HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

GRADUATION PROJECT

DESIGN, SIMULATE AND LAYOUT

6T STATIC RANDOM-ACCESS MEMORY 64 BITS

TON HOANG UYEN NHI

Student ID: **19161041**

Major: **ELECTRONIC AND TELECOMMUNICATION ENGINEERING**
Advisor: **TRUONG QUANG PHUC, M.Eng.**

Ho Chi Minh City, June 2023

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY FOR HIGH QUALITY TRAINING

GRADUATION PROJECT

DESIGN, SIMULATE AND LAYOUT

6T STATIC RANDOM-ACCESS MEMORY 64 BITS

TON HOANG UYEN NHI

Student ID: **19161041**

Major: **ELECTRONIC AND TELECOMMUNICATION ENGINEERING**
Advisor: **TRUONG QUANG PHUC, M.Eng.**

Ho Chi Minh City, June 2023

THE SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom– Happiness

Ho Chi Minh City, June 21, 2023

GRADUATION PROJECT ASSIGNMENT

Student name: TON HOANG UYEN NHI Student ID: 19161041

Major: Electronic And Telecommunication Engineering Class: 19161CLA1

Advisor: Truong Quang Phuc, M.Eng. Phone number: 0765970743

Date of assignment: _____ Date of submission: _____

1. Project title: Design, Simulate, And Layout 6T Static Random-Access Memory 64 Bits

2. Initial materials provided by the advisor: _____

3. Content of the project: _____

4. Final product: _____

CHAIR OF THE PROGRAM

(Sign with full name)

ADVISOR

(Sign with full name)

Trương Quang Phúc

THE SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom– Happiness

Ho Chi Minh City, June 21, 2023

ADVISOR'S EVALUATION SHEET

Student name: TON HOANG UYEN NHI Student ID: 19161041

Major: Electronic And Telecommunication Engineering

Project title: Design, Simulate, And Layout 6T Static Random-Access Memory 64 Bits

Advisor: Truong Quang Phuc, M.Eng.

EVALUATION

1. Content of the project:

.....

2. Strengths:

.....

3. Weaknesses:

.....

4. Approval for oral defense? *(Approved or denied)*

Approved.

5. Overall evaluation: (Excellent, Good, Fair, Poor)

Excellent.

Ho Chi Minh City, June 21, 2023

ADVISOR

(Sign with full name)

Trương Quang Phúc

THE SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom– Happiness

Ho Chi Minh City, June __, 2023

PRE-DEFENSE EVALUATION SHEET

Student name: TON HOANG UYEN NHI Student ID: 19161041

Major: Electronic And Telecommunication Engineering

Project title: Design, Simulate And Layout 6T Static Random-Access Memory 64 Bits

Name of Reviewer: .....

EVALUATION

1. Content and workload of the project

.....

.....

2. Strengths:

.....

.....

3. Weaknesses:

.....

.....

4. Approval for oral defense? *(Approved or denied)*

.....

5. Reviewer questions for project valuation

.....

.....

.....

.....

6. Mark:…….(*in words*:.....)

Ho Chi Minh City, (month day, year)

REVIEWER

(Sign with full name)

ACKNOWLEDGEMENT

Firstly, I would like to express my appreciation to the University of Technology and Education, as well as the Faculty for High Quality Training, for providing all the support needed in the process of making this project.

I also want to give a sincere thanks to M.Eng. Truong Quang Phuc, who is my primary instructor in this capstone project, for his dedication. He offered such specific guidance along with on-time feedback to improve this project from the beginning. He played a big role in maintaining the direction of this study and ensuring its progress.

Furthermore, I desire to express my deep gratitude to all the lecturers at the University of Technology and Education for providing the fundamental background knowledge. The knowledge gained is used in this project as a solid foundation for studying and writing this paper.

With such limited time and resources for this project, there might be faults and mismatches. I am looking forward to receiving feedback from all the reviewers to further develop and improve my future work.

Sincerely,

Ton Hoang Uyen Nhi

ABSTRACT

This project aims at implementing and analyzing 64-bit SRAM with 6T memory cells. The SRAM design proposed in the paper is for the purpose of studying the architecture, operations, and characteristics of an SRAM memory. This study uses the Cadence Virtuoso ADE tool to implement and evaluate the design in the TSMC 90nm technology library.

