VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnanasangama, Macche, Santibastwada Road
Belagavi-590018, Karnataka

A
UG PROJECT REPORT
on

Generation and analysis of 6T SRAM memory instance - bank1, cm2, HDSP

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics & Communications Engineering - ECE
by

1DS19EC087	**Naresh H.**
1DS19EC103	**Prateek Chitnis**
1DS19EC116	**Sachin Bhat G.**
1DS20EC414	**Karthik S.**

Under the guidance
of
Dr. Ninu Rachel Philip

Assistant Professor, ECE Dept., DSCE, Bengaluru

Department of Electronics & Communication Engineering
(An Autonomous College affiliated to VTU Belgaum, accredited by NBA & NAAC, Ranked by NIRF)
Shavige Malleshwara Hills, Kumaraswamy Layout,
Bengaluru-560078, Karnataka, India
2022-23

Certificate

Certified that the project work entitled "Generation and analysis of 6T SRAM memory instance - bk1, cm2, HDSP" carried out by **Naresh H.** (1DS19EC087), **Prateek Chitnis** (1DS19EC103), **Sachin Bhat G.** (1DS19EC116), **Karthik S.** (1DS20EC414) are bonafide students of the ECE Dept. of Dayananda Sagar College of Engineering, Bangalore, Karnataka, India in partial fulfillment for the award of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2022-23. It is certified that all corrections / suggestions indicated for project work have been incorporated in the report deposited to the ECE department, the college central library & to the university. This final year project report **(Course Code : 19EC8ICPR2)** Phase-II has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

Dept. Project Coordinators (Section incharges)	Project Guide
Abhishek - Suma / Manasa / Srividya / Bindu	Dr. Ninu Rachel Philip

Head of the Department	Dr. B.G. Prasad
Dr. T.C.Manjunath, Ph.D. (IIT Bombay)	Principal, DSCE

External Project Viva-Voce

Name of the project examiners (int & ext) with date :

1 : Signature : _____

2 : Signature : _____

Declaration

Certified that the project work entitled, "Generation and analysis of 6T SRAM memory instance - bank1, cm2, HDSP" with the project work course code **19EC8ICPR2** is a bonafide work that was carried out by ourselves in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics & Communication Engg. of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2022-23. We, the students of the project group/batch no. R-42 do hereby declare that the entire project work has been done on our own & we have not copied or duplicated any other's work or may be the extension of the works done by the earlier students. The results embedded in this UG project report have not been submitted elsewhere for the award of any type of undergraduate degree.

Naresh H.
1DS19EC087

Sign : _____

Prateek Chitnis
1DS19EC103

Sign : _____

Sachin Bhat G.
1DS19EC116

Sign : _____

Karthik S.
1DS20EC414

Sign : _____

Date : 21/ 06 /2023

Place : Bengaluru - 78

Acknowledgement

The elation and contentment that accompany the triumphant culmination of any undertaking would be rendered incomplete without extending heartfelt gratitude to those instrumental in its realization. We take immense pride in our affiliation with Dayananda Sagar College of Engineering, an esteemed institution that has ardently nurtured our growth and fortified our pursuits in all facets of our journey.

We extend our heartfelt appreciation to the distinguished individuals who have played integral roles in our academic journey. Our gratitude goes to Dr. Hemachandra Sagar, Chairman; Dr. Premachandra Sagar, Vice Chairman; Galiswamy, Secretary and Tintisha Sagar, Joint Secretary.

We would also like to express our sincere thanks to Dr. B. G. Prasad, our esteemed Principal, and Dr. T. C. Manjunath, Professor and Head of the Department of Electronics and Communication Engineering (ECE), for their invaluable guidance and unwavering encouragement throughout the duration of our program.

Furthermore, we wish to convey our profound gratitude to the coordinators who have offered valuable suggestions and guidance during the course of our project. Our deepest appreciation goes to Dr. Ninu Rachel Philip, Assistant Professor, who served as our mentor, and our previous guide Prof M.N. Pradeep for providing indispensable guidance, continuous support, and unwavering motivation, leading to the successful completion of our project.

Lastly, we seize this opportunity to extend our earnest gratitude and utmost respect to our parents, the dedicated teaching and non-teaching staff of the department, the library staff, and all our friends who have provided direct or indirect support throughout our academic endeavors.

Table of Contents

List of Figures

Nomenclature and Acronyms

Abbreviations:

ADE	Analog Design Environment
ASIC	Application Specific Integrated Circuits
BL	Bit Line
CIW	Command Interface Window
DRAM	Dynamic Random Access Memory
EDA	Electronic Design Automation
FinFET	Fin Field Effect Transistor
GUI	Graphical User Interface
IoT	Internet of Things
MOSFET	Metal Oxide Field Effect Transistor
NMOS	N-Channel Metal Oxide Semiconductor
PMOS	P-Channel Metal Oxide Semiconductor
RAM	Random Access Memory
ROM	Read Only Memory
SoC	System On Chip
SRAM	Static Random Access Memory
VLSI	Very Large Scale Integration

Abstract

In the current generation, memory plays a pivotal role as the fundamental building block of microprocessors. Microprocessors employ two primary types of memory cells: Read-Only Memory (ROM) and Random Access Memory (RAM). RAMs can be further categorized as Static RAM (SRAM) and Dynamic RAM (DRAM). SRAMs are extensively utilized in cache memory and application-specific memories. When aiming to enhance microprocessor performance, designers consider various parameters, with a significant focus on designing an area-optimized cell. Random Access Memory serves as a form of computer data storage, responsible for the retention of data. It comprises peripheral circuits that establish connections between data lines and the targeted storage, enabling the reading and writing of data entries. The objective of this project is to design a single-port SRAM, encompassing the implementation of diverse SRAM bit cells and their corresponding peripheral circuits. Subsequently, a memory compiler capable of generating memory of any desired size will be created. The circuits were implemented utilizing the Cadence EDA tool, while simulations were conducted using the Virtuoso ADEL platform.

Keywords : *Generation, Memory, Microprocessor, Read-Only Memory,Random Access Memory, SRAM, DRAM, Cache Memory, Aera-optimized cell, Memory Compiler.*

Chapter-1

Introduction

Over the course of recent decades, the prominence of Complementary Metal Oxide Semiconductor (CMOS) and Metal Oxide Semiconductor Field Effect Transistor (MOSFET) technology has surged, establishing it as the dominant fabrication method and the exclusive choice for semiconductor memories. Of particular significance are Static Random-Access Memories (SRAMs), which hold a pivotal role in the memory hierarchy of modern computer systems and remain critical components across a broad range of microelectronics applications, encompassing consumer wireless devices, high-performance server processors, multimedia systems, and System-on-Chip (SoC) products.

Moore's Law formulated by Late Gordon Moore, stipulates that the number of transistors per generation on an integrated circuit approximately doubles every two years. This exponential growth has propelled VLSI designs to achieve a performance increase of five orders of magnitude over the past four decades. As a result, Moore's Law has become the fundamental guiding principle for the semiconductor industry, driving the future scaling down of process technologies.

More than half of the transistors in present-day high-performance microprocessors are dedicated to cache memories, with this ratio expected to grow in the foreseeable future. Static Random-Access Memory (SRAM) is typically the preferred choice for embedded memories due to its resilience in noisy chip environments. Consequently, significant attention has been directed towards the design of low-power, high-performance SRAMs, as they serve as critical components in both handheld devices and high-performance processors.

Semiconductor memory exhibits notably faster access times compared to other types of data storage, with the ability to read from or write to a byte of data within a few nanoseconds, whereas rotating storage mediums such as hard disks necessitate access times in the range of milliseconds. Hence, semiconductor memory is extensively employed as the primary storage for holding the data that a computer is currently processing. However, it is important to note that shift registers, processor registers, data buffers, and similar small digital registers lacking memory address decoding mechanisms are not considered memory, despite their ability to store digital data. The volatile nature of both SRAM and Dynamic RAM (DRAM) should also be

acknowledged, with large SRAM arrays commonly employed as cache memory in microprocessors and application-specific integrated circuits (ASICs), consuming substantial chip area.

Designers encounter a considerable challenge when endeavoring to strike an optimal performance-to-cost ratio for such processors. While large arrays of fast SRAMs contribute to system performance improvements, the cost of incorporating these arrays into a chip is directly proportional to the impact on chip area. Hence, the pursuit of minimizing the footprint of SRAM cells arises from the need to strike a balance between these conflicting requirements. Consequently, memory circuitry on a chip typically comprises densely packed millions of cells, rendering SRAM arrays the most densely packed circuitry on a chip. Notably, memory chips constitute approximately one-third of the entire semiconductor market, encompassing DRAM, SRAM, ROM, and flash memory.

The application of these memory cells extends to electronic devices such as digital cameras, cell phones, and electronic interface designs, owing to their power efficiency and bandwidth considerations. Notably, SRAM finds relevance across various sectors, including computing, communication, consumer electronics, automotive, and industrial domains. From Intel's initial 1Kb DRAM chip to the recent inclusion of SRAM-based caches occupying over 90% of the 1.78 billion transistors in Intel's Montecito processor, memory has served as the driving force behind the remarkable advancement in CMOS and MOSFET technology witnessed over the past few decades.

In Very Large-Scale Integration (VLSI) systems, microprocessors, and Static Random Access Memory (SRAM) designs, considerations regarding low power consumption and high throughput have gained increasing prominence. In modern chip designs, a significant portion of System-on-Chips (SoCs) for Internet of Things (IoT) applications incorporates substantial caches built with SRAM cells, resulting in larger chip areas and subsequent power consumption issues. Consequently, contemporary chip designers strive to achieve high power efficiency without compromising chip speed and stability. Traditional MOSFETs face limitations, primarily with regard to gate leakage current reduction, which hinders further reduction in specific point size. As a means to overcome these limitations, Fin Field Effect Transistors (FinFETs) have emerged as a prevalent choice in many state-of-the-art SRAM devices, supplementing or even replacing traditional MOSFETs. Notwithstanding these advancements, the prevailing SRAM structure predominantly relies on the traditional 6T structure.

1.1 Overview of the project work

Using 18nm FinFET technology involves the fabrication and implementation of Fin Field Effect Transistors (FinFETs) with a minimum feature size of 18 nanometers. FinFETs are a type of transistor design that offers improved performance and power efficiency compared to traditional MOSFETs.

The utilization of 18nm FinFET technology allows for greater transistor density, enabling more transistors to be packed onto a single chip. This increased transistor density results in improved performance capabilities, such as faster switching speeds and higher operating frequencies. Furthermore, FinFET technology offers enhanced power efficiency by reducing leakage current, which is a major concern in semiconductor devices. The three-dimensional fin-like structure of FinFETs helps to control the flow of current, resulting in reduced power consumption and improved energy efficiency.

However, it is important to note that transitioning to smaller process nodes, such as 18nm, poses certain challenges. Manufacturing at such scales requires advanced fabrication techniques and precise process control to ensure high yields and reliability. Additionally, the design complexity increases, requiring careful consideration of power distribution, signal integrity, and thermal management.

Overall, utilizing 18nm FinFET technology offers significant advantages in terms of improved performance, power efficiency, and integration capabilities, paving the way for the development of more advanced and efficient electronic devices.

1.2 Background information about the project work

The project aims to generate and analyze a specific instance of the 6T SRAM memory. The 6T SRAM cell is a common design for SRAM, consisting of six transistors that form a flip-flop circuit capable of storing a single bit of data. By analyzing this instance of the memory, the project likely seeks to evaluate its performance, power consumption, reliability, or other relevant characteristics.

1.3 Motivation obtained to take up the project work

The contemporary world is teeming with electronic devices housing intricate chips or integrated circuits. Remarkable advancements in sectors like automobiles, Aerospace, computer science, and medical machinery owe their success to the remarkable strides made in the field of Very Large Scale Integration (VLSI) over the past few decades. Present-day CPU designs often integrate Static Random Access Memory (SRAM) to achieve accelerated memory access. Streamlining the SRAM block design process through automation offers significant benefits, expediting overall design procedures. Leveraging state-of-the-art FinFET technology, our objective is to develop a compiler capable of automating the creation of SRAM arrays, complete with the essential peripheral circuits tailored to the designer's specifications. By doing so, the need for manual design is eliminated, ensuring optimal size and functionality for the desired application.

1.4 Problem statement of the project work

The conventional MOS doesn't provide a good control on the channel. If a good channel control can be achieved then lower nodes can be used which helps in reducing area and power which in turn helps in cost cutting. This problem is addressed in our project wherein we are using FinFET technology to achieve this. The design specifications that we have followed allows our design to achieve the above.

1.5 Objectives of the project work

The project work centers around accomplishing the following key objectives:

1. Developing a single port SRAM memory compiler using 18nm FinFET technology. This involves creating a compiler tool that automates the design process of SRAM memory utilizing the advanced capabilities of 18nm FinFET technology.
2. Designing High Density (HD) Bitcell for SRAM.
3. Designing essential peripheral circuits for the Bitcells, such as address decoders, pre-charge circuits, sense amplifiers, and write drivers. These circuits are crucial for the proper functioning of the SRAM memory.
4. Studying the performance of High density bitcell of 6T SRAM to assess their suitability for specific applications.

5. Implementing a memory array using SKILL programming. This step involves utilizing the programming language SKILL to construct a memory array that incorporates the designed Bitcells and peripheral circuits.

1.6 Organization of the project report

The project work undertaken by us is organized in the sequence as follows.

Chapter 1: Introduction

- This chapter provides an overview of the Semiconductor Memory Industry and its importance.
- It outlines the main objectives of the project and explains the methodology followed throughout its completion.

Chapter 2: Literature Survey

- This chapter presents a literature review of different SRAM configurations, discussing their design choices, advantages, and disadvantages.

Chapter 3: System Block Diagram

- This chapter describes the architecture of SRAM and its functional blocks.
- It includes block diagrams, flow charts of our project work.

Chapter 4: Implementation

- This chapter covers the schematic implementation and layout design of the blocks using Cadence Virtuoso tool.
- It also provides an introduction to Memory compiler using SKILL programming.

Chapter 5: Results

- This chapter presents simulation and experimental results of our project work.

Chapter 6: Conclusion and Future Scope

- This section summarizes the conclusions drawn from the project and discusses its future possibilities and areas of further exploration.

Chapter 2

Literature Survey

The increasing demand for higher embedded-SRAM densities in SOC applications requires the development of smaller yet high-performance 6T-SRAM cells using standard CMOS technology [1]. However, achieving aggressive cell sizes through alternative architectures and processing techniques can lead to added complexity, manufacturing difficulties, and a decrease in cell speed and standby current. [3] Nevertheless, SRAM cells produced using standard CMOS logic processes can achieve high-density by implementing inter-well isolation and optical proximity correction (OPC) in the cell layout. [5] The utilization of 248 nm lithography and optimized OPC methodology plays a crucial role in achieving smaller cell sizes. [8]

To tackle the challenges related to data stability, leakage power consumption, and memory integration density, a novel single-ended read SRAM cell has been designed using underlap engineered FinFETs [9]. This design aims to enhance voltage margins during read and write operations, reduce leakage power consumption, and maintain memory integration density and read data access speed [10]. Additionally, a single-ended read asymmetrical SRAM cell based on FinFETs has been proposed, offering stronger data stability, improved write ability, and minimized leakage power consumption [11].

Regarding technology scaling, CMOS systems encounter an increase in leakage current and power consumption[7]. To mitigate these effects, a study investigates the performance of a 6T SRAM cell using CMOS technology, demonstrating improved static noise margin (SNM) curves, reduced power consumption, and a smaller footprint compared to an 8T SRAM cell[12]. FinFET devices are also considered due to their advantages over traditional bulk silicon MOSFETs, including reduced sub-threshold leakage and gate leakage currents, as well as enhanced carrier mobility. A design for a 6T SRAM memory cell using the FinFET process, with PMOS transistors serving as pass-gate (PG) transistors instead of NMOS, exhibits benefits in terms of area, speed, power consumption, and read stability[13].

High-performance SRAMs face challenges arising from the large number of transistors used in cells, the utilization of the smallest transistors, leakage issues, and variations in sub-100 nm CMOS technologies[14]. Process variations and VT mismatch present limitations on cell size, while NBTI and PBTI impact PMOS and NMOS devices. Read

stability and disturbance problems emerge due to conflicting read/write requirements and voltage variations. Techniques such as optimizing the strengths of the pull-down NMOS and access pass-transistor NMOS, controlling word-line pulse width, and temporarily reducing bit-line voltage can improve cell stability and performance.

Technology scaling introduces concerns such as signal loss, degradation of noise margin, and long-term reliability issues. The article explores design approaches and SRAM circuit techniques that can tolerate leakage, variation, and degradation in conventional planar CMOS technology, aiming to alleviate these constraints.

Chapter 3

Project Details

3.1 Methodology of our project

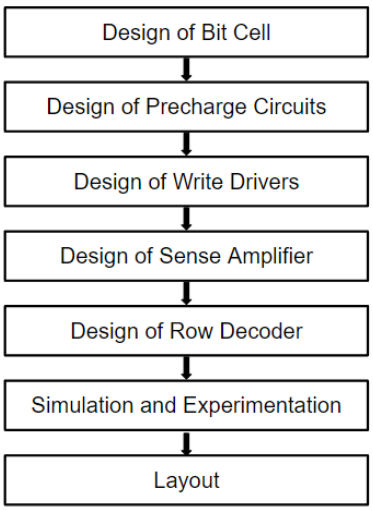


Fig.3.1 : Methodology

1. **Design of bitcell**: High-density 6T SRAM has been designed based on application requirements, while ensuring that the transistors are sized to meet the cell ratio constraints.
2. **Design of Precharge Circuits** Each column in the bitcell array is associated with a bit-line pair (BL and its complement). These bit-line pairs are connected to a precharge circuit, which equalizes and pulls up the selected column's bit lines to the VDD level before read or write operations. Since the bit-lines are charged to VDD regularly, PMOS transistors, which pass strong logic high signals, are utilized in the design of the precharge circuit.
3. **Design of Write Drivers** Write drivers are responsible for controlling the bit-line pair (BL and its complement) during write operations in an SRAM array. With high word-line and bit-line capacitance in large arrays like a 128x128 bitcell array, it is inefficient to increase the size of transistors in the main circuits to handle the capacitance. Instead, dedicated driver circuits are designed to efficiently drive the large capacitance and pull down the appropriate bit-line based on the input data, ensuring proper write operation.
4. **Design of Sense Amplifiers** Sense amplifiers (SA) play a crucial role in CMOS Static Random-Access Memories (SRAMs) as they amplify small differential

voltages on bit-lines during read access. We employed a latch-based sense amplifier, specifically the Strong-arm latch, which offers the advantage of low static power dissipation. Various types of sense amplifiers, including differential amplifiers, are available for this purpose.

5. **Design of Row Decoder** SRAMs use row decoders and column decoders to select specific rows and bit-line pairs in the array based on address bits. The row decoder has a pre-decoding and post-decoding stage, designed to simplify automation and consider wordline capacitance. The column decoder utilizes a 2:1 MUX with pass-transistor logic.
6. **Simulation and Experimentation**

   We experiment with various parameters of High Density Bitcell:

   - Read Current (*Iread*)
   - Leak Current (*Ileak*)
   - Write Margin (*WM*)
7. **Layout** Generate LVS and DRC clean Layout for High Density Bitcell and it's peripheral circuits.

3.2 SRAM Memory Chip

SRAMs have become a ubiquitous element incorporated in System-on-Chip (SoC), Application-Specific Integrated Circuit (ASIC), and microprocessor designs. Due to their extensive usage, diverse demands arise in terms of circuit design and memory configuration.

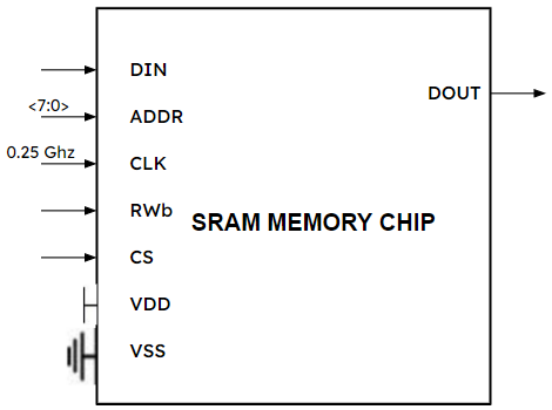


Fig. 3.2 : SRAM Memory chip

Fig 3.2 shows the chip level configuration of SRAM memory.

CLK: Global clock is given to the chip for the start of the operation

DIN: To send in data to the array.

DOUT: To read data from the array.

ADDR: Address for selection of worldline.

VDD: Power supply.

VSS: Ground connection.

CS: Chip select.

RWb: Read enable or write enable bar.

3.3 SRAM Memory Architecture

The architecture of SRAM as shown in Fig 3.3. memory comprises different functional components, including an array of bi-stable memory bitcells, along with peripheral circuits like address decoders (row and column), sense amplifiers, write drivers, and bit-line precharge circuits. The control block unit transmits necessary signals to the memory array and other peripheral circuits to facilitate read and write operations on the array.

