification of SDRAM controller using system verilog **Venkatesh Vutukuri, Vijaya Bhaskar Adusumilli, Pavan Kumar Uppu, Swasthik Varsa, Ravi Kumar Thummala** 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 1-6, 2020

9. FSM-based digital design using Verilog HDL **Peter D Minns, Ian Elliott John Wiley & Sons**, 2008

10. Finite state machine design and VHDL coding techniques **Iuliana Chiuchisan, Alin Dan Potorac, Adrian Graur** Development and Application Systems, 75, 2010