The design features 6T memory arrays, which are widely used in nowadays SRAM designs, as well as peripherals for controlling the arrays, including Address Decoder, Precharge, Write Driver, Sense Amplifier, Read/Write Pass and I/O Latches. The components will be implemented using Virtuoso Schematic Editor. Also, their layout will be drawn using Virtuoso Layout Suite XL. Finally, the components will be assembled to form a complete SRAM memory.

The performance of the proposed design is verified through Virtuoso ADE L. The read and write processes are verified using the timing diagram. The timing diagram represents various test read/write cases. The stability of the 6T memory cells is guaranteed by performing SNM analysis. From the layout aspect, every component is fully DRC and LVS-verified using Assura.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

SRAM	Static Random-Access Memory
CPU	Central Processing Unit
SoCs	System on Chips
DRAM	Dynamic Random-Access Memory
6T, 8T	6 transistors, 8 transistors
SNM	Signal to Noise Margin
PVT	Process Voltage Temperature
BJT	Bipolar Junction Transistor
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
RAM	Random-Access Memory
CAM	Content-addressable memory
MRAM	Magnetoresistive random-access memory
ROM	Read Only Memory
MROM	Masked Read Only Memory
PROM	Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
CMOS	Complementary metal–oxide–semiconductor
TSMC	Taiwan Semiconductor Manufacturing Company
UMC	United Microelectronics Corporation
BL	Bitline
WL	Wordline

CHAPTER 1: INTRODUCTION

1.1. OVERVIEW

The current era of digitalization allows for every daily task being done online, from paying bills to automatically driving a car. All of this is made possible by the tremendous development of computer technology. In essence, a computing system is a system that can be programmed to perform a series of logical or mathematical operations automatically. In a computing system, the processor, often known as the Central Processing Unit (CPU), is where arithmetic or logical operations are carried out. A CPU stores and processes digital data in different memory location. Hence, the performance of a computing system relies on the bandwidth and latency of the memory. There are two type of computer memory: volatile and non-volatile. Volatile memory stores data and operation which the computer need in real time, and erases them when the power gets cut off. On the other hand, the memory that stored in non-volatile memory statically remains even when computer's power is switched off.

As electronic devices have scaled down to a limited size, System on Chips (SoCs), brings another deal to the competitive market with its advantages such as better performance, lower power consumption and smaller in semiconductor die area. Yet, every System-on-a-Chip (SoC) suffers from a performance bottleneck since the main memory and the CPU operate at different speeds. Using cache memory, which is built up of Static Random-Access Memory (SRAM) cells, is one approach to lessen the bottleneck. Furthermore, SRAM has the highest integration densities consuming most of the transistors of a typical SoCs [1]. Hence, designing a low latency, low power SRAM memory plays a major role in developing SoCs, or the computing systems in general.

In comparison with the other memory type which is Dynamic Random-Access Memory (DRAM), SRAM is faster in read/write operation with lower power consumption. However, SRAM takes a lot of space due to the high number of transistors that it contains. The size of the SRAM cell is being reduced using scaling over the past three decades [2]. The scaling techniques also have their drawbacks such as higher leakage and bigger delay, which then cause more power consumption and slower operation. Realizing the importance of designing a proper SRAM, this work focuses on building a low power, high speed 64 bits 6-transistors (6T) SRAM based on TSMC gpdk090nm technology. The designed SRAM's layout will be design and present using Virtuoso Cadence. Furthermore, simulations will be carried out to demonstrate the performance of the memory regarding power dissipation, signal-to-noise margin (SNM) and read/write delay.

1.2. RELATED WORKS

In [3], the paper analyzes the efficiency of different SRAM designs which use TSMC 90nm technology. 6T and 8T SRAM have been simulated in term of SNM, read/write operation, power consumption and delay with different process-voltage-temperature (PVT) condition. The paper successively compared the advantages and disadvantages of each SRAM structure. SRAM architecture, 6T/8T cell and their operation are also mentioned and explained in this paper. However, a thorough view has not been carried out for a deeper understanding.

In [4], the research offers a design for a 90nm CMOS UMC technology, 32 KB synchronous SRAM module which uses 6T memory cell structure. This SRAM design also combines row and column redundancy, pre-decoder, column multiplexer and self-timing circuitry. The simulations for read/write operation, cycle time and power consumption are also carried out for all process corners. A drawback in this research is the lack of detailed circuitry of each instance and their operation.