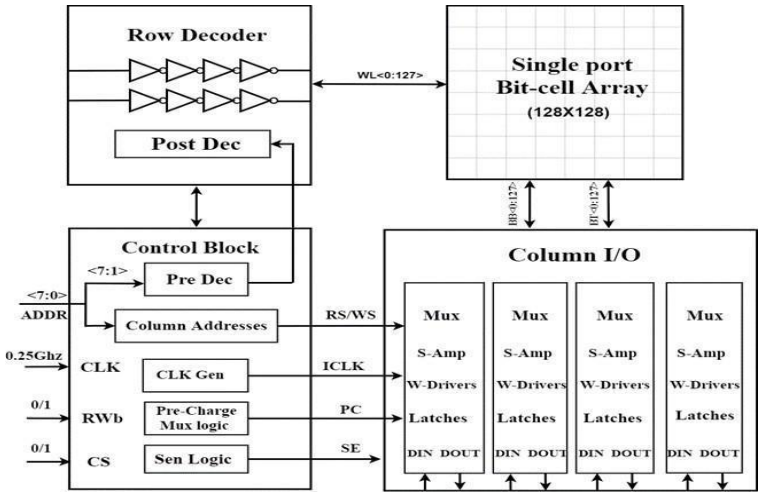


Fig. 3.3 : SRAM Memory Architecture

3.4 Single port Bitcell Array

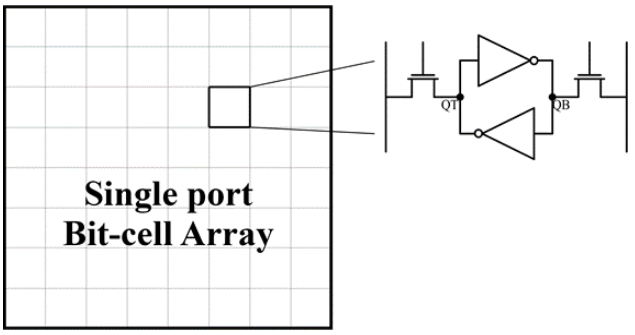


Fig.3.4 : Single port Bitcell Array

An SRAM array consists of numerous identical bitcells, each of which is made up of cross-coupled CMOS inverters. These inverters form a bistable latch, allowing the bitcell to retain data even when power is supplied. The bistability refers to the latch's ability to hold two states: one side high and the other low, or vice versa. In the depicted Fig 3.4, the data is stored in nodes QT and QB. A memory bitcell serves the purpose of storing a single bit of information, either "1" or "0". The bitcells in an SRAM array share a common word-line (WL) in each row and bit-line pairs (BL and complement of BL) in each column. The dimensions of the SRAM array are constrained by electrical characteristics such as the capacitances and resistances of the bit-lines and word-lines, ensuring uniform access delay throughout the array.

3.5 Column I/O

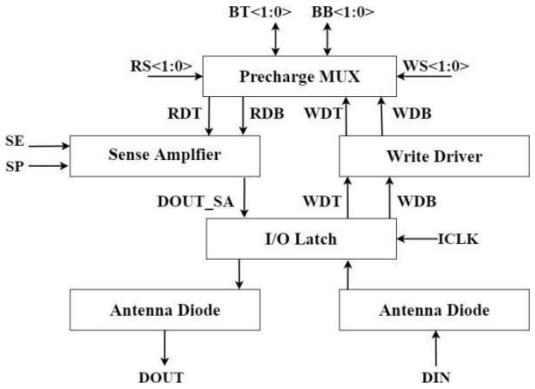


Fig.3.5 : Column I/O

1. **PreCharge MUX:** Its purpose is to initialize the operation of bit cells by pre-charging the pairs of bit lines, BT and BB. During reading, the voltage difference at BT ¡1:0 and BB1:0 is considered as input, while during writing, data is written by creating a voltage difference in the bit-line pairs. RS1:0 and WS1:0

are signals from the control block that indicate the selection of read and write operations, respectively. This ensures separate paths for reading and writing.

2. **Sense Amplifier:** Its function is to detect the data being read by sensing the differential voltages present at the bit-line pairs. When reading, RDT and RDB signals are used, while when writing, WDT and WDB signals are utilized to write data into the bit cells. SE and SP signals, generated by the control block, activate or deactivate the precharge circuit in the sense amplifier and enable the amplification of the differential voltage.
3. **Write Drivers:** These are employed by a group of columns in an SRAM array to control the behavior of the bit-line pair (BL and its complement) during a write operation. Since the bit-line pair is precharged to VDD before each operation, the write driver only needs to pull down one of the bit lines below the write margin of the SRAM cell based on the input data. Generally, write drivers consist of properly sized inverters or buffers crucial for a successful write operation and optimal flip time.
4. **I/O Latches:** These components are responsible for managing the flow of data in and out of the memory by utilizing the internal clock.
5. **Antenna Diode:** Its purpose is to prevent any antenna effects at the data terminals of the memory chip. Typically, these diodes are placed at the DIN and DOUT terminals.

3.7 Row Decoder

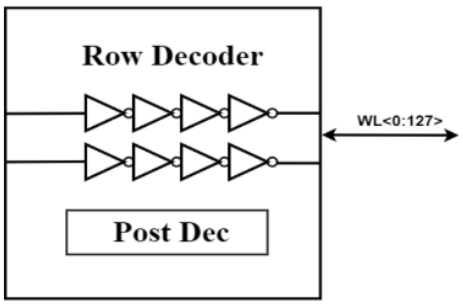


Fig.3.6 : Row Decoder

The post decoder generates the Word-line signal based on the decoded addresses received from the pre-decoder in the control block. The row decoder is responsible for selecting a single row of word-lines from a set of rows in the memory array based on the address bits. To achieve efficient row decoding, we have implemented a two-stage

decoding approach. This split or multi-stage decoder method has been found to be more effective for larger memories as it reduces the number of transistors, fan-in, power consumption, and loading on the address input buffers. The pre-decoding stage occurs in the control block, while the post-decoding stage takes place in the row decoder. The two-stage row decoding approach is adopted to simplify the complexity of the row decoder.

3.6 Control Block

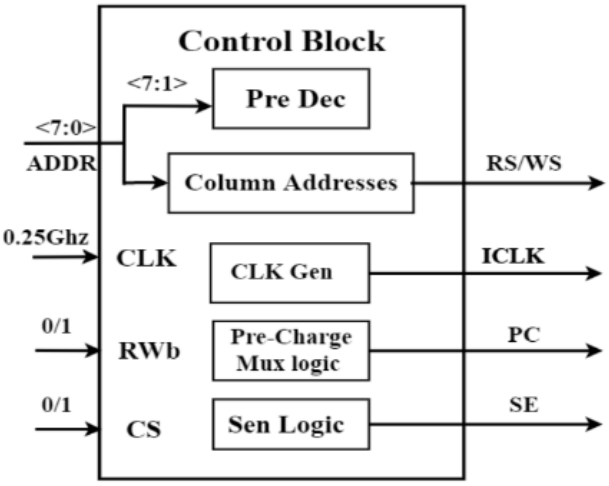


Fig.3.7 : Control Block

Control block in memory design is a very essential part, without it the memory cell cannot work properly. Signals which are required to operate the memory core are generated here.

1. **Clock Generation**: The Internal Clocks are generated from the External CLK pulse.

2. **Pre-Decoder Logic:** This logic module comprises the decoded address information obtained from the address bits. The decoded address is subsequently transmitted to the Row-Decoder block, where it generates the word-line responsible for selecting a memory cell during Read or Write operations, depending on the specified operation.

3. **RWb:** The Read Enable or Write Enable bar signal serves as an input to the control block, indicating the type of operation being performed. Based on this signal, the control block generates the corresponding logic required for the operation.

4. **Sense Logic:** This logic module generates the SE and SP signals that are utilized by the Sense Amplifier to enable the amplification of the differential voltage.

5. **Pre-Charge Mux Logic:** This logic module generates the PC signals required by the PreCharge MUX. These signals are used to precharge the bit-line pairs before a read or write operation.

6. **Column Address Logic :** This logic generates the RS and WS signals for Column MUX for Read and write operation.

7. **Chip Select :** This generates an internal Chip select signal which will enable/disable Read/Write operation in your memory. This will make sure that power consumption during no memory operation will be reduced and saved.

3.7 Bit-cell Design (Read, Standby, Write Mode)

The SRAM bitcell is the fundamental component of the SRAM array and is capable of storing a single bit of information. It enables non-destructive read operations, supports writing of data, and maintains data storage as long as it remains powered.

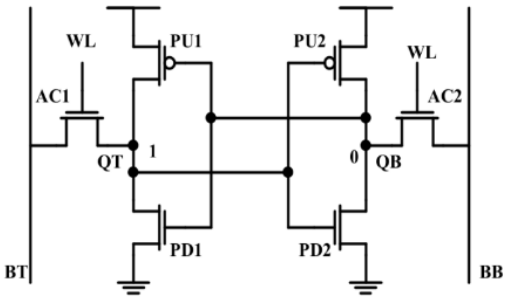


Fig.3.8 : Schematic of 6T SRAM Bit Cell

The standard 6T SRAM bitcell, illustrated in Figure 3.8 , consists of two cross-coupled inverters and two access transistors connected to each data storage node. The pull-up networks (PU1 and PU2) are PMOS devices, the pull-down networks (PD1 and PD2) are NMOS devices, and the access transistors (AC1 and AC2) are also NMOS devices within the bitcell. The latch formed by the inverter pair retains the binary information, and the true and complementary versions of the binary data are stored in the storage nodes (QT and QB).

During read and write operations, the access transistors allow access to the data storage nodes and isolate them from neighboring circuits during the hold state. The bit cells are accessed horizontally by activating the word-line (WL) during read and write operations. When the word-line of a row is asserted high, all the memory bit cells in the selected row become active and ready for read and write operations. The SRAM bit cell

operates in three modes: read, write, and standby, corresponding to reading, writing, or data retention states.

3.7.1 Bit cell in Standby mode

When the power supply to the memory chip is disconnected, the bit-cell is considered to be in standby mode. In Figure 3.9 , this corresponds to the word-line (WL) being low, causing both access transistors (AC1 and AC2) to be turned off. In this state, there is no access to the data storage nodes, and the data stored in QT and QB is preserved.

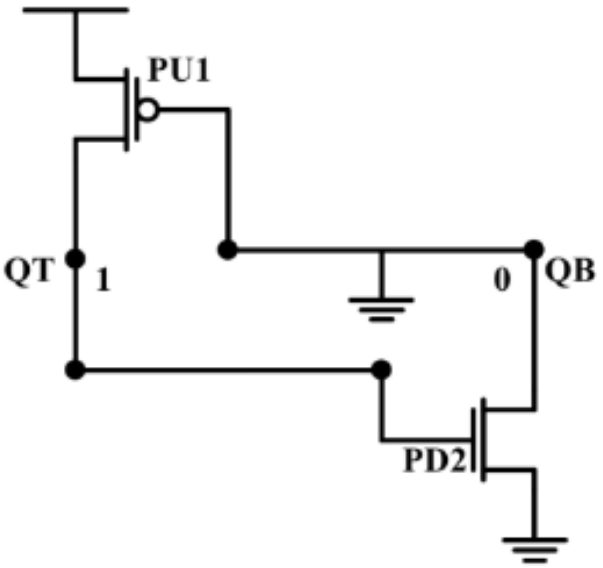


Fig.3.9 : Bit Cell in Standby mode

In Figure 3.9, assuming that the previous data bits stored are QT=1 and QB=0, the bit-cell circuit in standby mode is illustrated. With AC1 and AC2 turned off, QT=1 activates PD2, causing it to pull QB low, effectively preserving the data stored in QB. Similarly, with QB=0, PU1 is activated, pulling QT high to VDD, representing logic 1. Therefore, when the WL is low, indicating the bit-cell is in standby mode.

3.7.2 Bit cell in Read mode

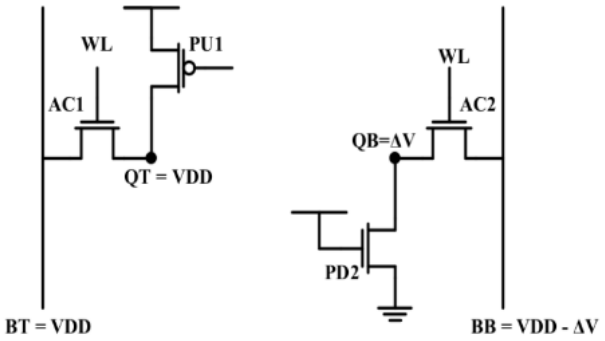


Fig.3.10 : Bit cell in Read mode

1. Assuming the data stored in QT and QB are 1 and 0 respectively, as depicted in Figure 3.10.
2. The bit-line pair, BT and BB, both need to be charged to VDD (logic 1).
3. When WL (word line) is set to 1, both access transistors are activated.
4. Since the data stored in QT and QB are 1 and 0 respectively, PU2 and PD1 are turned off, while PU1 and PD2 are turned on. This configuration can be seen in Figure 3.10.
5. In the above figure, QT and BT have the same voltage potential, resulting in no change. However, when examining QB and BB, the voltage difference causes BB to discharge through PD2.
6. In the effective circuit configuration shown, the MOSFETs are connected in an nMOS inverter arrangement. PD2 attempts to pull the QB node down to GND, while AC2 tries to pull it up to VDD.
7. Due to the conflict between AC2 and PD2, if we make the PD2 transistor larger, QB can be discharged faster than AC2's attempt to charge it to VDD.
8. This causes a slight rise in the QB node voltage to V, while the voltage at BB decreases by V. A successful read operation occurs when a differential voltage is developed across the bit-line pairs, with BB being pulled down slightly. The sense amplifier detects this differential voltage, allowing the data stored in QB to be read as "0". Similarly, when QT is 0 and QB is 1, a differential voltage is developed at BT, and during a read operation, the sense amplifier detects the data stored in QT as "1". Therefore, we can

conclude that for a successful read operation, PD2 must be stronger than AC2.

3.7.3 Bit cell in Write mode

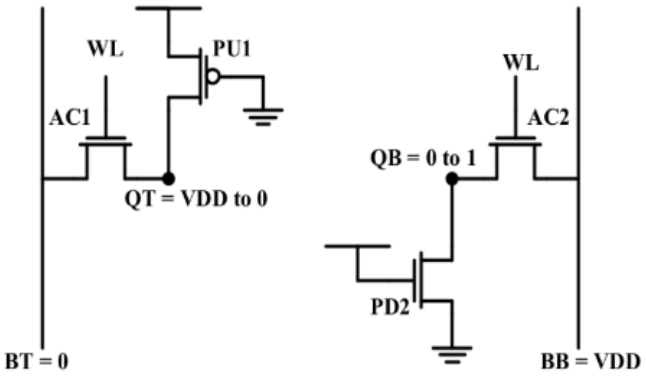


Fig.3.11 : Bit cell in write mode

1. Assuming the data stored in QT and QB are 1 and 0 respectively, as depicted in Figure 3.11.
2. During a write operation, the bit-line adjacent to the node storing 1 needs to be discharged to GND. In this case, BT is discharged to GND while BB is charged to VDD.
3. When WL (word line) is set to 1, both access transistors are activated.
4. Since the data stored in QT and QB are 1 and 0 respectively, PU2 and PD1 are turned off, while PU1 and PD2 are turned on. This results in the circuit configuration shown in Figure 3.11.
5. AC1 attempts to discharge QT through BT, while PU1 tries to charge QB to VDD.
6. On the other hand, AC2 aims to charge QB, while PD2 attempts to discharge QB to GND. As we have made PD2 stronger than AC2 during read operations, in order to maintain a value of 1 on QB, PD2 needs to be turned off. To achieve this, QT must be pulled down to a certain point where the inverter switches.
7. To create this condition, AC1 should discharge QT faster than PU1 charges it. Hence, in a write operation, we always begin by writing 0 at the node where 1 is stored. Similarly, when QT is 0 and QB is 1, BB is discharged to GND during the write operation. Consequently, the data is

overwritten at QB with "0" and at QT with "1". Therefore, we can conclude that for a successful write operation, AC1 must be stronger than PU1.

3.8 Characterization

Characterization of bitcell involves experimentation of the following parameters

1. Read current (Iread)
2. Leak current (Ileak)
3. Write Margin (BT and WL driven)

3.8.1 Read Current

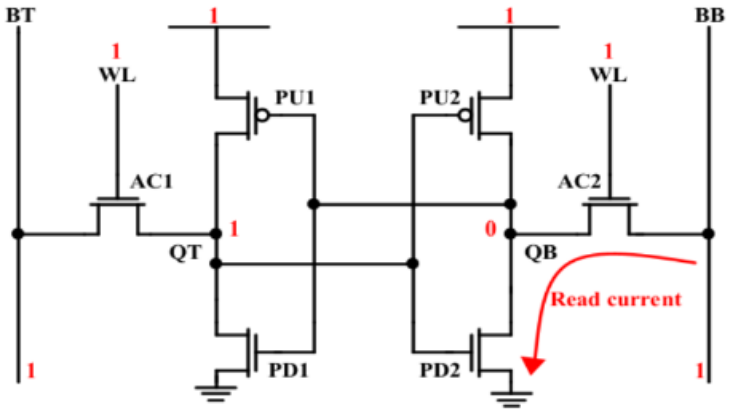


Fig.3.12 : Iread-DC characterization

This parameter refers to the current observed at the node that holds a value of zero during a read operation. It plays a role in determining the speed of the read operation, which is influenced by the sizes of the access and pull-down transistors. When the read current increases, the discharge rate of the BT/BB line becomes faster, resulting in an enhanced read operation speed.

1. Initially, QT is set to 1 (VDD), and QB is set to 0.
2. The WL (word line) is connected to VDD.
3. BT and BB are initialized to VDD.
4. The current flowing through the access transistor adjacent to the node containing zero is measured.

3.8.2 Leak Current

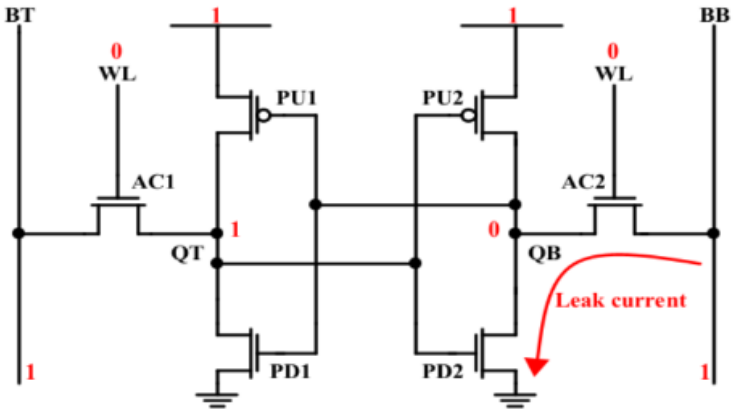


Fig.3.13 : Ileak-DC Characterization

This parameter refers to the current measured at the node that holds a value of zero when the access transistors are turned off. It determines the maximum number of bit cells that can be accommodated per column in a bank. The size of the access transistors plays a role in determining this current. It is desirable to have bit cells with lower leakage current, as it allows for a greater number of bit cells to be placed in a column, reducing the overall area occupied.

The DC characterization of the leakage current is as follows:

1. Initially, QT is set to 1 (VDD), and QB is set to 0.
2. The WL (word line) is connected to GND (AC1 and AC2 are turned off).
3. BT and BB are initialized to VDD.
4. The current flowing through the access transistor adjacent to the node containing zero is measured.

3.8.3 Write Margin

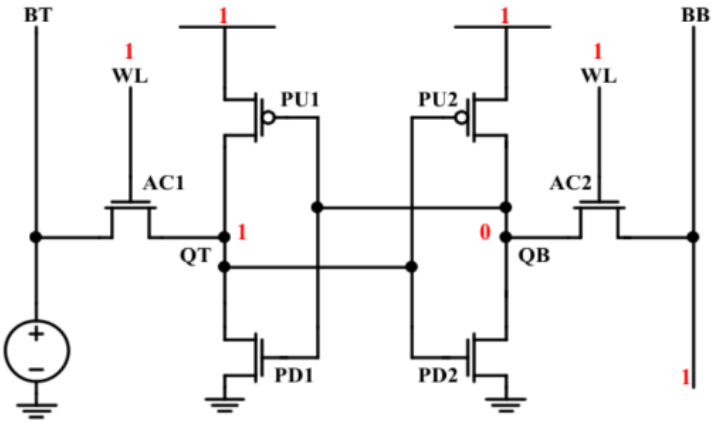


Fig.3.14 : DC Characterization on Write Margin

The Write Margin is a crucial factor in ensuring the writability of a memory cell, balancing the need to write data effectively while conserving energy by minimizing the voltage required to pull down the bit-line voltage to zero. The Write Margin represents the maximum voltage on BT/BB (BT driven) or the minimum voltage on WL (WL driven) at which the data flips, indicating a successful write operation. It measures how favorably a bit cell responds to a write operation and is also referred to as the Write-Trip Point. The Write Margin is determined by the characteristics of the access and pull-down transistors.

Here are the steps to determine the Write Margin:

1. Initially, QT is set to 1 (VDD), and QB is set to 0.
2. The WL (word line) is connected to VDD.
3. BB is initialized to VDD.
4. A gradually decreasing voltage is applied to BT, and the QT and QB nodes are continuously monitored. The voltage level at BT is recorded when the data stored in QT or QB flips.