In [5], the paper proposed three design methods used to achieve high performance, low power synchronous single port 1024x32 SRAM Using 28NM Technology. The three designs methods include folding with column multiplexer 4 to 1, pre-decoding and tracking technique. Using Synopsys, HSPICE, Cadence and Custom Sim as supported tools, the authors carried out simulations in term of power consumption, leakage current and memory cycle time. The research also presents and read/write operation of the memory in different PVT variations. Same as [4], this paper focuses on analyzing the operation of proposed SRAM. Therefore, a clear view of the instance's circuit was not presented. In addition, in spite of analysis on set up and hold time of data, set up and hold time of addresses were not mentioned.

In [6], the author works on designing and analyzing 6T, 128x128, low voltage SRAM with applying self-time dummy replica technique using TSMC 180nm technology. The circuitry of all SRAM's instances is fully presented. The simulation for read/write operation, self-time operation and sense amplifier operation are also carries out. In addition, the access time of the proposed memory is also analyzed with high and low data case.

In [7], the research presents design and implementation of 8K bits, low power SRAM in 180nm technology. The work focuses on performing low power design techniques including sub-array memory, multi-stage decoding and dynamic NOR decoder. The SNM of memory cell, the operation of peripherals and read/write operations are also analyzed through simulations and waveforms.

In [8], the paper proposes two topologies for decreasing leakage in SRAM. The proposed topologies are presented and simulated for comparison with the conventional 6T SRAM cell. The detailed SRAM block structure, cell circuitry and its operation are

provided in this research. Simulations for leakage currents in SRAM cell are performed for each topology including 6T SRAM cell. Yet, the paper only focuses on analyze leakage currents of the memory cells. The other important parameters are not mentioned and evaluated such as delay time, SNM of the operation, etc.

1.3. OBJECTIVES

- Construct 8x8 SRAM memory array using 6T memory cells, a 3-to-8 address decoder and the peripherals of the memory. Then, assemble all the components into a complete 8x8 SRAM instance.
- Simulate the SRAM memory and verify its read and write operation through timing diagram. Also, analyze the Read and Write Noise Margin of the 6T memory cell.
- Calculate the power consumption in both the active and inactive stages as well as the delay times for various read and write operations.
- Layout the components of the SRAM memory and assemble them together into a complete layout file. Verify each part of the layout with DRC and LVS rule and optimize its size and symmetry.

1.4. METHODOLOGY

In order to conduct research and put the topic into practice, data is collected and analyzed. Specifically, I start off with collecting and synthesizing theory from different sources to build the draft model. Then, necessary parameters will be calculated for customizing and enhancing the designs. With experimental approach, multiple simulations will be carried out to perform data analysis and evaluation. Lastly, using statistical analysis method to ascertain if a predictor variable and an outcome variable have a statistically significant relationship. The research method is achieved through the process of solving following problems:

- Problem 1: Study the schematic and operation of 6T SRAM
- Problem 2: Build the model of SRAM arrays and all the peripheries which control them then proceed simulations to assure the functional accuracy of memory's read and write operations.
- Problem 3: Carry out calculation and evaluation from previous simulations to make adjustments for the design for better delay time and power consumption in read and write operation.
- Problem 4: Study layout design rule and appropriate layout method to design and present the layout for proposed SRAM design.
- Problem 5: Compound the components into a finished product and evaluate the final achieved result then give comments and propose future work.

1.5. REPORT'S LAYOUT

This research is presented in 5 chapters including:

- **Chapter 1 – Introduction:** Give an overview of the current technology state and accentuate the importance of memory in the development of computing systems. Point out the objectives of the research, the current related works inside and outside of the country and the approaching method for this project.
- **Chapter 2 – Literature Review:** Present the structure of a complete SRAM including its 6T memory cell and all array controlling peripheries. Give explanations of the memory read and write operation. Show basic understanding of setup and hold time of data and address in an SRAM design. Also, power consumption and SNM measuring method is explained.
- **Chapter 3 – System Design:** Present the block diagram and detailed schematics of SRAM components. Compound all components into a complete SRAM design and predict its operation for later comparison with the results provided in next chapter.
- **Chapter 4 – Results:** Carry out simulations for all the components and the final design then perform calculation to achieve desired results for evaluation. Present the layout design result and sum up characteristic parameters of the proposed design.
- **Chapter 5 – Conclusion and future work:** Give comments on the achieved results and make conclusion on the accomplishments from the research. Present the limitation of the topic and propose future works for enhancing the proposed design.

CHAPTER 2: LITERATURE REVIEW

2.1. MEMORY IN COMPUTING SYSTEM

Every computing system is based on receiving raw data, processing entered data and returning results. These operations are made possible by the instructions and data which are already stored in computer's memory. The operation of a system is desired to be fast and accurate, therefore, memory should be designed high speed and stable. In term of primary memory, it uses semiconductor technology so it can also be referred to as semiconductor memory.

The first semiconductor memory was made based on bipolar junction transistor (BJT) in the 1960s. However, BJT is too large and costly for an integrated application such as memory. Hence, this approach is considered impractical. Later, the metal–oxide–semiconductor field-effect transistor (MOSFET) was invented in 1959, by Mohamed M. Atalla and Dawon Kahng at Bell Labs. This invention is the first step of the development of current memory technology. MOS transistors-based memory gradually replaced magnetic memory and became the most used memory tech in the early 1970s.

The main memory of computers is classified into two types: Random-Access Memory (RAM) and Read Only Memory (ROM). The first difference between these two types of memory is the access mode of the memory. RAM can be accessed for reading data from and writing data to its memory cells while ROM only allows read operation. The read and write operations of RAM can be done in two separate clock cycle or in a single cycle depending on number of ports it has. Another major distinction between them is the ability of keeping data when the power supply is cut off. While ROM is capable of storing data without system's supply, data stored in RAM will be lost as computer shutting down. These two major types of memory are further categorized. Table 1 represents some of the popular memory types which are currently used or studied.

- SRAM (Static Random-Access Memory): SRAM is volatile memory which uses flip-flops to store data bits, each flip-flop store one bit of data. Because multiple of transistors are used to made one SRAM memory cell, it is more area consuming and expensive than DRAM. SRAM keeps computer's data as long as power supply is applied and is often used for cache memory.

- DRAM (Dynamic Random-Access Memory): DRAM is volatile memory which uses transistors and capacitors to store data. Each DRAM memory cell usually consists of one transistor and one capacitor. The bitcell's value is stored by the states of that capacitor with charged is '1' and discharged is '0'. However, the charge on capacitors will gradually leak away after milliseconds. Hence, DRAM needs refreshing circuitry which rewrites the data back to capacitors. DRAM's characteristics make it slower than

SRAM. Yet, DRAM is cheaper and higher density, it is used as computer's main memory.

- CAM (Content-Addressable Memory): CAM is a special computer memory which can compare the input data with data stored in its cells and return the address of matched cell. It is also called associative memory and often used in network routing or database applications. There are two types of CAM: BCAM (binary CAM) and TCAM (ternary CAM). BCAM has the ability of searching binary bits ('0' and '1') in its storage while TCAM can search for a third element which is "don't care" case ('X').

- MRAM (Magneto-resistive Random-Access Memory): Unlike other types of RAMs, MRAM is a non-volatile memory. MRAM stores data bits in the form of magnetic states. An MRAM cell consists of ferromagnetic plates separated by an insulating layer which create a magnetic tunnel junction. One of the plates is fixed in term of magnetic orientation while the other is free. The state of MRAM cell is determined by the orientation of the free plate. If the free plate has the same direction as the fixed plate, the junction has low resistance and the cell is storing logical '0'. If two plates are not parallel, then the junction is high resistive and logical '1' is being stored. The orientation of the free plate can be changed by applying magnetic field or polarized currents. MRAM is high density, low power, fast and non-volatile. Hence, it is a very potential type of memory.

- MROM (Masked Read Only Memory): Masked ROM is the first type of non-volatile memory. MROM is pre-programmed during fabrication. Its data and instructions are physically encoded in the circuit. Hence, MROM data is not erasable and changeable. MROM is inexpensive.

- PROM (Programmable Read Only Memory): PROM (Programmable ROM): PROM is non-volatile, digital memory which can be programmed once after manufacture. Unlike ROM which is programmed during fabrication, PROM is manufactured blank and then programmed afterwards. After programming PROM, its data cannot be changed. Because of unerasable characteristic, PROM is used for storing low level programs.

- EPROM (Erasable Programmable Read Only Memory): EPROM, which can also be called EROM, is a type of PROM. However, its data can be erased and pre-programmed a limited amount of time by ultra-violet light. In the programming process, an electrical charge is implanted in the floating fate region of a transistor. EPROM data is erased by using ultra-violet light to dissipate the charge trapped inside. EPROM are used in microcontroller before the invention of EEPROMs.