3.9 Schematic design and Layout

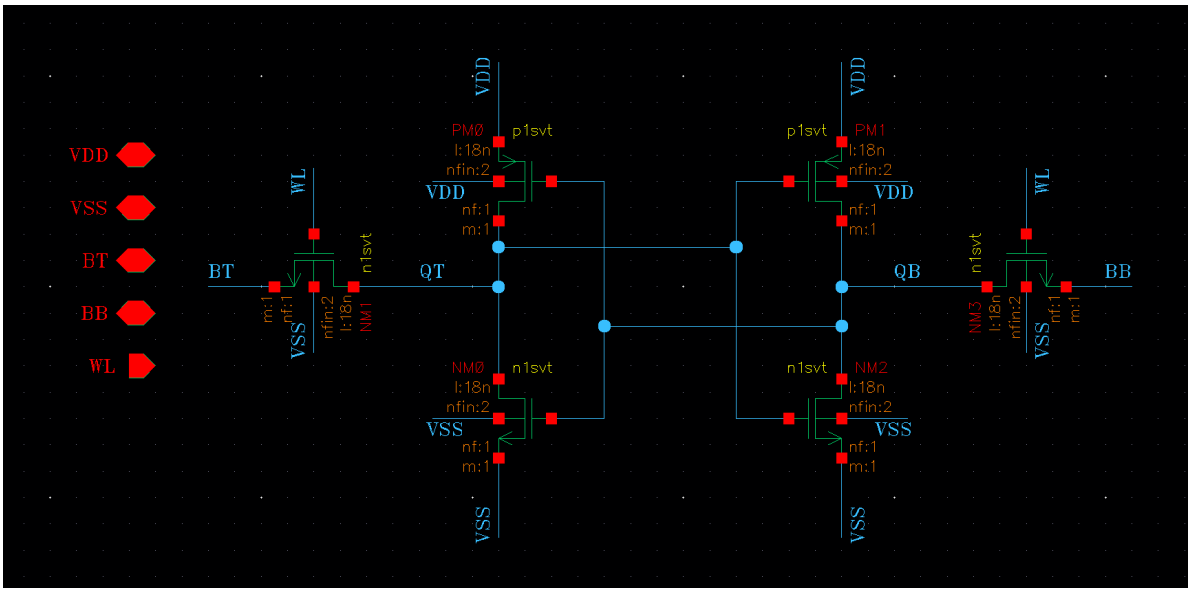


Fig 3.15 : Bitcell Schematic

3.9.1 Bit-cell design and Layout

The bit-cell serves as a fundamental component in SRAM memory, and multiple bit-cells are combined to form a single memory core. After designing the bit-cell, a layout is created with careful consideration of the following criteria to ensure a compact design with minimal DRC (Design Rule Check) and LVS (Layout versus Schematic) issues.

1. To create a full cell design with 6 transistors, start by designing a half cell layout for 3 transistors. Then, flip and position the half cell horizontally and vertically to achieve the desired full cell layout.
2. Optimize space utilization by sharing the source and drain connections of transistors AC1, PD1, AC2, and PD2.
3. Improve efficiency by sharing the BT (Bit-Line True) and BB (Bit-Line Bar) signals across columns
4. Enhance compactness by sharing the WL (Word Line) signal across rows.
5. Maximize space utilization by sharing contacts on the PR (Poly Resistor) boundary.
6. Maintain a consistent layout pattern by keeping the poly (poly-silicon) layer vertical, M1 (Metal Layer 1) and M3 (Metal Layer 3) horizontal, and M2 (Metal Layer 2) and M4 (Metal Layer 4) vertical. Use M1 for BB and BT signals, M2 for WL signals, and M3 and M4 for a strong power mesh.
7. Ensure signal integrity by implementing shielding for the BT/BB and WL signals.

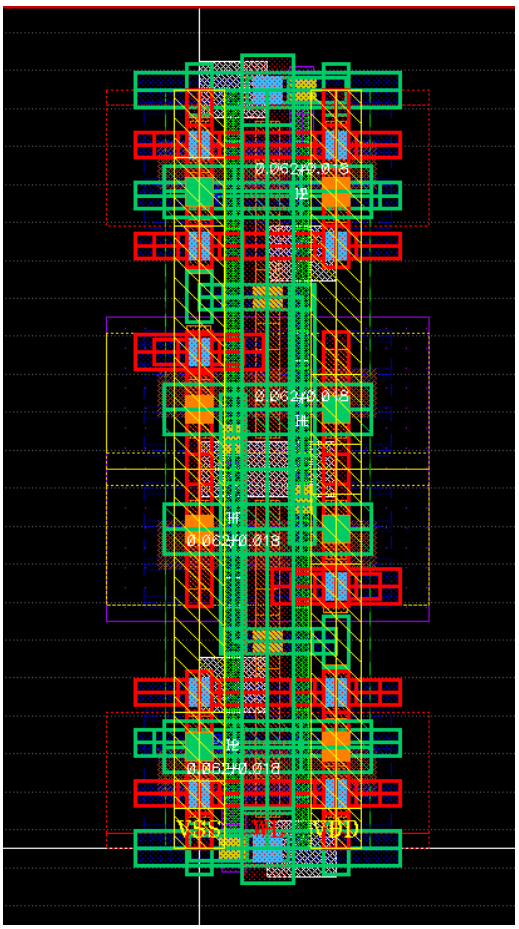


Fig.3.16 : Layout design for 6T SRAM High Density Bitcell

From the provided figure 3.16, illustrates the arrangement of devices where NMOS devices are positioned at the origin and PMOS devices at the top. The half cell is flipped both horizontally and vertically to ensure the continuity of the NWELL region for PMOS devices. The allocation of the M1 metal layer is shown, with green masked M1 used for bit-cell signals (BT/BB and WL) and red masked M1 used for shielding (VDD and VSS signals). L-shaped M1 metals represent QT and QB nodes, and cut metals are utilized at the end of each of these metals to isolate the QT/QB voltages for each bitcell within a memory core. Finally, displays the complete design with all enabled layers.

3.9.2 Edge Cells

Edge cells are designed to prevent abrupt endings and to facilitate the connection between peripheral circuits and the bit cell array. These cells have body connection taps that are utilized for LVS (Layout vs. Schematic) verification. The creation of these edge cells involves using the bit cell and preventing channel formation in the devices through the utilization of a CutActive layer. Various types of edge cells have been designed for this purpose.

1. The left edge cell is created using the bitcell design. The CutActive layer is employed to prevent the active region, while the Nwell-tap and sub-tap are sized to match the height of the bitcell and incorporated alongside it. The power mesh follows a similar pattern as observed in the bitcell.

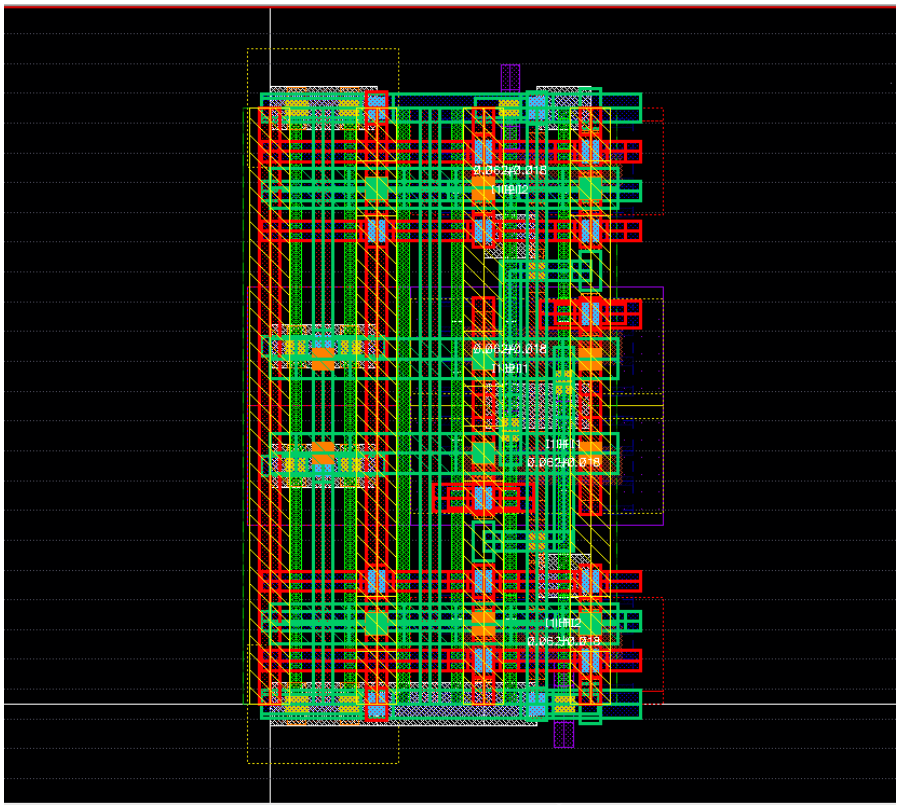


Fig.3.17 : Left Edge Cell

2. The bottom edge cell is formulated by taking into account the NMOS transistors positioned on the axis and employing the CutActive layer to prevent channel formation in the transistors. The layout of the bottom edge cell.

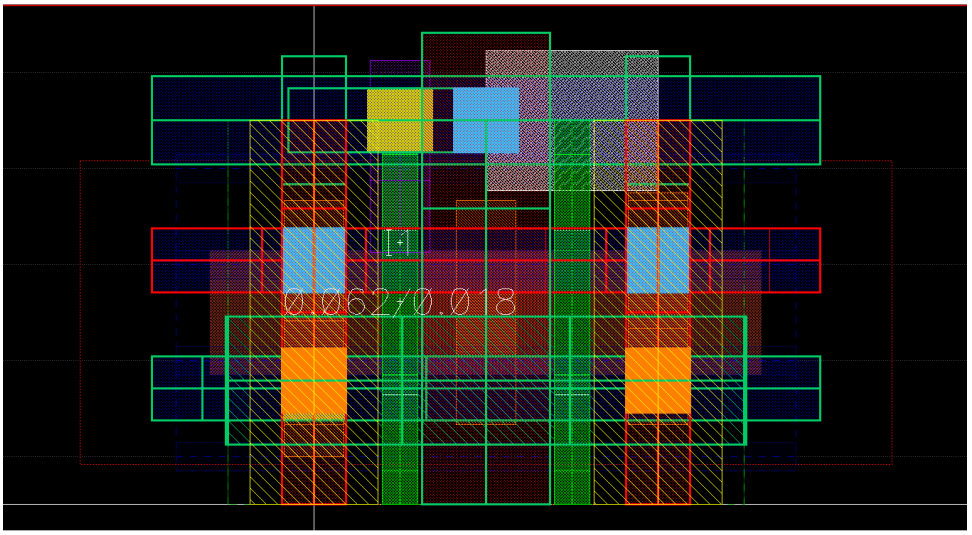


Fig.3.18 : Bottom Edge Cell

3. The corner edge cell is created by utilizing the corner portion of the left edge cell..

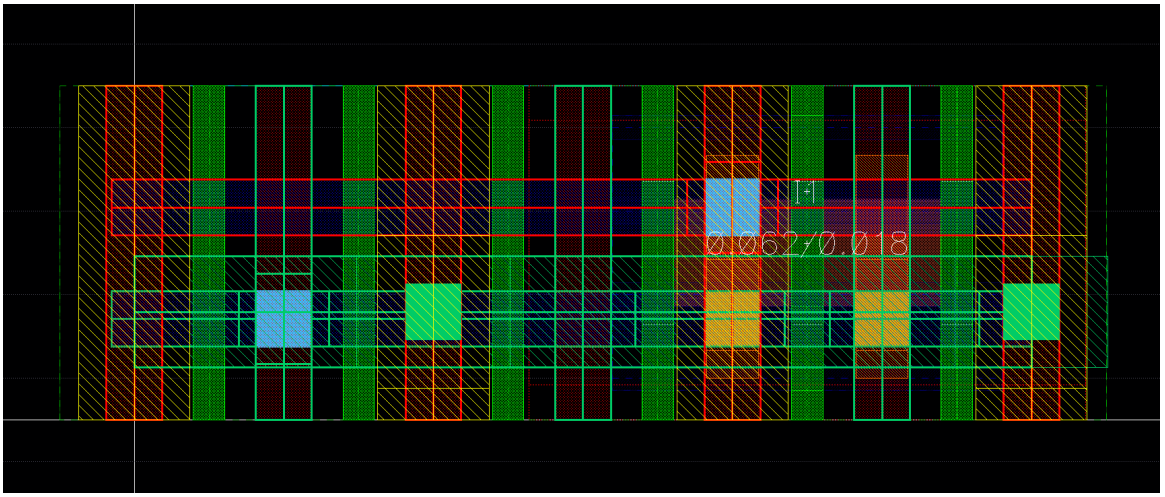


Fig.3.19 : Corner Edge Cell

4. The mid-tap cell is constructed by employing two left edge cells, with one of them being flipped vertically to serve as a right edge cell. The Nwell-tap and sub-tap connections of the two cells are overlapped to create the mid-tap.

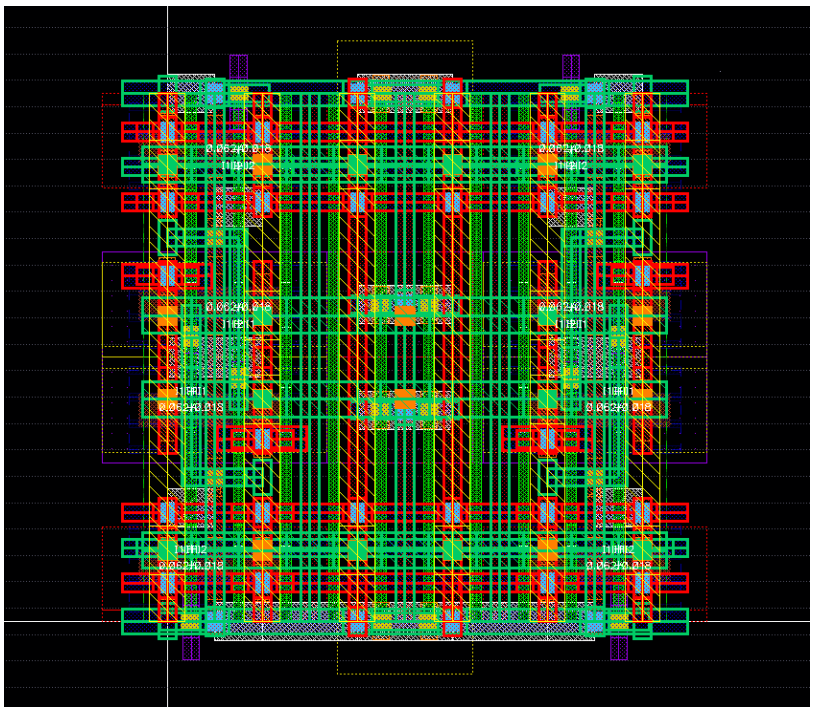


Fig.3.20 : MidTap

5. The layout for the mid-tap corner cell is created by following the same guidelines as the bottom edge cell. It showcases the layout for the mid-tap corner cell.

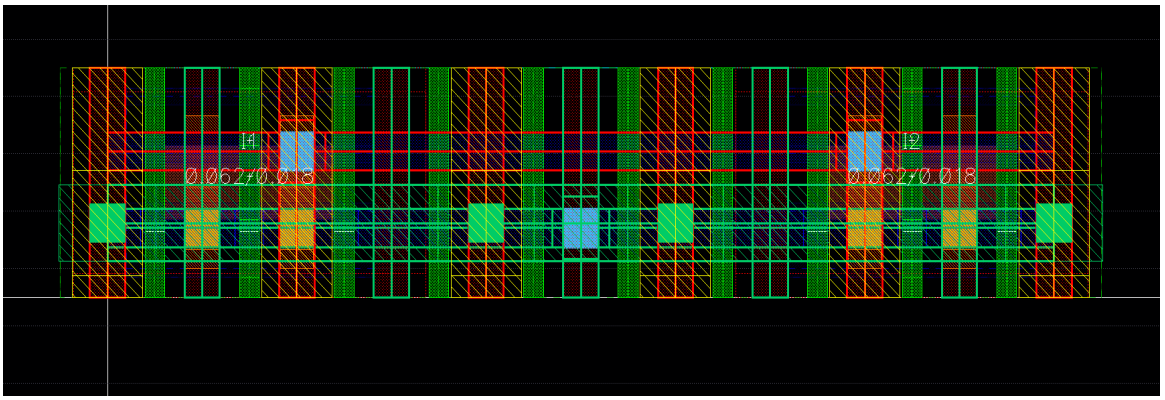


Fig.3.21 : MidTapBottomCell

The bit-cell array can be implemented using the designed layouts for the bit-cell and edge cells. It illustrates the floorplan for a 4x4 array, where the arrangement of edge cells surrounding the bit-cell array, and displays the layers involved. The horizontal metal layers, specifically the M1 metal layer, carry the BT/BB signals to the column IO located

adjacent to the memory core. The WL signal is routed in the M2 metal layer, which runs vertically to the row decoder positioned on top of the memory core.

3.9.3 Column I/O design and Layout

The column IO circuitry comprises different circuits that are utilized for the selection of columns within the memory core. It enables the selection of specific bit-line pairs from a group of bit-line pairs in the chosen row. These circuits are responsible for the read and write operations, facilitating the transfer of data bits into and out of the memory array..

3.9.4 Pre Charge MUX

The PreCharge MUX is employed to pre-charge the bit-line pairs prior to each read and write operation. It utilizes PMOS transistors, which are effective in passing logic 1 signals. A column MUX connects multiple columns to a single Sense Amplifier (SA), offering separate channels for read and write operations. In most compiler designs, MUX 2/4/8/16 configurations are commonly used. In the presented design, the precharge MUX is implemented using the MUX 2 technique.

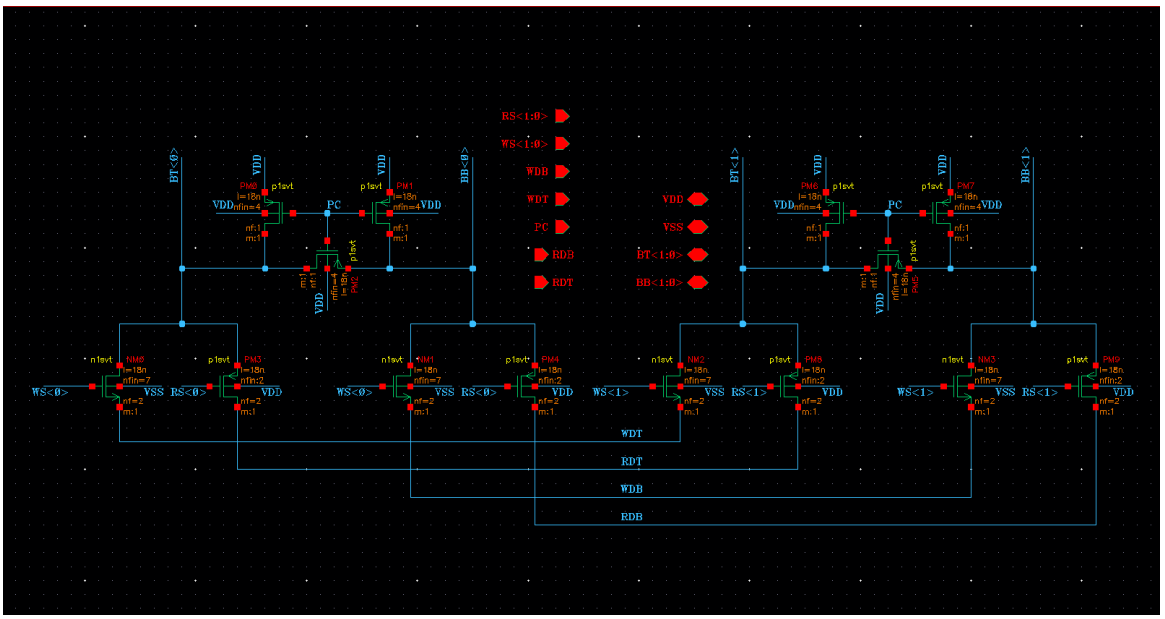


Fig.3.22 : Precharge Schematic

The precharge MUX is controlled by the PC signal originating from the control block. When the PC signal is low, the PMOS transistors are activated, precharging the bit-lines to the VDD voltage level. An equalizer PMOS is included in the MUX to minimize any voltage differences between the bit-line pairs. The read/write operation occurs only

when the PC signal is high, causing the PMOS transistors to turn off. To establish separate paths for reading and writing into the bit-cells, a read select network (PMOS) and a write select network (NMOS) are designed to operate based on the RS and WS signals generated by the control block. The RDT/RDB signals enable the read path, while the WDT/WDB signals enable the write path within the MUX. Generally, the size of the precharge devices is kept at least twice the size of the access transistors present in the bit-cell.

Layout Design guidelines for Pre Charge MUX:

1. The column IO is designed with a vertical height equivalent to two bit-cells.
2. The design follows a half-cell approach to accommodate the implementation of two precharge circuits.
3. The connections for the top level of the Precharge layout are represented by RDT/RDB and WDT/WDB.
4. Tap cells are not required in this case, as the precharge MUX is positioned adjacent to the edge cells of the memory array block.