- EEPROM (Electrically Erasable Programmable Read Only Memory): EEPROM, or E^2PROM, is nonvolatile ROM which can be erased and re-programmed electrically. EEPROMs are integrated in microcontrollers and can be re-programmed up to ten

thousand times. EEPROM data can be changed in one-byte scale rather than an entire chip, which make it flexible in exchange of speed. Hence, it is usually used in small storage applications.

***Table 2. 1**. Types of Memory*

RAM	ROM
Static Random-Access Memory (SRAM)	Masked Read Only Memory (MROM)
Dynamic Random-Access Memory (DRAM)	Programmable Read Only Memory (PROM)
Content-Addressable Memory (CAM)	Erasable Programmable Read Only Memory (EPROM)
Magneto-resistive Random-Access Memory (MRAM)	Electrically Erasable Programmable Read Only Memory (EEPROM)

2.2. SRAM STRUCTURE

2.2.1. SRAM array

SRAM array consists of multiple memory cells, particularly 64 memory cells in case of 8x8 SRAM. These cells are 6T SRAM cells, which include 2 pull up transistors, 2 pull down transistors and 2 pass gate transistors. The cells are connected into multiple rows and columns controlled by wordlines (WLs) and bitlines (BLs). Every particular cell has a unique address and can be accessed by the WL signal. For each cell, the pass gate transistors are switched by WL signal while the bitline (BL) and bitline bar (BLB) are in charge of writing and reading operations in that cell. The detailed operations will be explained further in Section 2.3.

2.2.2. Address Decoder

As mentioned in the previous section, the cells are identified by unique addresses. In case of a large amount of address, decoders are used for simplification. With address decoder, fewer bits are needed for defining a memory cell. There are two type of address decoder: row decoder and column decoder. In some special design, there may be pre-decoder for low power application.

2.2.3. Periphery Control Block

A complete SRAM design consists of different peripheries which are used to control the memory cell’s read/write operations. The control block responsible for triggering the right signal at the right time. Because timing signals plays a massive role

in an SRAM operation, the main purpose of this block is generating internal clock signals from external clock pulse. Also, other input signals such as address, memory enable, write enable, etc. are also managed in the control block.

2.2.4. Pre-charge, Sense Amplifier and Write Driver

- *Pre-charge:*

The pre-charge circuit's function is manipulating the charging and discharging process of the bitlines. In the beginning of every read/write operation, the bitlines must be pre-charged to VDD. When read pass or write pass signals trigger, the pre-charge will be switched off and one of the bitline will be dis-charged or pulled down depending on whether it is a read or write operation.

- *Sense Amplifier:*

The read operation of SRAM is based on the voltage difference between the bitline and bitline bar. For power efficiency, that difference is desired to be as small as possible. Sense amplifier is made for recognizing such a small amount of voltage. Sense amplifier associating with output data latch points out the cell's value.

- *Write Driver:*

On the other hand, the write operation of SRAM is based on pulling down bitline to ground level. However, this process needs a strong driver which is capable of pulling the bitline down fast. The write driver associating with input data latch supports write operation.

2.2.5. I/O Control Block

I/O block contains circuits which help directing the route of data. As new data is put in for write operation, the input data will be latched before new clock cycle. As data is read from memory cells, it is latched out before the sense amplifier switched off so that the output is stable. These processes are performed in the I/O control block.

2.3. SRAM READ/WRITE OPERATION

2.3.1. 6T SRAM Cell Operation

As mentioned in the previous section, SRAM stores data in form of binary bits and each bit is stored in a unit called "cell". Conventional SRAM cells consist of 6 transistors

forming cross-coupled CMOS inverters and a pair of pass gates M5 – M6 as can be seen in Figure 2.1. These two pass gates can be switched by WL signal.