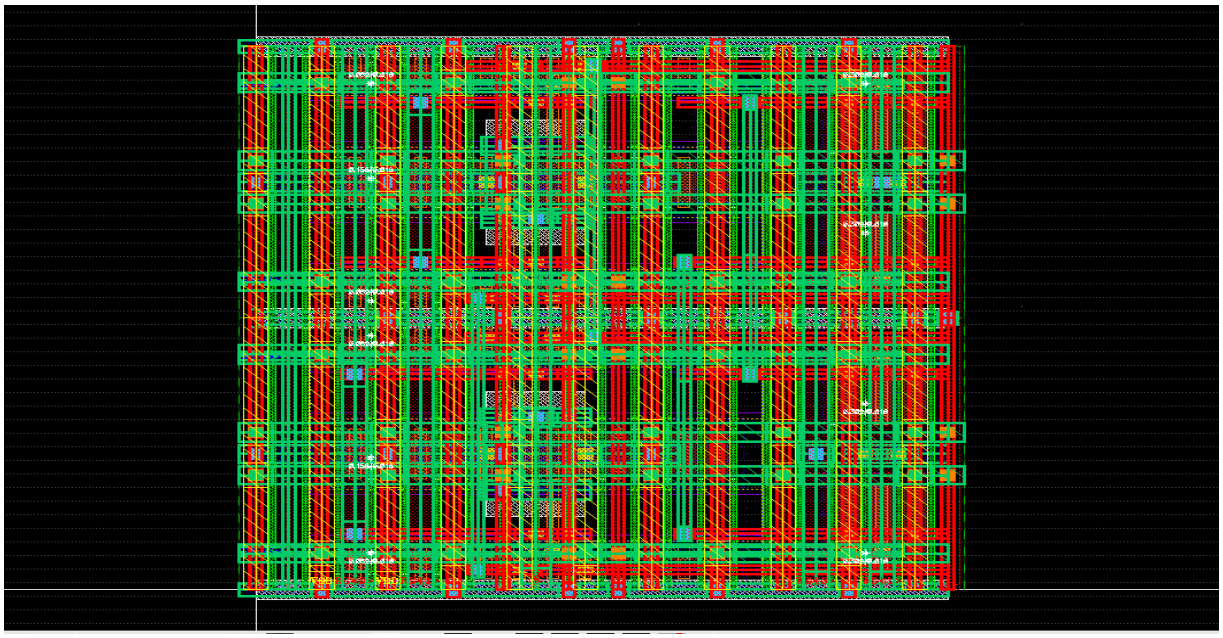


Fig.3.23 : Precharge mux layout

3.9.5 Sense Amplifier

The Sense Amplifier circuit is utilized for reading data from the bitcell. It consists of pMOS pass transistors that are controlled by the SE (SA Enable) signal. Additionally, there is a Precharge and Equalizer circuit controlled by the SP (SA Precharge Enable) signal, a Tail nMOS transistor controlled by SE to enable differential voltage amplification, and two inverters to convert the double-ended output to a single-ended DOUT. The schematic implementation of a Sense Amplifier is illustrated in Figure 3.24.

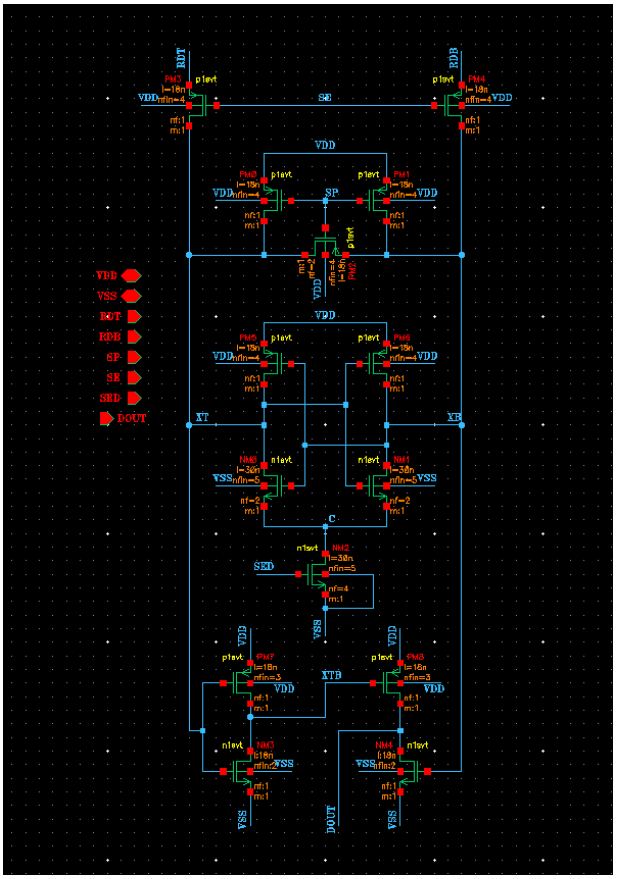


Fig.3.24 : Sense Amplifier Schematic

Layout Design guidelines for Sense Amplifier

1. To ensure an even number of devices, it is recommended to employ the Half cell approach.
2. Utilizing Euler's path can help maintain a continuous oxide area across the precharge, differential amplifier, and dual-ended output to single-ended output converter.
3. Since the differential amplifier in the SA is an analog block, it should be placed within a guard ring in the layout to prevent any latch-up issues.

4. It is advisable to include a minimum of 5 dummy devices on each side for p-devices to mitigate any LOD (Latch-Up-Induced-Double-Diffusion) effects.
5. Since the WDT/WDB signals from the Precharge MUX are not utilized in the Sense amplifier, it is sufficient to maintain continuity throughout the design so that they can be directly connected to the write drivers.

Fig.3.25 : Sense amplifier layout

3.9.6 Write Driver

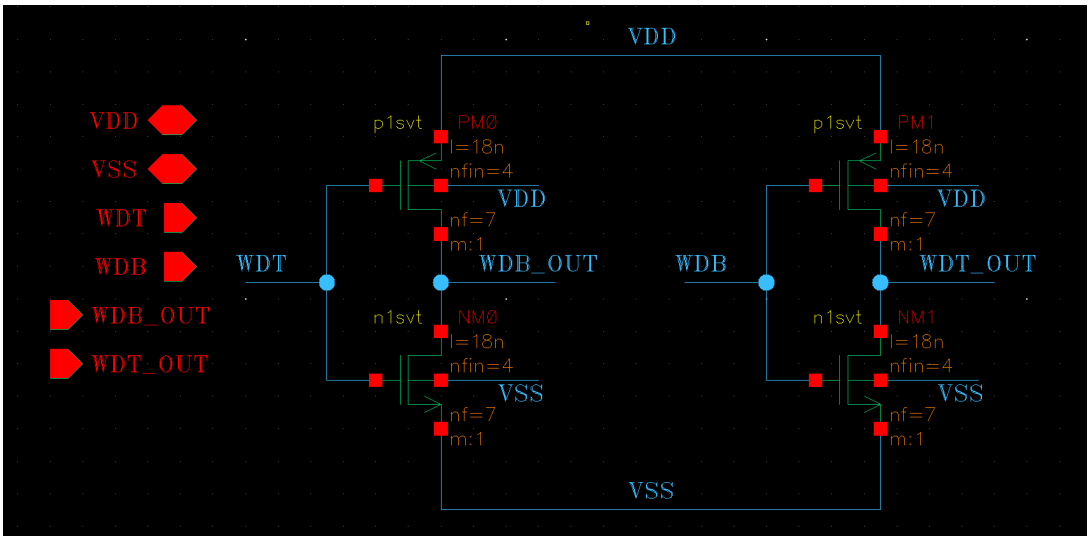


Fig.3.26 :. Write driver schematic

The write drivers, which are typically an inverter/buffer, play a crucial role in driving the write MUX. Proper sizing of the write drivers is essential to ensure a successful write

operation and achieve an optimal flip time. Figure 3.26 illustrates the schematic implementation of the write drivers.

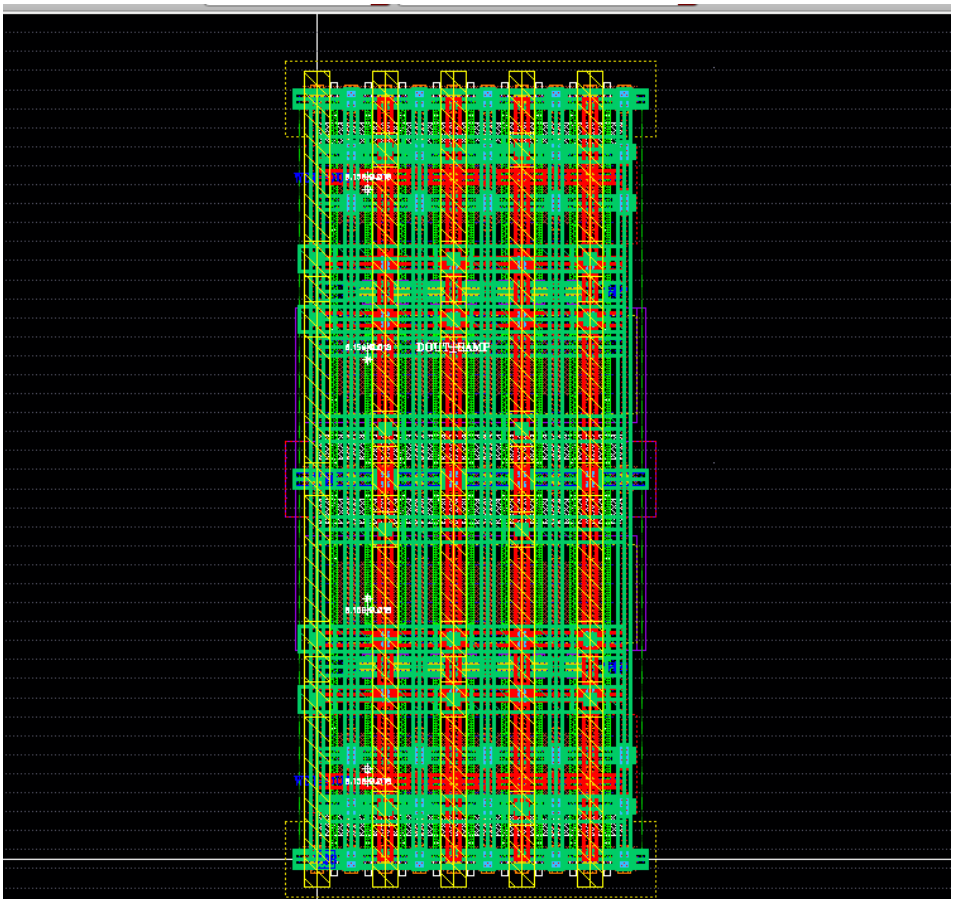


Fig.3.27 : Write driver Layout

3.9.7 I/O Latch

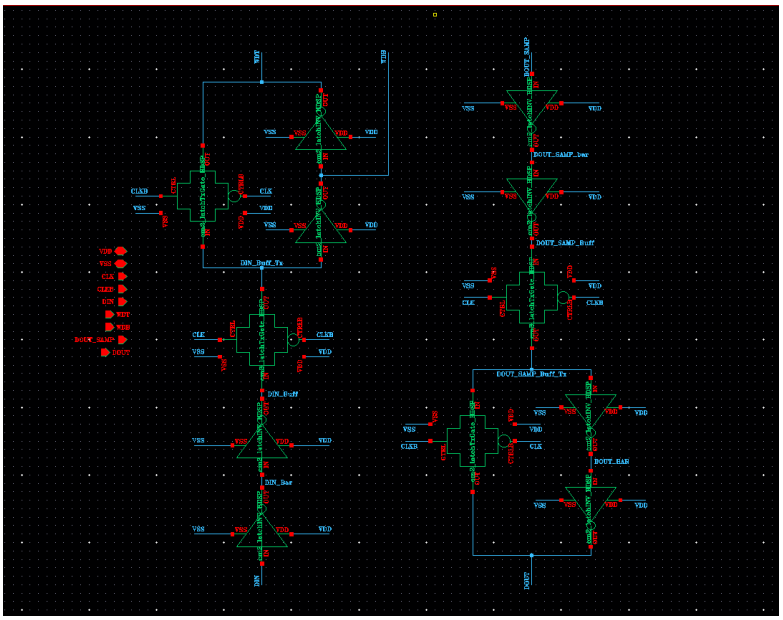


Fig.3.28 : I/O latch schematic

This circuit plays a crucial role in managing the flow of data into and out of the memory. Latches are utilized to control the movement of data by holding and synchronizing it with an internal clock signal. Figure 3.28, depicts the schematic representation of the latches, while figure 3.29, illustrates their layout implementation.

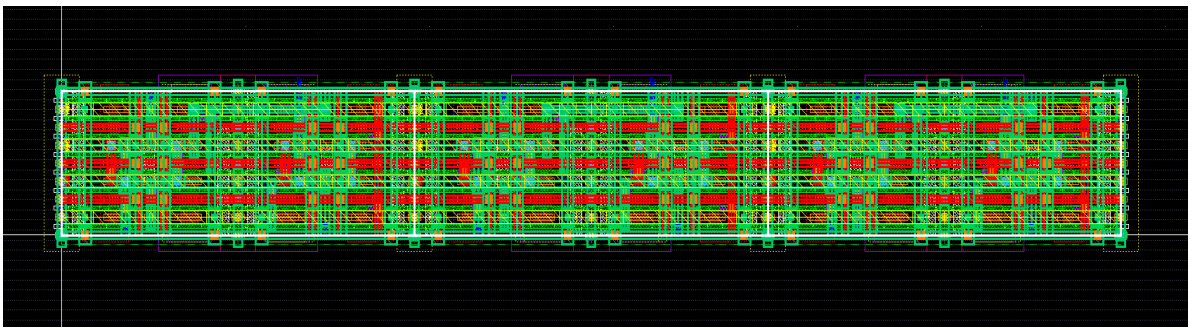


Fig.3.29 : I/O Latch Layout

3.9.8 Antenna Diode

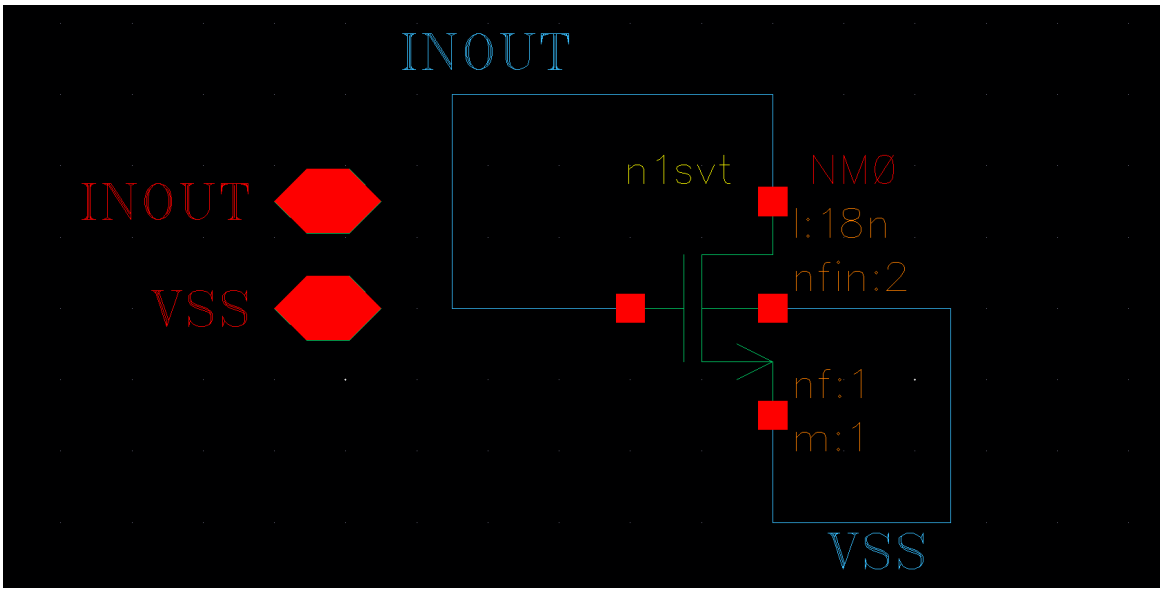


Fig.3.30 : Antenna diode schematic

An antenna diode is employed to mitigate any potential antenna effect that may arise at the data terminals (DIN and DOUT). In this design, an NMOS is utilized with its drain and gate connected to the same potential. Figure 3.30 presents the schematic representation of the antenna diode, while figure 3.31 showcases the layout design for the antenna diode.

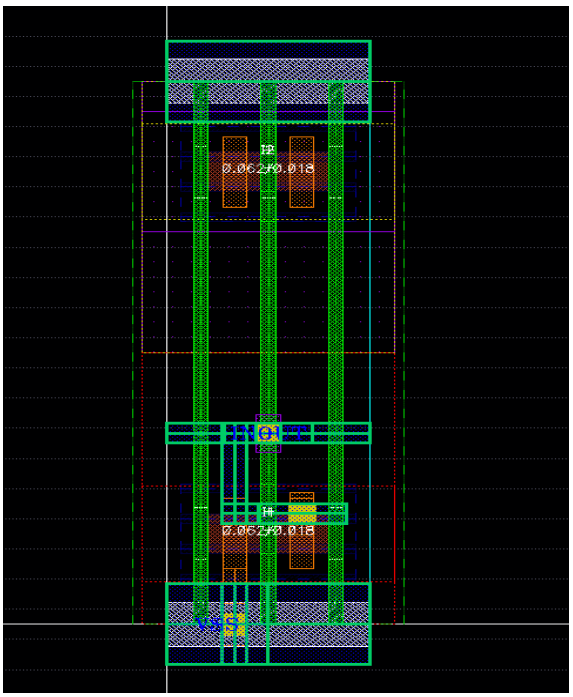


Fig.3.31 : Antenna diode Layout

3.9.9 Row Decoder

Row decoders are essential for selecting a specific row of word-lines from a set of rows in the memory array based on address bits. In our design, we have implemented a two-stage decoding process to accomplish row decoding. This approach, known as split or multi-stage decoding, has been found to be more efficient for larger memories. It helps in reducing the number of transistors, fan-in, power consumption, and loading on the address input buffers. The pre-decoding stage occurs in the control block, while the post-decoding stage takes place in the row decoder. The decision to employ a two-stage row decoding technique is motivated by the desire to simplify the row decoder and reduce its overall complexity.

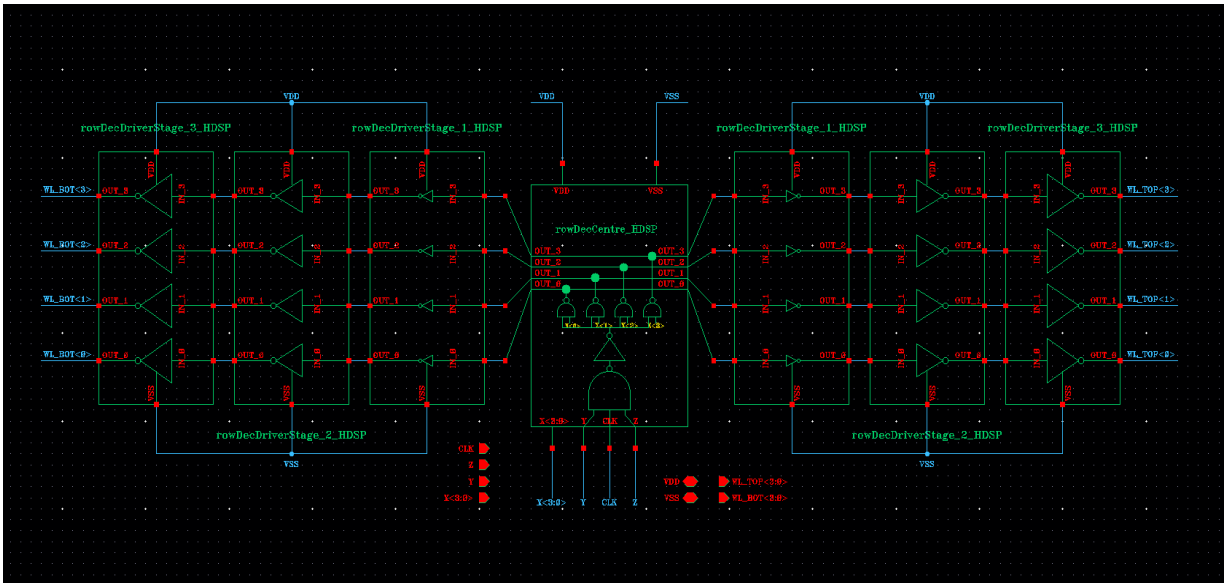


Fig.3.32 : RowDecoder schematic

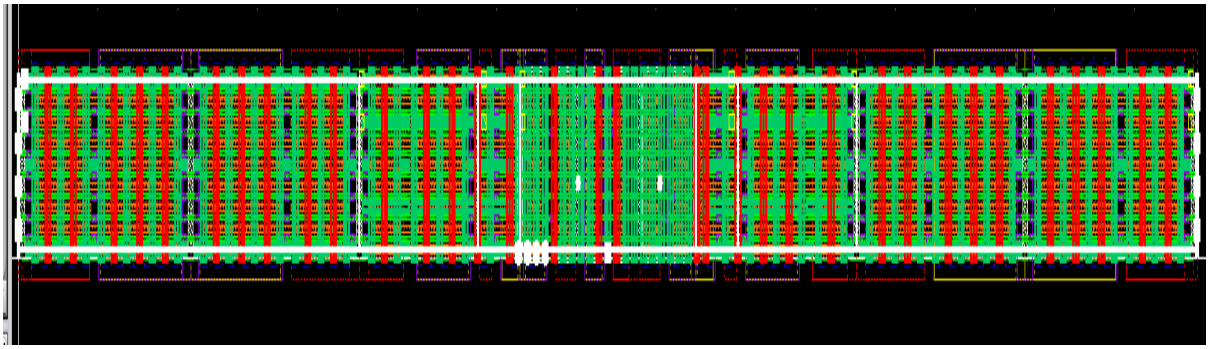


Fig.3.33 : Row Decoder Driver Stage1 layout

3.9.10 Control Block

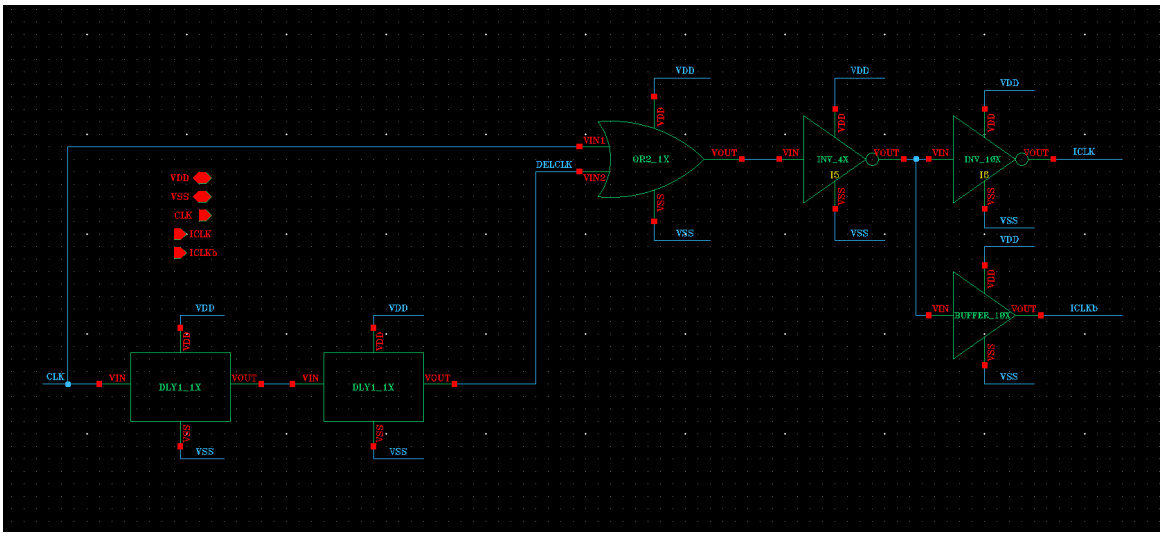


Fig.3.34 : iClkGen schematic

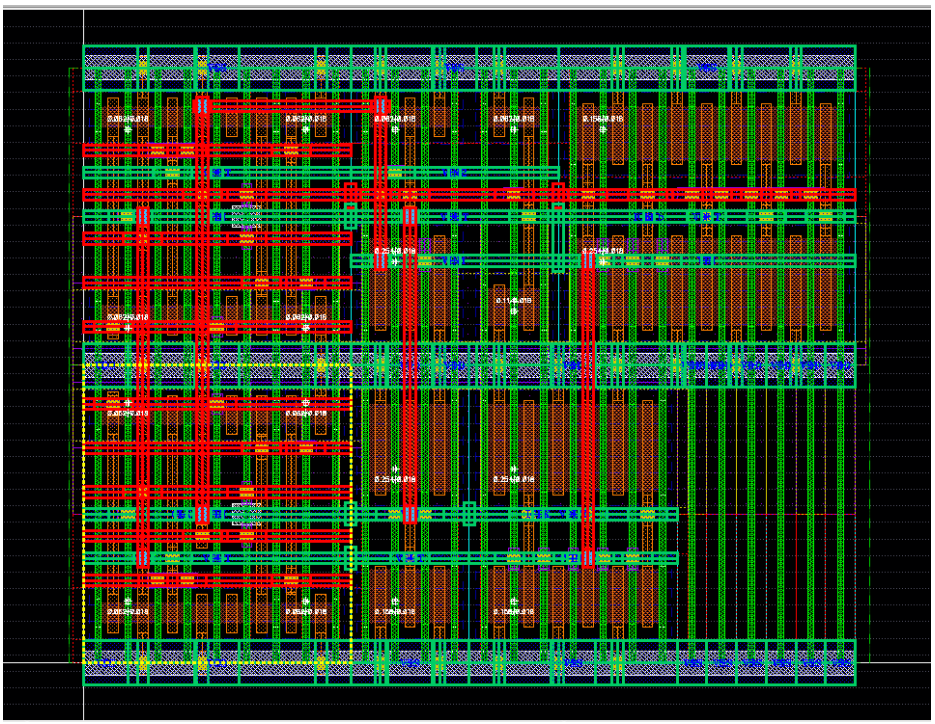


Fig.3.35 : iClkGen layout

Within this project, we have incorporated two distinct varieties of delay blocks, namely DELAY1 1X and DELAY4 1X. The former generates a delay of 109ps, while the latter produces a delay of 355ps. Each of these blocks employs 4 and 10 inverter stages,

respectively, to achieve the desired delay. Notably, due to the even number of inverter stages employed, the output signal remains in-phase, ensuring that an inverted output is not obtained.

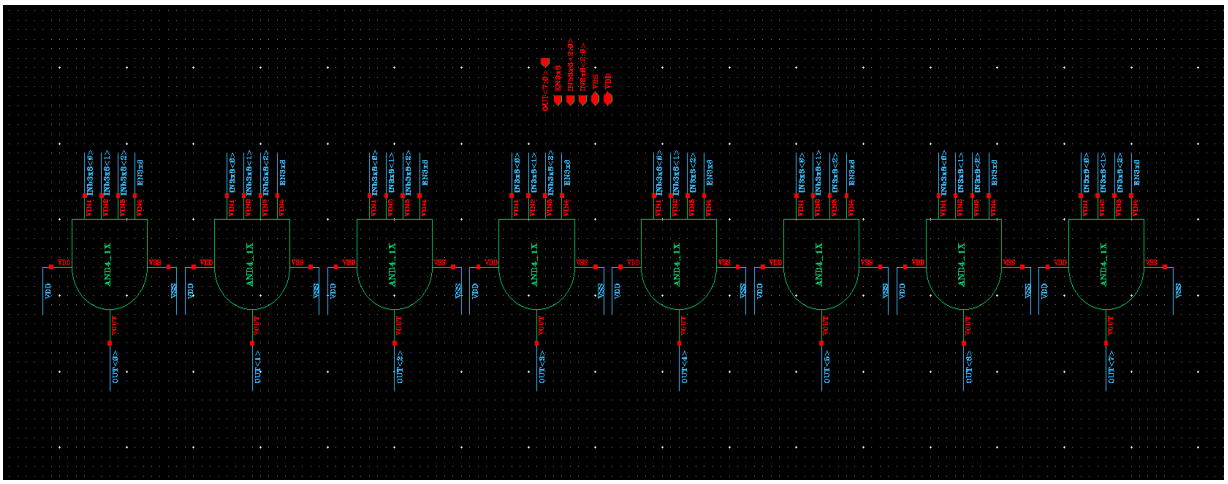


Fig.3.36 : 3x8 decoder schematic

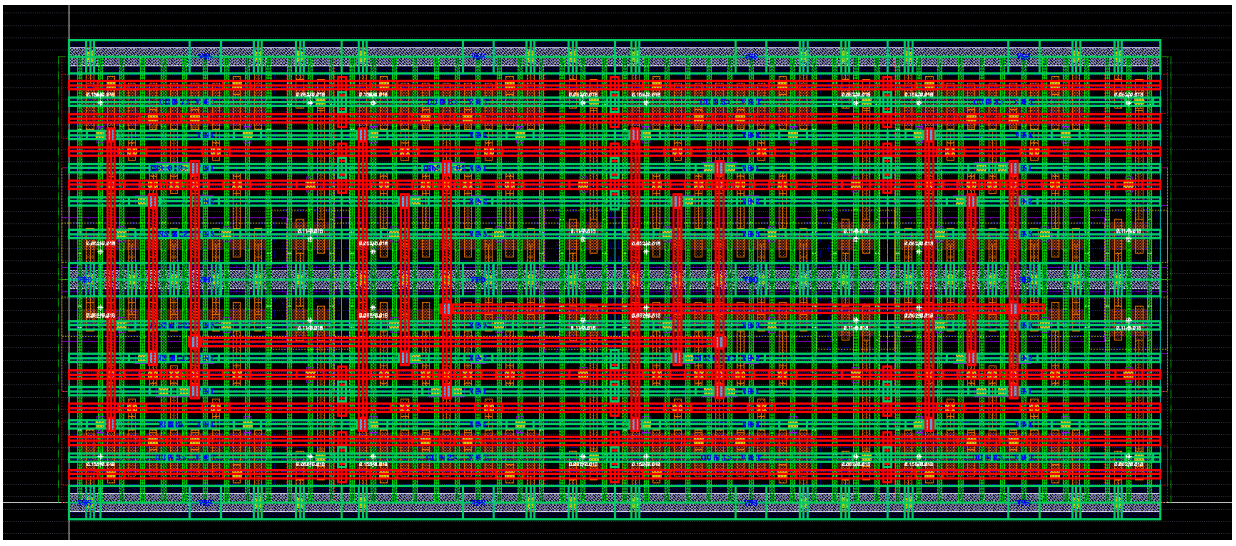


Fig.3.37 : 3x8 decoder layout

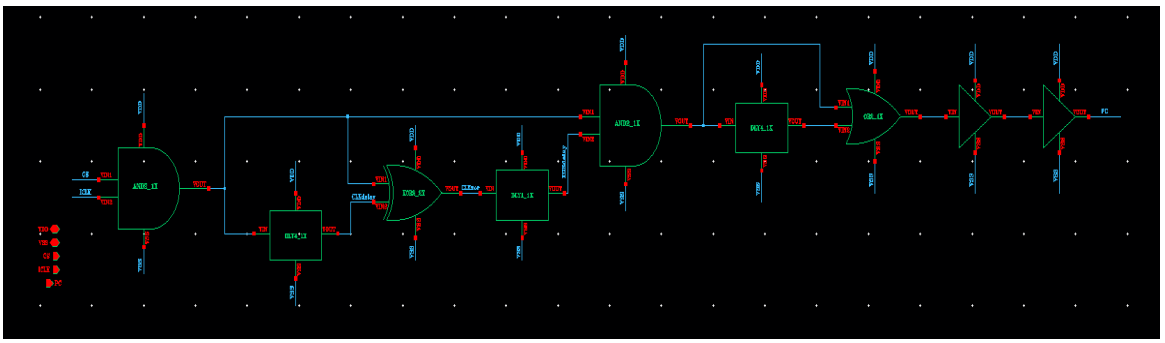


Fig.3.38 : pcGenClk schematics

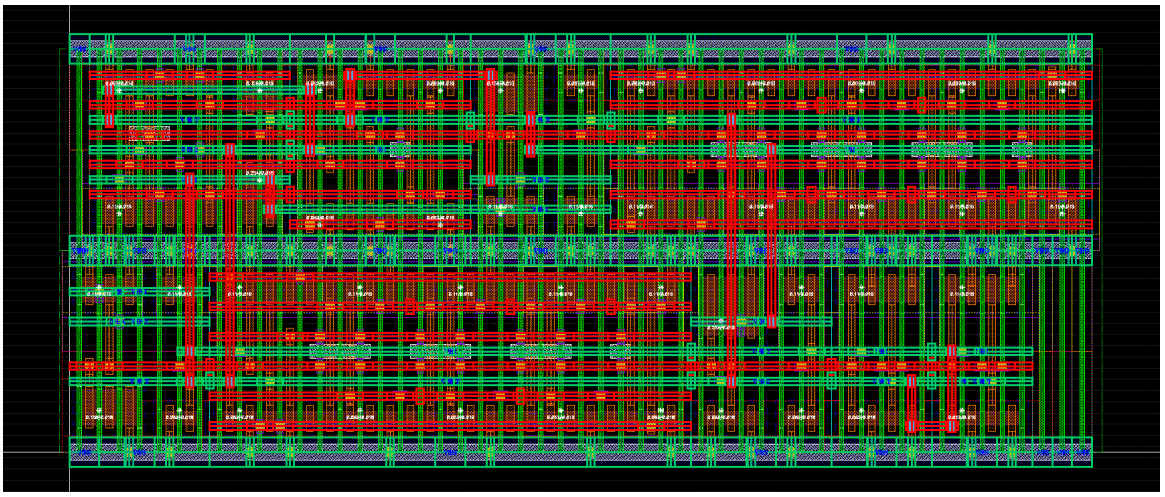


Fig.3.39 : pcGenClk layout

3.10 Compiler Design

3.10.1 SKILL IDE and language

The SRAM compiler design heavily relies on the SKILL IDE and the SKILL programming language. Automating the generation of the SRAM array and the placement of peripheral circuits necessitated the utilization of Cadence's scripting language and the associated development tool.

3.10.2 SKILL IDE

The development environment greatly assists in the design of SKILL functions and programs. It includes a graphical user interface (GUI) for code writing, a "Finder" tool for accessing built-in SKILL functions, and a debugging tool. Figure 3.40, showcases the SKILL IDE and provides sample code from this compiler design. While it is also possible to write SKILL code in other text editors or terminal-based editors like VIM, the SKILL IDE offers the aforementioned features, which help streamline the process of writing and verifying SKILL code, ultimately saving time.

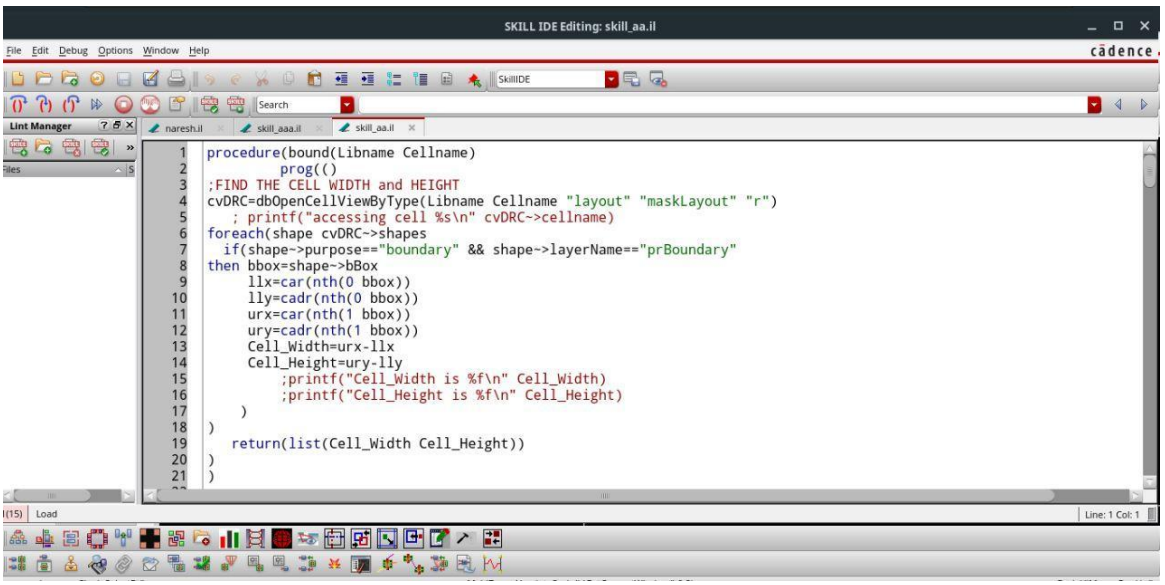


Fig.3.40 : SKILL IDE

3.10.3 SKILL Language

The SKILL language, developed by Cadence, is a scripting language based on the high-level LISP programming language [R]. It combines elements of the C programming language with the functionality of a higher-level scripting language. This unique combination allows for automation and control of various Cadence software tools and

their features. With SKILL, users can access, create, and modify database items and designs, perform schematic and layout operations, manage graphics controls, and more.

It's worth noting that the extensive capabilities of the SKILL language as a CAD development tool go beyond the requirements of this project. In this section, we will focus on the key SKILL functions relevant to this compiler design. Before diving into the SKILL functions, it's important to understand the file structure and execution of SKILL code. SKILL files have a ".il" extension, and all activities related to SKILL files occur within the Command Interpreter Window (CIW). The CIW serves multiple purposes, such as managing the design session, launching the file manager and the SKILL IDE, and executing specific SKILL commands and functions.

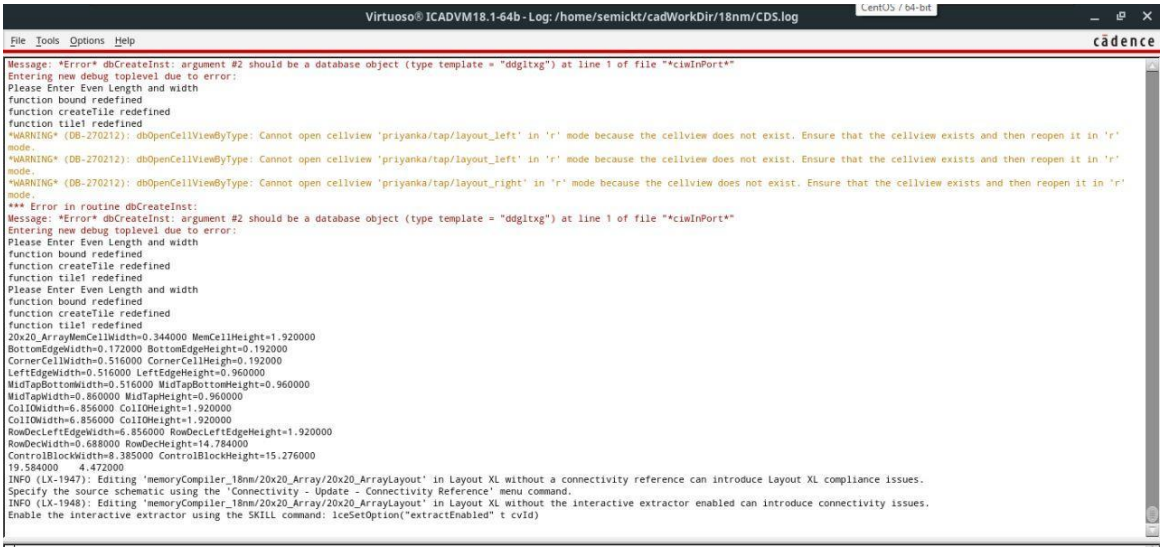


Fig.3.41 : Cadence CIW

3.10.4 SKILL Compiler Design

The majority of the SRAM compiler revolves around the design of the SKILL files responsible for generating the SRAM layouts. This involves utilizing key SKILL functions to create the compiler itself and handle tasks such as generating SRAM arrays, placing edge cells, and arranging peripheral blocks. This section provides comprehensive coverage of these essential SKILL functions and their role in the SRAM compiler.

3.10.5 SKILL Layout Functions

The primary SKILL functions utilized in this design involves manipulating cell view layouts through database objects, finding the bounding box of parameterized cells,

placing parameterized cells, pins, and terminals. Below, we explain the main SKILL functions used:

1. dbOpenCellViewByType: This function opens a cell view, allowing various modes such as read, append, write, or scratch. When opened in read mode, the cell view must already exist. If opened in other modes, the function creates the cell view if it doesn't exist. It is used to determine the bounding boxes of required instances, which represent the coordinates of the rectangular perimeter.
2. dbCreateInst: With this function, an instance is created in the cell view using specified origin coordinates (l point) and orientation (t orient). It is used to place the required instances at specific x and y coordinates.
3. dbSave: This function saves the modifications made to a cell view that has been opened in write or append mode. Optionally, it can specify a different cell view where the data is to be saved. If a different cell view is specified, its contents overwrite the old cell view. A cell view opened in read or scratch mode cannot be saved or overwritten. This function is used to save the final cell view (layout) once all instance placements are completed.
4. dbClose: This function closes a cell view. When the number of close operations matches the number of open operations, the cell view is purged from virtual memory, freeing up memory for subsequent dbOpenCellViewByType calls. The system internally keeps track of the open and close counts.
5. hiDisplayForm: This function creates a graphical user interface (GUI) window with user-designed fields and parameters. It allows for a customized interface to be displayed.

These SKILL functions play a crucial role in manipulating and managing the layout and placement of cell views in the SRAM compiler design.

3.10.5 SKILL Compiler Overview

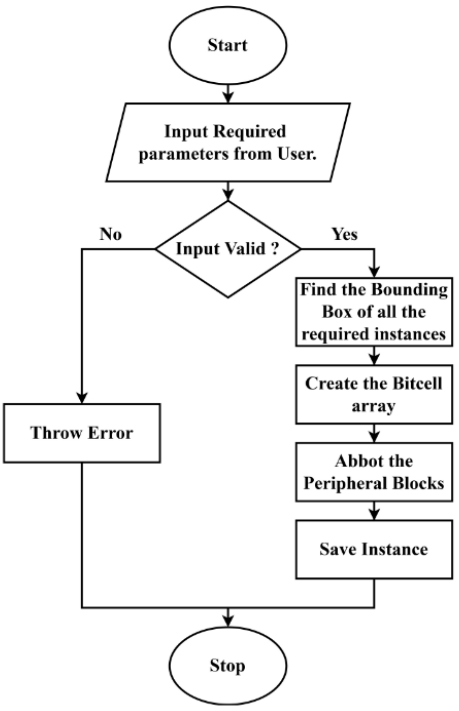


Fig.3.42 : Compiler Flowchart

1. **Start:** Upon clicking the Tile button within the layout XL, the compiler program is prompted and initiates a callback. This, in turn, triggers the launch of a graphical user interface (GUI) window, providing users with the opportunity to input the desired number of rows and columns for the memory instance. Upon entering the values, users can then proceed by selecting either the Ok or Apply button. Figure 3.43 visually illustrates the appearance of the GUI window, with an exemplary input of 8x8 showcased.

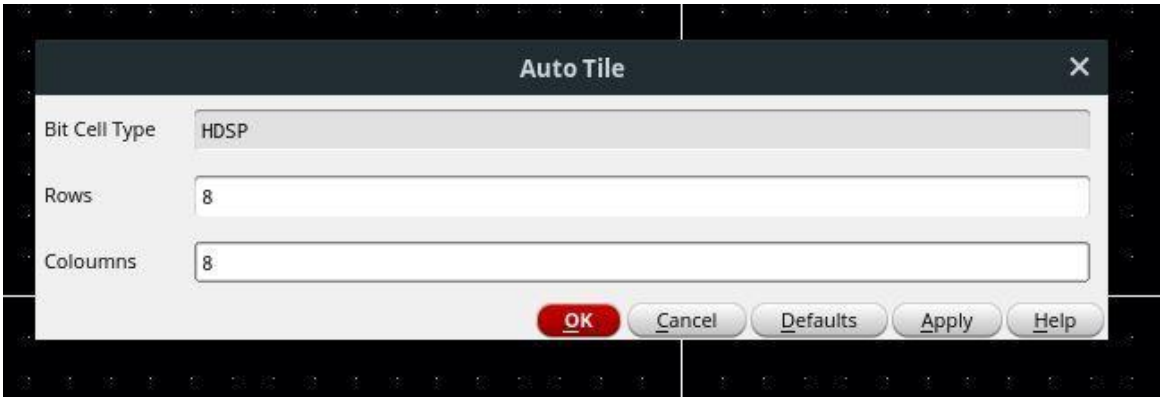


Fig.3.43 : Compiler GUI Window

2. **Input Required Parameters**: The user provides the desired number of rows and columns for the memory instance. The program validates the input by checking the following conditions:
    - The number of rows must be a multiple of 4, considering the design of the Row Decoder for 4 bit cells.
    - The number of columns must be a multiple of 2, considering the design of the Column Io for 2 bitcells.

    If either of these conditions is not met, the compiler program throws an error.

3. **Determining the Bounding Box of Required Instances:** If the user input is valid, the program dynamically calculates the bounding box for each required instance. The bounding box represents the coordinates of the border that encloses each instance, determining the perimeter. By obtaining the x and y coordinates of the endpoints of the minor diagonal, the height and width of each instance can be derived. This dynamic calculation facilitates future upgrades by allowing the flexible adjustment of bitcell types and peripheral block sizes, if needed. Figure 3.44 illustrates the program's output in the CIW, displaying the height and width of each sub-instance for the given example input of 8x8.

```
8x8_ArrayMemCellWidth=0.344000 MemCellHeight=1.920000
BottomEdgeWidth=0.172000 BottomEdgeHeight=0.192000
CornerCellWidth=0.516000 CornerCellHeigh=0.192000
LeftEdgeWidth=0.516000 LeftEdgeHeight=0.960000
MidTapBottomWidth=0.516000 MidTapBottomHeight=0.960000
MidTapWidth=0.860000 MidTapHeight=0.960000
ColIOWidth=6.856000 ColIOHeight=1.920000
ColIOWidth=6.856000 ColIOHeight=1.920000
RowDecLeftEdgeWidth=6.856000 RowDecLeftEdgeHeight=1.920000
RowDecWidth=0.688000 RowDecHeight=14.784000
ControlBlockWidth=8.385000 ControlBlockHeight=15.276000
8.064000 2.408000
```

Fig.3.44 : Calculated Height and Width by program

4. **Create the Bitcell Array:** Generate the bitcell array according to the specified number of rows and columns. To ensure a smooth transition, edge cells are strategically placed at the boundaries of the array. Figure.3.45 illustrates the resulting floor plan of the generated bitcell array for the given example input of 8x8.

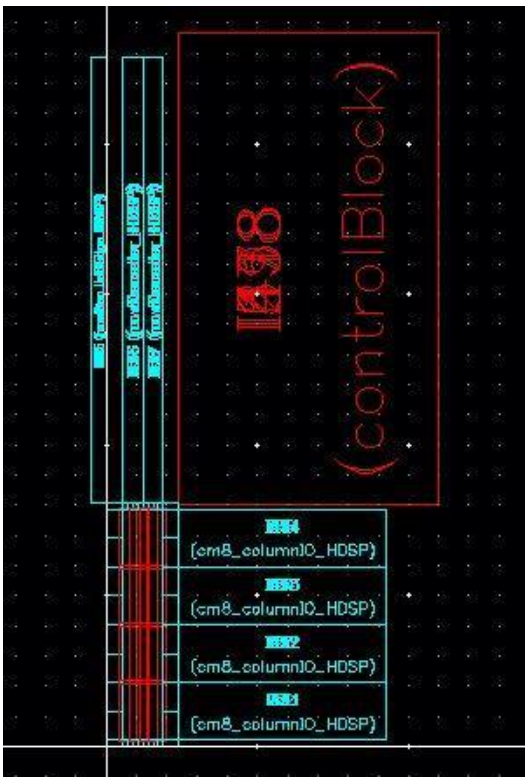


Fig:3.45 : 8x8 Full Instance Floor Plan

5. **Place the Peripheral Blocks:** Position all the necessary peripheral blocks, including Row Decoders, Column IOs, and the Control Block, at their designated locations. The floor plan of the generated memory instance, indicating the placement of these peripheral blocks for the example input of 8x8.
6. **Save Instance:** Save the newly created Memory layout instance, incorporating the user-defined parameters. The complete layout of the instance, encompassing all the components and configurations, serves as a representation of the generated layout for the example input of 8x8.

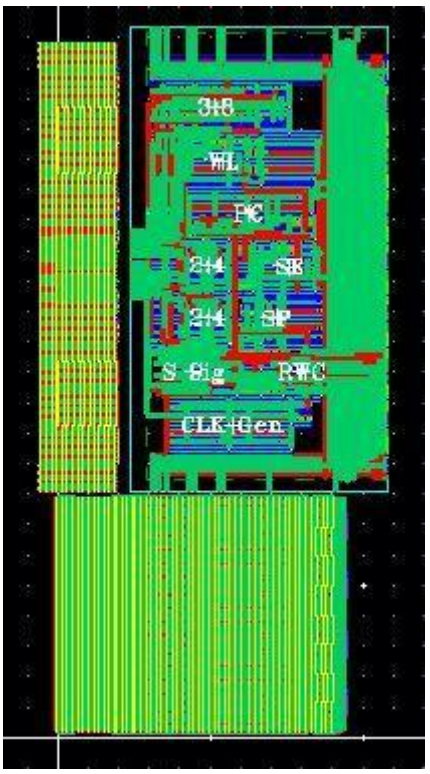


Fig.3.46 : Compiler Generated 8x8 Full Instance

Overall Conclusions of the Chapter 3:

In this chapter we have comprehensively explained the methodology and approach of our project in its entirety. The design of a single bitcell and its layout and an entire array of bit cells has been completed adhering to the layout designs and DRCs maintaining a reduced area. The design and layout of all the peripheral blocks, Column IO, Control block and RowDecoder and the individual sub-blocks that make up these blocks.

Chapter-4

Results and Discussions

4.1 Simulation Results

4.1.1 Read Current (I_{read})

To understand the behavior of the bit cell, a read current operation is conducted. The read current is measured in microamperes (uA) while varying the voltage from 0.6V to 1.2V. The relationship between read current and power supply, ranging from 0.6V to 1.2V is plotted and calculated. Similarly, the read current is examined across temperature variations ranging from -40 to 120 degrees Celsius. It reveals that the current decreases as the temperature increases, suggesting that the read operation becomes slower at higher temperatures. Therefore, the worst-case temperature is considered to be the lowest temperature.

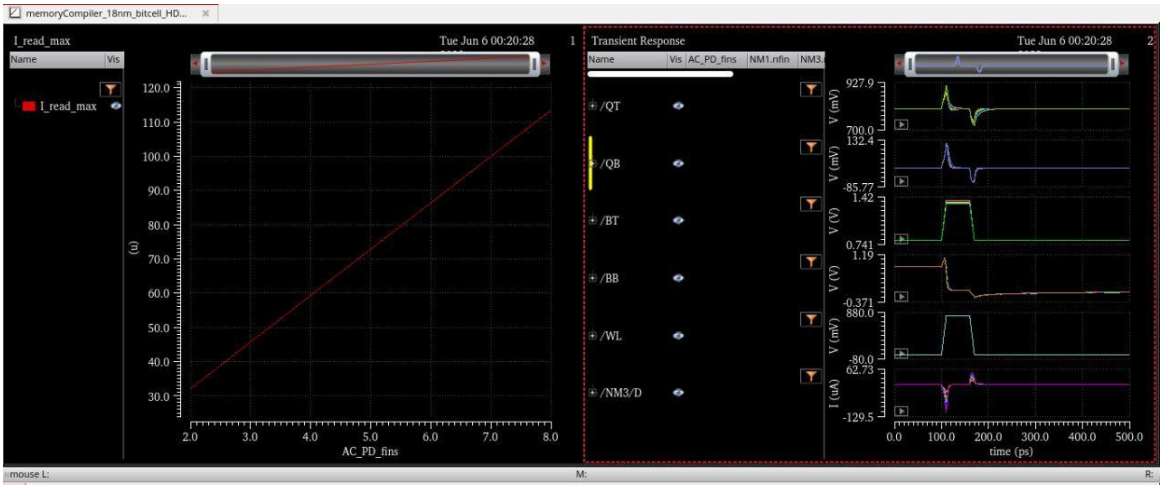


Fig 4.1 : Bit Cell Read Current Analysis

4.1.2 Leak Current (I_{leak})

The behavior of the bit cell is examined through a leak current operation. The leak current is measured in nanoamperes (nA) while varying the voltage from 0.6V to 1.2V, and the corresponding values are recorded. Leak current is also observed across a temperature range of -40 to 120 degrees Celsius. The leakage current is plotted against temperature revealing a similar trend to the voltage variation. Therefore, we can draw the same conclusion from the temperature variation analysis.

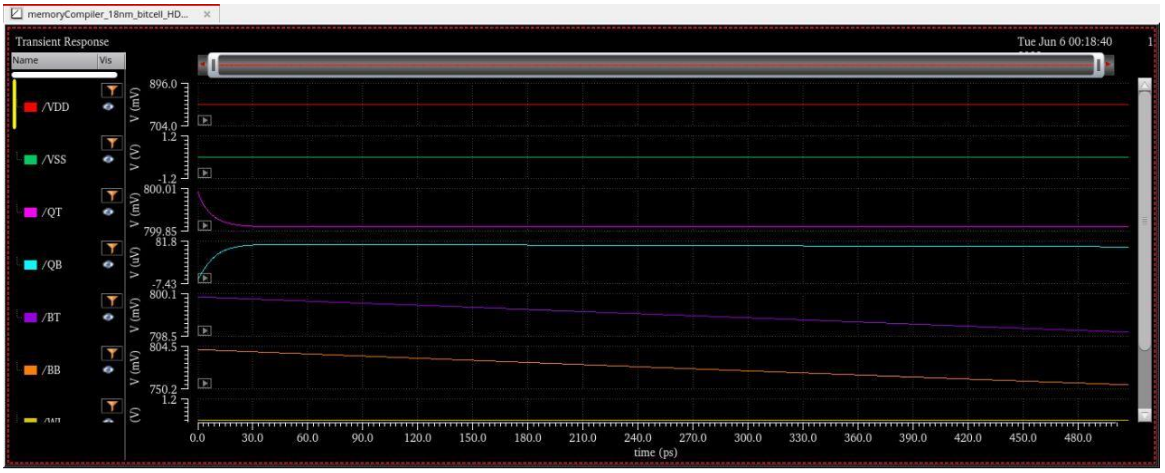


Fig 4.2 **:** Bit Cell Leak Current Analysis

4.1.3 Write Margin

The write margin indicates the ease with which a given bit cell can be written. It can be measured in millivolts with respect to both the word-line and the bit-line pairs. The write margin is measured by varying the voltage from 0.6V to 1.2V. Similarly, the write margin is examined at different temperature points ranging from -40 to 120 degrees Celsius. The same trend is observed for the BT-driven write margin and the WL-driven write margin. The minimum voltage of the WL required to flip the state of the cell is measured in millivolts. The voltage is varied from 0.6V to 1.2V.

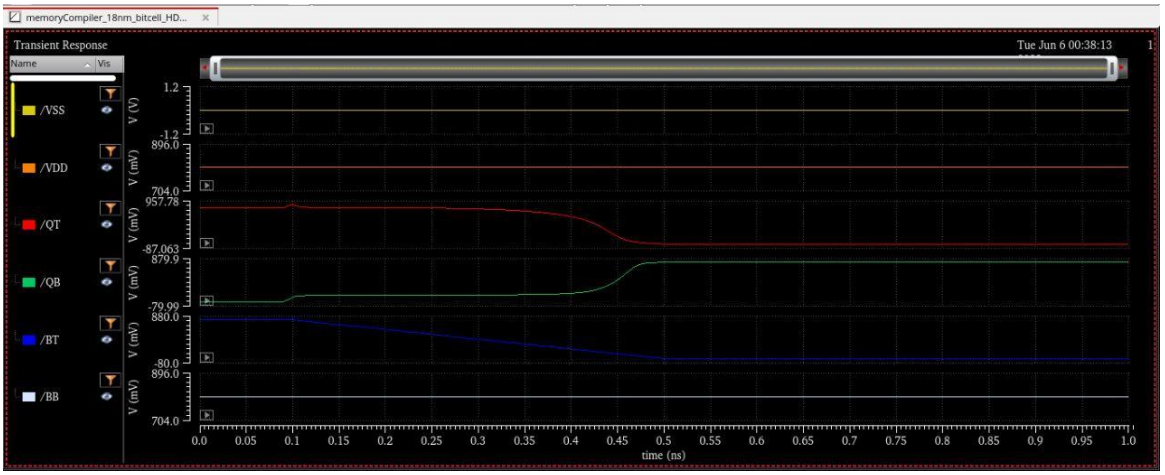


Fig 4.3 : Write Margin

4.2 Compiler Results

1. Input: Figure 4.4 illustrates an example input of 8 rows and 8 columns. This input represents the minimum requirement, where the row decoder width aligns with 4 bit cells and the column IO height aligns with 2 bit cells. Consequently, this input poses no issues and executes successfully. Figure 4.4 showcases the resulting floor plan generated for the 8x8 input, maintaining the same format as demonstrated. Additionally, Figure 4.4 exhibits the complete layout of the instance for the 8x8 test case.

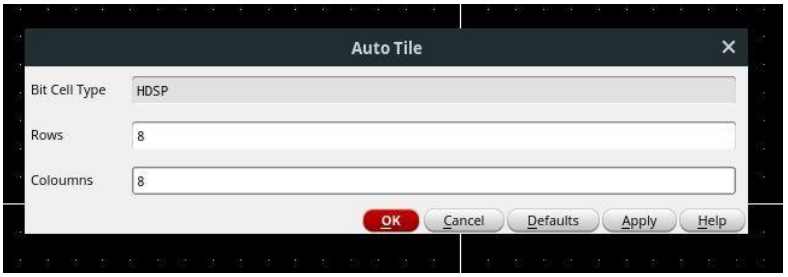


Fig 4.4 8x8 Input

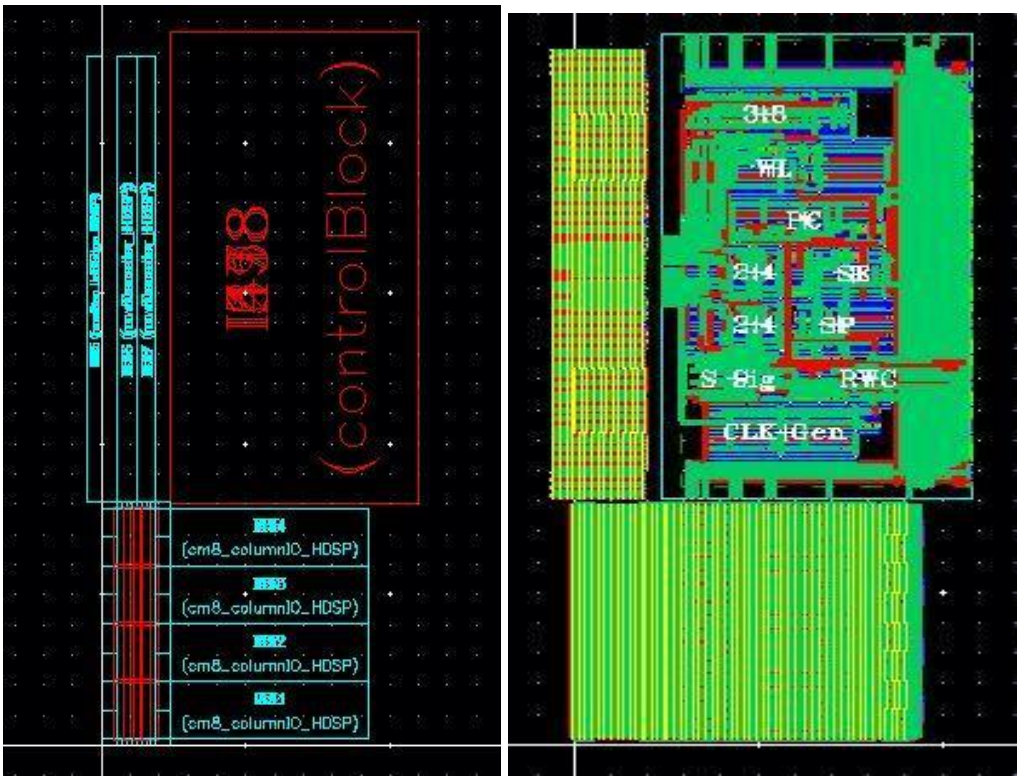


Fig 4.5 8x8 Instance and Layout

2. Input: Figure 4.6 displays an example input of 32 rows and 32 columns. This input is considered valid since the number of rows is a multiple of 4 and the

number of columns is a multiple of 2, enabling successful execution. However, it is important to note that the maximum instance that can be generated is limited to 128x128 due to the imposed constraints on row decoders.

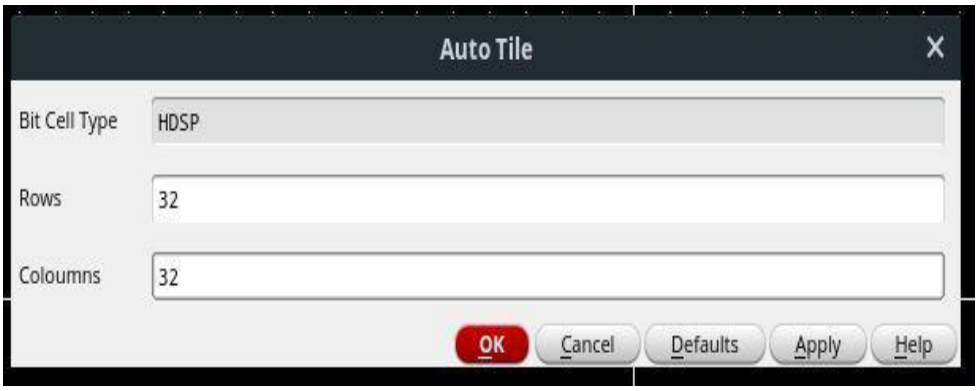


Fig 4.6 32x32 Input

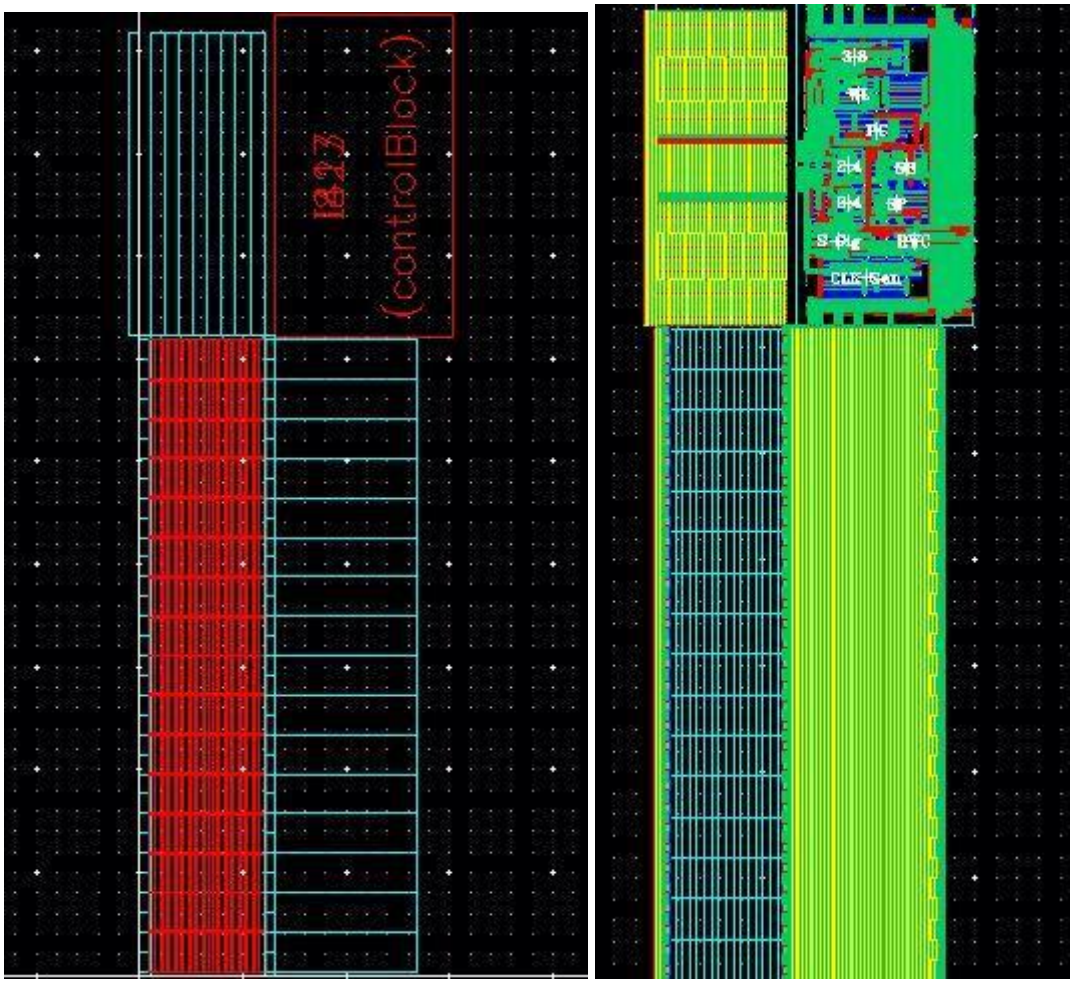


Fig.4.7 : 32x32 Instance and Layout

4.3 Applications, Advantages, Limitations

Applications

SRAMs remain an essential component in various microelectronics applications, including consumer wireless devices, high-performance server processors, multimedia systems, and System on Chip (SoC) applications. As the demand for on-chip memory increases in modern processors and SoCs, there is a need to balance performance and throughput requirements. High-performance processors prioritize operating speed and bit-cell area to achieve high-density caches while ensuring reliability. Conversely, in energy-constrained applications like sensor nodes or medical implants, the main concerns revolve around energy efficiency and reliability.

Advantages

1. The utilization of smaller 6T SRAM bit cells allows for a higher density of bit cells within a given area, resulting in a compact and densely packed memory configuration.
2. Cache memory enhances the efficiency of data retrieval from a computer's memory by serving as a temporary storage area that the CPU can quickly access. This cache memory is accessed faster by the CPU compared to DRAM. The majority of SRAM is employed in cache memory.
3. Register files, which consist of a collection of registers, are commonly defined by a CPU's instruction set architecture and are used to facilitate data transfer between memory and the processor's functional units. Modern IC-based register files typically utilize fast static RAMs with multiple ports. These RAMs differ from conventional multi ported SRAMs as they possess separate read and write ports.
4. In network switches, which are crucial for business and corporate networks, speed and performance are paramount. The choice of memory installed directly impacts the system's speed and performance, hence SRAMs are extensively utilized in network switches to ensure reliable operation.

Limitations

1. Maximum array of bit cells that can be accommodated is 128x128. It cannot exceed the array columns and rows, beyond this value.
2. We have designed a single port bitcell. So, a dual port functionality isn't possible i.e. Read and write operations can happen simultaneously, which is not possible in our design.

4.3 Conclusions of the chapter-4

The simulation results and characterization (Ileak, Iread, WLmargin) of the bitcell array is shown in this chapter. The compiler results for 8x8 and 32x32 memory instances are also shown along with their corresponding layouts.

Chapter-5

Conclusions, Future Work & Scope of Project

Conclusions

This project demonstrates a modular approach to memory block design, where the blocks are sized accordingly to facilitate easier placement and integration. The project provides detailed explanations of each operation and block functionality, accompanied by meticulously designed and optimized layouts that minimize metal usage and area wastage. Extensive testing has been conducted to validate the functionality of each block.

Testing is an essential component of any project to substantiate claims and establish credibility.It is worth noting that High Density bitcells occupy less area while performing comparably to their larger counterparts.

In the field of integrated circuit design, where the planning, revision, and testing of layouts can significantly extend the time required to build chips, memory compilers serve as invaluable tools. This project showcases such a dynamic compiler capable of designing SRAM memory layouts within seconds, thus expediting the design process and enabling reusability. The compiler has been developed using the SKILL scripting language by Cadence, a highly sought-after expertise in the VLSI industry.

While the compiler is capable of generating 128x128 single-port SRAM layouts, this constraint arises from the limitations imposed during the design of blocks such as Row Decoders and Column IO. These blocks need to drive substantial capacitive loads, and compromises in terms of area and speed have to be made to enhance their drive strengths.

Scope for future works in this domain

The project has the potential for improvement by incorporating a feature in the compiler that allows the sizing of peripheral blocks to be customized according to the requirements of the end user. This enhancement would effectively eliminate the size constraint of 128x128 and provide greater flexibility in generating instances tailored to specific needs.

Outcome of the project works

1. High Density Single Port 6T SRAM bit cell was designed along with its layout.
2. All the peripheral blocks are designed to assist in read and write operations of the bitcell.

3. A memory compiler is designed by using SKILL scripting language wherein the tiling of memory blocks is automated and we generate an entire memory instance.

References

[1] Yaqi Ma,Lijun Zhang, and Jinchen Liu. A New 6T SRAM Memory Cell Based on FINFET Process 2020 The 5th International Conference on Integrated Circuits and Microsystems DOI : 978-1-7281-8978-9/20/$31.00 © 2020 IEEE

[2] Ching-Te Chuang, Saibal Mukhopadhyay, Jae-Joon Kim, Keunwoo Kim, and Rahul Rao.High-Performance SRAM in Nanoscale CMOS: Design Challenges and Techniques IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U. S. A. DOI : 978-1-4244-1656-1/07/$25.00 © 2007 IEEE

[3] W.Kong, R.Venkatraman, R.Castagnetti, F.Dnan and S.Ramesh.High-Density and High-Performance T-SWLMfor System-on-Chip in 130 nm CMOS Technollogy2001 Symposium on VLSI Technology Digest of Technical Papers DOI :4-8911 4-012 -7/01 © 2001 IEEE

[4] Shairfe Muhammad Salahuddin and Volkan Kursun.High-Speed and Low Leakage FinFET SRAM Cell with Enhanced Read and Write Voltage Margins. 2014 International Symposium on Integrated Circuits (ISIC). DOI : 978-1-4799-4833-8/14/$31.00c 2014 IEEE

[5] Pratiksha Kulkarni, Punithra H M, Nalina H D, Preethana M,and Kajal Kumari.Low Power 6T SRAM Design using 45nm Technology. NCCDS - 2020 Conference Proceedings.Volume 8, Issue13

[6] Do Anh-Tuan, Kong Zhi-Hui, and Yeo Kiat-Seng.Hybrid-Mode SRAM Sense Amplifiers: New Approach on Transistor Sizing. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 55, NO. 10, OCTOBER 2008.DOI : 1549-7747/$25.00 © 2008 IEEE

[7] Sanjana S R, Balaji Ramakrishna S, Samiksha, Roohila Banu and Prateek Shubham. DESIGN AND PERFORMANCE ANALYSIS OF 6T SRAM CELL IN 22nm CMOS AND FINFET TECHNOLOGY NODES. 978-1-5090-6701-5/17$31.00 © 2017 IEEE

[8] B. S. Amrutur and M. A. Horowitz. A replica technique for wordline and sense control in low-power SRAMs.JSSC, 33(8):1208–1219, Aug 1998.

[9] A. Cabe, Z. Qi, W. Huang, Y. Zhang, M. Stan, and G. Rose.A flexible, technology adaptive memory generation tool.Cadence CDNLive, 2006

[10] S. M. Salahuddin, H. Jiao, V. Kursun, "Low-leakage hybrid FinFET SRAM cell with asymmetrical gate overlap/underlap bitline access transistors for enhanced read data stability," Journal of Low Power Electronics, Proceedings of the IEEEInternational Symposium Circuits and Systems, pp. 2331-2334, May 2013.[13] Z. Liu, S. A. Tawfik, and V. Kursun, ""An independent-gate FinFET SRAM cell for high data stability and enhanced integration density," Journal of Low Power Electronics, Proceedings of the IEEE International Systems on Chip (SOC) Conference, pp. 63-66, September 2007

[11] N. Gierczynski, B. Borot, N. Planes, and H. Brut, "A new combined methodology for write-margin extraction of advanced SRAM," Proceedings of the IEEE International Conference on Microelectronic Test Structures, pp. 97-100, March
2007.

[12] J. Yang et al., "Enhanced performance and SRAM stability in FinFET with reduced process steps for source/drain doping," Proceedings of the IEEE Symposium on VLSI Technology, Systems, and Applications pp. 20-21, April 2008.

[13] S. A. Tawfik and V. Kursun, "Enhanced performance and SRAM stability in FinFET with reduced process steps for source/drain doping," Proceedings of the IEEE Symposium on VLSI Technology, Systems, and ApplicationsProceedings of the IEEE International Symposium on Circuits and Systems pp. 788- 791, May 2008.

Appendix

```
procedure(bound(Libname Cellname)
        prog(()
;FIND THE CELL WIDTH and HEIGHT
cvDRC=dbOpenCellViewByType(Libname Cellname "layout" "maskLayout" "r")
  ; printf("accessing cell %s\n" cvDRC~>cellname)
foreach(shape cvDRC~>shapes
 if(shape~>purpose=="boundary" && shape~>layerName=="prBoundary" then
    bbox=shape~>bBox
    llx=car(nth(0 bbox))
    lly=cadr(nth(0 bbox))
    urx=car(nth(1 bbox))
    ury=cadr(nth(1 bbox))
    Cell_Width=urx-llx
    Cell_Height=ury-lly
       ;printf("Cell_Width is %f\n" Cell_Width)
       ;printf("Cell_Height is %f\n" Cell_Height)
   )
)
  return(list(Cell_Width Cell_Height))
)
)



procedure(createTile(memHeightVal memWidthVal)
  let(()
    sprintf(memfile "%dx%d_Array" memHeightVal*2 memWidthVal*2 )
    printf(memfile)



    tapCellLibrary = "memoryCompiler_18nm"
```

```
    Libname = "memoryCompiler_18nm"
    Cellname= "memarray_2x2_HDSP"

m=bound(Libname Cellname)
MemCellWidth=nth(0 m)
MemCellHeight=nth(1 m)
printf("MemCellWidth=%f MemCellHeight=%f\n" MemCellWidth
MemCellHeight)

m=bound(Libname "bc_bottomEdge_HDSP")
BottomEdgeWidth=nth(0 m)
BottomEdgeHeight=nth(1 m)
printf("BottomEdgeWidth=%f BottomEdgeHeight=%f\n" BottomEdgeWidth
BottomEdgeHeight)

m=bound(Libname "bc_cornerCell_HDSP")
CornerCellWidth=nth(0 m)
CornerCellHeight=nth(1 m)
printf("CornerCellWidth=%f CornerCellHeigh=%f\n" CornerCellWidth
CornerCellHeight)

m=bound(Libname "bc_leftEdge_HDSP")
LeftEdgeWidth=nth(0 m)
LeftEdgeHeight=nth(1 m)
printf("LeftEdgeWidth=%f LeftEdgeHeight=%f\n" LeftEdgeWidth
LeftEdgeHeight)

m=bound(Libname "bc_midTapBottom_HDSP")
MidTapBottomWidth=nth(0 m)
MidTapBottomHeight=nth(1 m)
printf("MidTapBottomWidth=%f MidTapBottomHeight=%f\n"
MidTapBottomWidth MidTapBottomHeight)
```

```
m=bound(Libname "bc_midTap_HDSP")
MidTapWidth=nth(0 m)
MidTapHeight=nth(1 m)
printf("MidTapWidth=%f MidTapHeight=%f\n" MidTapWidth MidTapHeight)

m=bound(Libname "cm8_columnIO_HDSP")
ColIOWidth=nth(0 m)
ColIOHeight=nth(1 m)
printf("ColIOWidth=%f ColIOHeight=%f\n" ColIOWidth ColIOHeight)

m=bound(Libname "cm8_columnIO_HDSP")
ColIOWidth=nth(0 m)
ColIOHeight=nth(1 m)
printf("ColIOWidth=%f ColIOHeight=%f\n" ColIOWidth ColIOHeight)

m=bound(Libname "rowDec_leftEdge_HDSP")
RowDecLeftEdgeWidth=nth(0 m)
RowDecLeftEdgeHeight=nth(1 m)
printf("RowDecLeftEdgeWidth=%f RowDecLeftEdgeHeight=%f\n"
RowDecLeftEdgeWidth RowDecLeftEdgeHeight)

m=bound(Libname "rowDecoder_HDSP")
RowDecWidth=nth(0 m)
RowDecHeight=nth(1 m)
printf("RowDecWidth=%f RowDecHeight=%f\n" RowDecWidth RowDecHeight)

m=bound(Libname "controlBlock")
ControlBlockWidth=nth(0 m)
ControlBlockHeight=nth(1 m)
printf("ControlBlockWidth=%f ControlBlockHeight=%f\n" ControlBlockWidth
ControlBlockHeight)
```

```
;OPEN layout_3x3 view and place the cells in below format
Libname = "memoryCompiler_18nm"
Libname=ddGetObj(Libname)
cvDRC=dbOpenCellViewByType(Libname memfile strcat(memfile "Layout") "maskLayout" "a")
MemCell=dbOpenCellViewByType(Libname Cellname "layout" "maskLayout" "r")
CornerCell=dbOpenCellViewByType(Libname "bc_cornerCell_HDSP" "layout" "maskLayout" "r")
BottomEdge=dbOpenCellViewByType(Libname "bc_bottomEdge_HDSP" "layout" "maskLayout" "r")
LeftEdge=dbOpenCellViewByType(Libname "bc_leftEdge_HDSP" " layout" "maskLayout" "r")
ColIO=dbOpenCellViewByType(Libname "cm8_columnIO_HDSP" " layout" "maskLayout" "r")
RowDecLeftEdge=dbOpenCellViewByType(Libname "rowDec_leftEdge_HDSP" "layout" "maskLayout" "r")
RowDec=dbOpenCellViewByType(Libname "rowDecoder_HDSP" "layout" "maskLayout" "r")
ControlBlock=dbOpenCellViewByType(Libname "controlBlock" "layout" "maskLayout" "r")


dbCreateInst(cvDRC CornerCell "" list(0 0) "R0" 1)
i=0
while(i<memWidthVal*2-1
dbCreateInst(cvDRC BottomEdge "" list(CornerCellWidth+i*BottomEdgeWidth 0) "R0" 1)
dbCreateInst(cvDRC BottomEdge "" list(CornerCellWidth+(i+1)*BottomEdgeWidth 0) "R0" 1)
i=i+2)
```

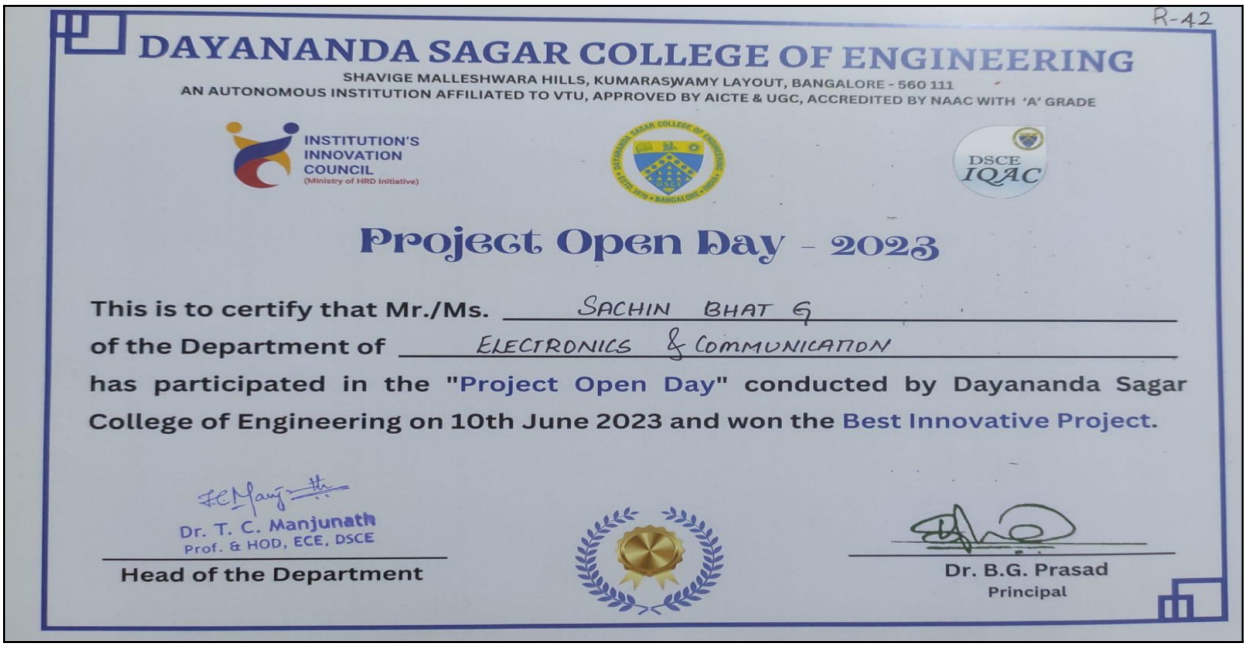
```
dbCreateInst(cvDRC CornerCell "" list(i*BottomEdgeWidth+2*CornerCellWidth
0) "MY" 1)
j=0
while(j<memHeightVal
dbCreateInst(cvDRC LeftEdge "" list(0 2*j*LeftEdgeHeight+BottomEdgeHeight)
"R0" 1)
dbCreateInst(cvDRC LeftEdge "" list(0
(2*j+2)*LeftEdgeHeight+BottomEdgeHeight) "MX" 1)


i=0
while(i<memWidthVal
dbCreateInst(cvDRC MemCell "" list(LeftEdgeWidth+i*MemCellWidth
2*j*LeftEdgeHeight+BottomEdgeHeight) "R0" 1)
i=i+1)

dbCreateInst(cvDRC LeftEdge "" list(i*MemCellWidth+2*LeftEdgeWidth
2*j*LeftEdgeHeight+BottomEdgeHeight) "MY" 1)
dbCreateInst(cvDRC LeftEdge "" list(i*MemCellWidth+2*LeftEdgeWidth
(2*j+2)*LeftEdgeHeight+BottomEdgeHeight) "R180" 1)


j=j+1)
dbCreateInst(cvDRC CornerCell "" list(0
(2*j)*LeftEdgeHeight+2*BottomEdgeHeight) "MX" 1)
i=0
while(i<memWidthVal*2-1
dbCreateInst(cvDRC BottomEdge "" list(CornerCellWidth+i*BottomEdgeWidth
(2*j)*LeftEdgeHeight+2*BottomEdgeHeight) "MX" 1)
```

```
dbCreateInst(cvDRC BottomEdge ""
list(CornerCellWidth+(i+1)*BottomEdgeWidth
(2*j)*LeftEdgeHeight+2*BottomEdgeHeight) "MX" 1)

i=i+2)
dbCreateInst(cvDRC CornerCell "" list(i*BottomEdgeWidth+2*CornerCellWidth
(2*j)*LeftEdgeHeight+2*BottomEdgeHeight) "R180" 1)
ArrayHeight = 2*CornerCellHeight+MemCellHeight*memHeightVal
ArrayWidth=2*CornerCellWidth+MemCellWidth*memWidthVal
printf("%f\t%f",ArrayHeight,ArrayWidth)

i=0
while(i<memHeightVal
dbCreateInst(cvDRC ColIO "" list(ArrayWidth CornerCellHeight+ColIOHeight*i)
"R0" 1)
i=i+1)
dbCreateInst(cvDRC RowDecLeftEdge "" list(0 ArrayHeight) "R0" 1)

i=0
while(i<memWidthVal/2
dbCreateInst(cvDRC RowDec"" list(CornerCellWidth+i*RowDecWidth
ArrayHeight) "R0" 1)
i=i+1)

dbCreateInst(cvDRC ControlBlock "" list(ArrayWidth ArrayHeight) "R0" 1)
dbSave(cvDRC)
dbClose(cvDRC)
dbClose(MemCell)
dbClose(CornerCell)

dbClose(BottomEdge)
dbClose(LeftEdge)
```

```
)
)
procedure(tile1()
let( ()
;libName = deGetCellView()~>libName
;cellName = deGetCellView()~>cellName
;If gui is not required
cellType=hiCreateStringField(?name `cellType ?prompt "Bit Cell Type" ?defValue
"HDSP" ?editable nil) ;hiCreateStringField
memHeight=hiCreateIntField(?name `memHeight ?prompt "Rows" ?defValue 2
?editable t) ; hiCreateIntField

memWidth=
hiCreateIntField(?name `memWidth ?prompt "Coloumns" ?defValue 2 ?editable
t) ; hiCreateIntField

Form = hiCreateAppForm(?name `Form ?formTitle "Auto Tile" ?fields
list(cellType memHeight memWidth) ;?dontBlock
)
; hiCreateAppForm
hiDisplayForm(Form)
cellTypeName = cellType~>value
memHeightVal= (memWidth~>value)
memWidthVal= (memHeight~>value)
;comment till here if Gui form is not required
if((mod(memHeightVal 4)!=0 || mod(memWidthVal 2)!=0 ) then
printf("Please Enter Even Length and width")
layoutDRCLVS(libName cellName)
dispString=hiCreateStringField(?name `dispString ?prompt "Message" ?defValue
"Please Enter Even Number of Rows and Coloumns"
?editable nil
) ; hiCreateStringField
```

```
Form1 = hiCreateAppForm(?name `Form1 ?formTitle "ERROR" ?fields
list(dispString) ?dontBlock
) ; hiCreateAppForm

hiDisplayForm(Form1)
else
createTile(memHeightVal/2 memWidthVal/2))
)
)
;tile1()
```

Awards, Certificates Recognitions & Photographs



R-42

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE - 560 111
AN AUTONOMOUS INSTITUTION AFFILIATED TO VTU, APPROVED BY AICTE & UGC, ACCREDITED BY NAAC WITH 'A' GRADE

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

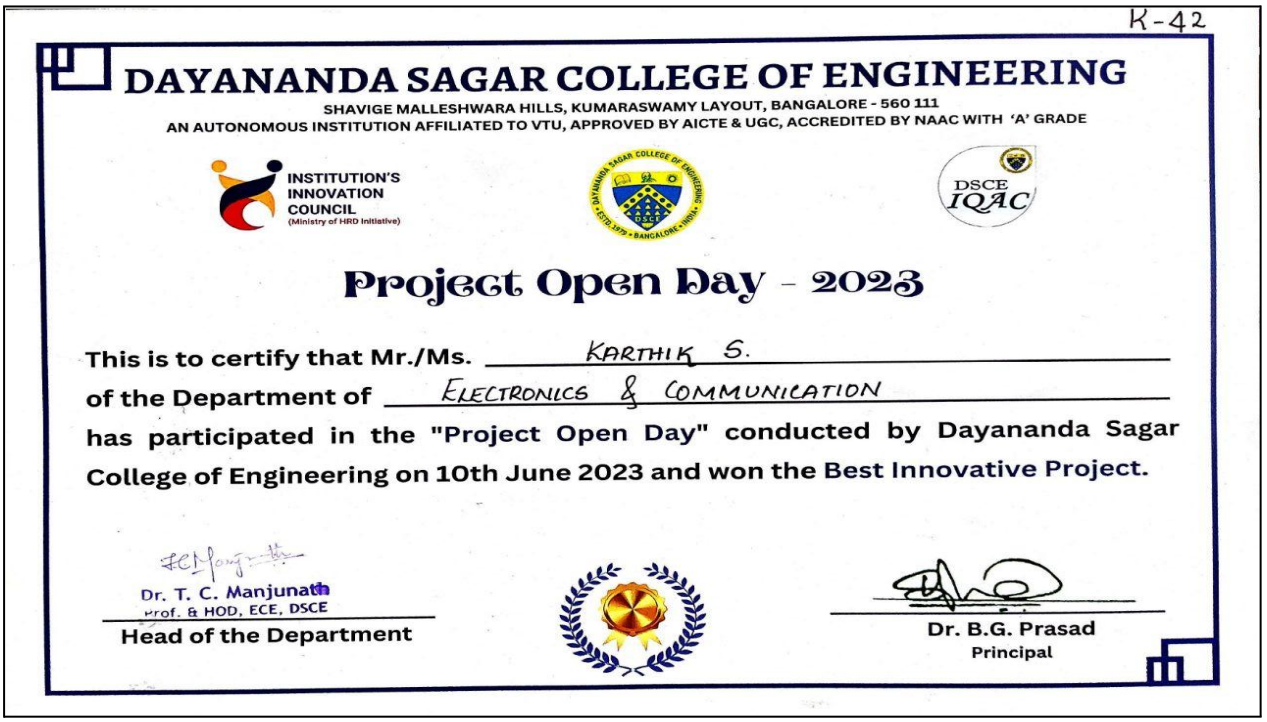
DSCE IQAC

Project Open Day - 2023

This is to certify that Mr./Ms. SACHIN BHAT G
of the Department of ELECTRONICS & COMMUNICATION
has participated in the "Project Open Day" conducted by Dayananda Sagar College of Engineering on 10th June 2023 and won the Best Innovative Project.

Dr. T. C. Manjunath
Prof. & HOD, ECE, DSCE
Head of the Department

Dr. B.G. Prasad
Principal



K-42

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE - 560 111
AN AUTONOMOUS INSTITUTION AFFILIATED TO VTU, APPROVED BY AICTE & UGC, ACCREDITED BY NAAC WITH 'A' GRADE

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

DSCE IQAC

Project Open Day - 2023

This is to certify that Mr./Ms. KARTHIK S.
of the Department of ELECTRONICS & COMMUNICATION
has participated in the "Project Open Day" conducted by Dayananda Sagar College of Engineering on 10th June 2023 and won the Best Innovative Project.

Dr. T. C. Manjunath
Prof. & HOD, ECE, DSCE
Head of the Department

Dr. B.G. Prasad
Principal

R-42

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE - 560 111
AN AUTONOMOUS INSTITUTION AFFILIATED TO VTU, APPROVED BY AICTE & UGC, ACCREDITED BY NAAC WITH 'A' GRADE

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

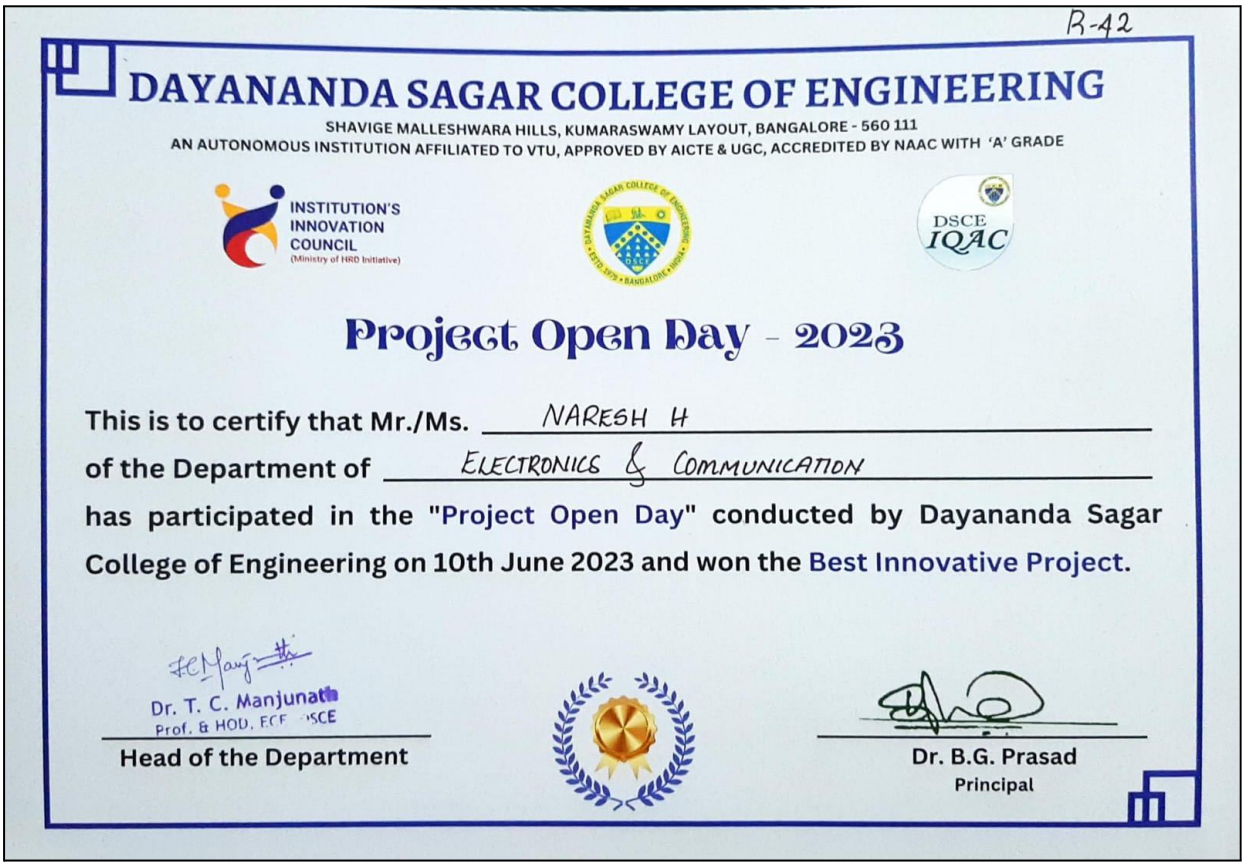
DSCE IQAC

Project Open Day - 2023

This is to certify that Mr./Ms. PRATEEK CHITNIS
of the Department of ELECTRONICS & COMMUNICATION
has participated in the "Project Open Day" conducted by Dayananda Sagar College of Engineering on 10th June 2023 and won the Best Innovative Project.

Dr. T. C. Manjunath
Prof. & HOD, ECE, DSCE
Head of the Department

Dr. B.G. Prasad
Principal



R-42

DAYANANDA SAGAR COLLEGE OF ENGINEERING

SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE - 560 111
AN AUTONOMOUS INSTITUTION AFFILIATED TO VTU, APPROVED BY AICTE & UGC, ACCREDITED BY NAAC WITH 'A' GRADE

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

DSCE IQAC

Project Open Day - 2023

This is to certify that Mr./Ms. NARESH H
of the Department of ELECTRONICS & COMMUNICATION
has participated in the "Project Open Day" conducted by Dayananda Sagar College of Engineering on 10th June 2023 and won the Best Innovative Project.

Dr. T. C. Manjunath
Prof. & HOD, ECE, DSCE
Head of the Department

Dr. B.G. Prasad
Principal


PROJECT
OPEN DAY
2023

Plagiarism test

ORIGINALITY REPORT

13%	6%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Visvesvaraya Technological University Student Paper	3%
2	Jawar Singh, Saraju P. Mohanty, Dhiraj K. Pradhan. "Robust SRAM Designs and Analysis", Springer Nature, 2013 Publication	2%
3	digitalcommons.calpoly.edu Internet Source	1%
4	Submitted to Higher Education Commission Pakistan Student Paper	1%
5	fr.scribd.com Internet Source	1%
6	mafiadoc.com Internet Source	<1%
7	Submitted to CSU, San Jose State University Student Paper	<1%
8	Saurabh, P. Srivastava. "Low power 6T-SRAM", 2012 International Conference on Emerging	<1%

	Electronics, 2012 Publication	
9	Tseng, Yuh-Kuang. "Sram", Principles and Applications in Engineering, 2003. Publication	<1 %
10	km2000.us Internet Source	<1 %
11	Shairfe Muhammad Salahuddin, Volkan Kursun. "High-speed and low-leakage FinFET SRAM cell with enhanced read and write voltage margins", 2014 International Symposium on Integrated Circuits (ISIC), 2014 Publication	<1 %
12	www.researchgate.net Internet Source	<1 %
13	Kim, Keejong, Hamid Mahmoodi, and Kaushik Roy. "A Low-Power SRAM Using Bit-Line Charge-Recycling", IEEE Journal of Solid-State Circuits, 2008. Publication	<1 %
14	www.mdpi.com Internet Source	<1 %
15	L. Zangara. "High flexibility CMOS SRAM generator using multiplan architecture", Twelfth Annual IEEE International ASIC/SOC Conference (Cat No 99TH8454) ASIC-99, 1999 Publication	<1 %

16	P. Saraza-Canflanca, R. Rodriguez, M. Nafria, F.V. Fernandez et al. "Design Considerations of an SRAM Array for the Statistical Validation of Time-Dependent Variability Models", 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2018 Publication	<1 %
17	aml.ece.iisc.ac.in Internet Source	<1 %
18	hdl.handle.net Internet Source	<1 %
19	ruor.uottawa.ca Internet Source	<1 %
20	ece.uwaterloo.ca Internet Source	<1 %
21	Yaqi Ma, Lijun Zhang, Jinchen Liu. "A New 6T SRAM Memory Cell Based on FINFET Process", 2020 IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM), 2020 Publication	<1 %
22	www.freepatentsonline.com Internet Source	<1 %

23	Haozhe Ma, Zhihui Hu, Daquan Yu. "A Low-Loss Bandpass Filter with Stacked Double-Layer Structure using Glass-based Integrated Passive Device Technology", IEEE Electron Device Letters, 2023 Publication	<1 %
24	www.coursehero.com Internet Source	<1 %
25	studylib.net Internet Source	<1 %
26	Alberto Bosio, Luigi Dilillo, Patrick Girard, Serge Pravossoudovitch, Arnaud Virazel. "Advanced Test Methods for SRAMs", Springer Science and Business Media LLC, 2010 Publication	<1 %
27	Mohd Nazri Ismail, Abdullah Mohd Zin. "Development of Emulation Network Analyzer Tool for Computer Network Planning", Bulletin of Electrical Engineering and Informatics, 2016 Publication	<1 %
28	Prakhar Sharma, Kirti Gupta, Neeta Pandey. " A Single-Ended 10-Transistors Static Random Access Memory Bit-Cell With Improved Write-Margins at Low Voltages and High I -I Ratio ", 2019 International Conference on Smart	<1 %

Systems and Inventive Technology (ICSSIT), 2019
Publication

29	lup.lub.lu.se Internet Source	<1%
30	Ashish Sachdeva, Vinay Tomar. "Design of Low Power Half Select Free 10-T Static Random Access Memory cell", Journal of Circuits, Systems and Computers, 2020 Publication	<1%
31	H. Hidaka, Y. Matsuda, K. Fujishama. "A divided/shared bit-line sensing scheme for ULSI DRAM cores", IEEE Journal of Solid-State Circuits, 1991 Publication	<1%
32	S. Muhammad Salahuddin, Hailong Jiao, V. Kursun. "A novel 6T SRAM cell with asymmetrically gate underlap engineered FinFETs for enhanced read data stability and write ability", International Symposium on Quality Electronic Design (ISQED), 2013 Publication	<1%
33	Silva López Héctor Hugo. "Low-voltage power supply distribution system development in a scientific space mission", TESIUNAM, 2014 Publication	<1%
34	documents.mx Internet Source	<1%

35	ethesis.nitrkl.ac.in Internet Source	<1%
36	ijireeice.com Internet Source	<1%
37	Luigi Dilillo. "Analysis and Test of Resistive-Open Defects in SRAM Pre-Charge Circuits", Journal of Electronic Testing, 11/01/2007 Publication	<1%
38	Md Imranur Rahman, Tasnim Bashar, Satyen Biswas. "Performance evaluation and read stability enhancement of SRAM bit-cell in 16nm CMOS", 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016 Publication	<1%
39	Mohamed H. Abu-Rahma, Mohab Anis. "Nanometer Variation-Tolerant SRAM", Springer Science and Business Media LLC, 2013 Publication	<1%
40	escholarship.org Internet Source	<1%
41	github.com Internet Source	<1%
42	ndl.ethernet.edu.et Internet Source	<1%

43	ntnuopen.ntnu.no Internet Source	<1%
44	uwspace.uwaterloo.ca Internet Source	<1%
45	www.ijert.org Internet Source	<1%
46	www.ijser.org Internet Source	<1%
47	www.science.gov Internet Source	<1%
48	"Data stability and write ability enhancement techniques for FinFET SRAM circuits", Nano-CMOS and Post-CMOS Electronics Circuits and Design, 2016. Publication	<1%
49	Sherif A. Tawfik, Volkan Kursun. "Portfolio of FinFET memories: Innovative techniques for an emerging technology", 2008 International SoC Design Conference, 2008 Publication	<1%
50	W. Kong, R. Venkatraman, R. Castagnetti, F. Duan, S. Ramesh. "High-density and high-performance 6T-SRAM for system-on-chip in 130 nm CMOS technology", 2001 Symposium on VLSI Technology. Digest of Technical Papers (IEEE Cat. No.01 CH37184), 2001	<1%


Publication
51
"Design of 13T SRAM Bitcell in 22nm Technology using Fin FET for Space Applications", International Journal of Recent Technology and Engineering, 2019
<1%
Publication
Exclude quotes On
Exclude matches Off
Exclude bibliography On

CO-PO Mapping Justification Sheets

Course Objective

1. Examine various components of a project plan: Literature survey, modern tools, methodologies and execution of the plan.
2. Learn modern tools and techniques to develop the proposed model.
3. Provide solutions to the societal and environmental needs
4. Ability to design and develop sustainable solutions to the problems
5. To gain the ability of efficient time and finance management
6. Improve interpersonal skills, enhance team work, written and oral communication skills.

Course Outcomes:

Students will be able to:

COs	PROJECT WORK
CO1	Demonstrate proficient knowledge on the concepts involved.
CO2	Identify the problem and propose the possible solution through literature survey.
CO3	Design and develop engineering solutions to complex problems through systematic approach.
CO4	Build prototype/simulation for the proposed solution and articulate the work
CO5	Provide sustainable solutions considering societal needs by exhibiting individual and cooperative learning.
CO6	Complete the proclaimed work within stipulated time span with financial constraints

Title of the Project: "Generation and analysis of 6T SRAM memory instance- bank1, cm2, HDSP".

CO-PO Mapping:

Mapping of Course Outcomes to Program Outcomes and Program Specific Outcomes:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2		2	3		2				2		2	2
CO2		3	2	3		2	1							
CO3	3		3		2			2		2		2	2	2
CO4	2			2	2					2		2	2	3
CO5			1			2	3	1	2	1				
CO6									2		3			
CO AVG	2.33	2.5	2	2.33	2.33	2	2	1.5	2	1.66	2.5	2	2	2.33

*High=3, Medium=2, Low=3

Justification

CO1	Utilized multiple software and hardware tools effectively
CO2	Identified problems through literature survey by many papers and analyzed the auto check auto and improved farming technique.
CO3	Designed a memory instance with appropriate software applications
CO4	Developed the designed system to work efficiently in real time
CO5	Developed the skill program to automate layout design following engineering ethics and guidelines.
CO6	Completed the project according to designed timeline with teamwork and management skills

PO and PSO Mapping for Project - 2022-23

Batch Number: R42

USN	Name
1DS19EC087	NARESH H
1DS19EC103	PRATEEK CHITNIS
1DS19EC116	SACHIN BHAT G
1DS20EC414	KARTHIK S

Guide Name: DR. NINU RACHEL PHILLIP

Justification for PO & PSO mapping for Project

Project Title		
PO	Levels 3/2/1	Justification
PO1	3	Apply the knowledge of hardware and software in designing using current practices,and standards
PO2	2	We reviewed the research literature and able to analyze the mechanism and implement it
PO3	3	Designed and developed a prototype working model
PO4	3	Reviewed many articles and applied different algorithms. Finally able to provide best possible algorithm.
PO5	3	Cadence virtuoso and SKILL ide is used for implementing of our project and better working of it.
PO6	3	Developed model is helped for automating layout design
PO7	3	Prototype model was not harmful to the society.
PO8	3	Ethics was strictly followed and followed the norms of engineering
PO9	3	team completing the work assigned individually in the stipulated time.

PO10	2	Communication was successfully done.
PO11	3	Financial support was provided by sponsors
PO12	3	Further trying to apply for patent
PSO1	3	Designed and developed and integrate electronics circuits and systems using current practices and standards
PSO2	3	Apply knowledge of hardware and software in designing embedded and communication systems.

Signature of Guide

Budget Estimation

Batch Number: R-42

USN	Name
1DS19EC087	NARESH H
1DS19EC103	PRATEEK CHITNIS
1DS19EC116	SACHIN BHAT G
1DS20EC114	KARTHIK S

Guide Name: DR. NINU RACHEL PHILIP

Sl. No.	Particulars	Estimated Cost in Rs.
1	a) Materials / Consumables (Cadence software with skill ide.)	4000
2	b) Labor	0.00
3	c) Travel (Transportation Costs to acquire products and parts)	1100.00
4	e) Miscellaneous	0.00
Total		5100

Signature of Guide