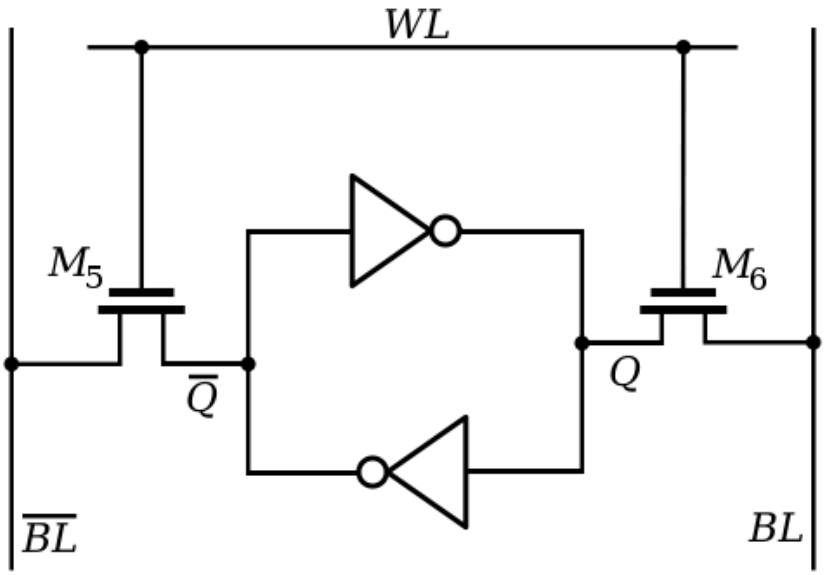


Figure 2. 1. *SRAM Cell Inverter Loop [9]*

In case of WL = ‘1’, the cell is selected for read or write operation. The cell which connected to WL = ‘0’ is unselected and keep its previous value.

Figure 2.2 illustrates the detailed schematic of 6T SRAM cell with transistors M1 – M6. M1, M3 are pull down transistors while M2, M4 are pull up ones of 2 inverters. Data is stored in Q and Q_bar which are input and output of the inverters and have opposite value. When the WL is selected, the BLs are connected to Q and Q_bar. Depend on read or write operation, the value stored in Q and Q_bar is read onto BLs or BLs write value back to Q and Q_bar.

2.3.2. Read Operation

In read operation, the value stored in Q and Q_bar are read onto the BLs. Assume that in this case, the bit stored in this cell is ‘0’ as in Figure 2.3. Figure 2.3 shows the waveform of read operation. Before WL is triggered, the BLs are pre-charged to VDD by pre-charge circuit. The WLs triggers and turn on the pass gates M5 – M6. With Q is ‘0’, there is a voltage difference between BL and Q. The active transistors in this case are M2 and M3 because Q = ‘0’ and Q_bar = ‘1’. The BL_bar remains high while BL discharges through M3. This process creates a voltage difference between the BLs. Sense amplifier circuit is responsible for recognizing the difference and amplify it. In

the opposite case in which Q stores ‘1’, the process is the same however the discharge process will happen on BL_bar.

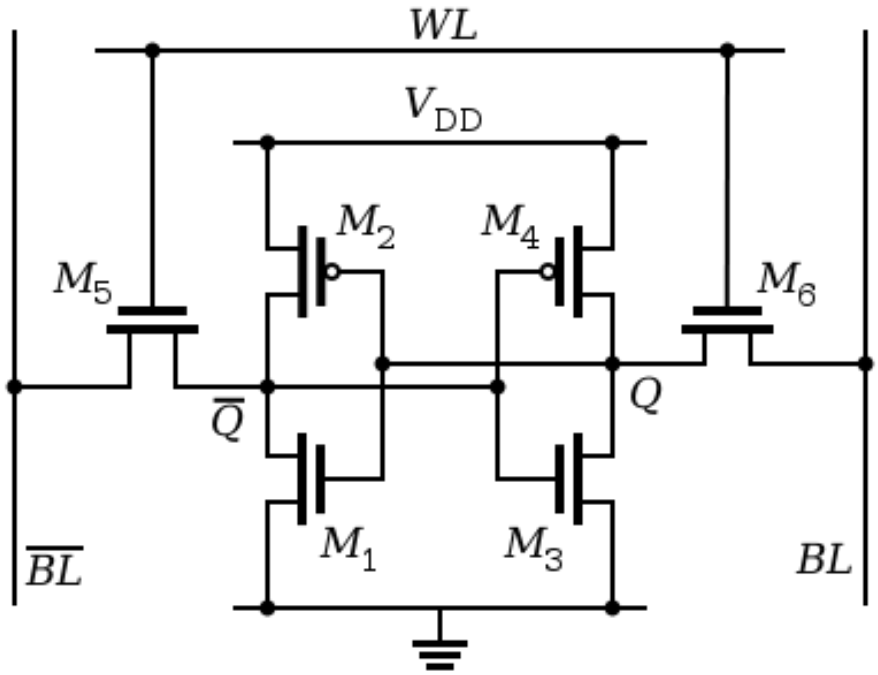


Figure 2. 2. *6T SRAM Cell Schematic [10]*

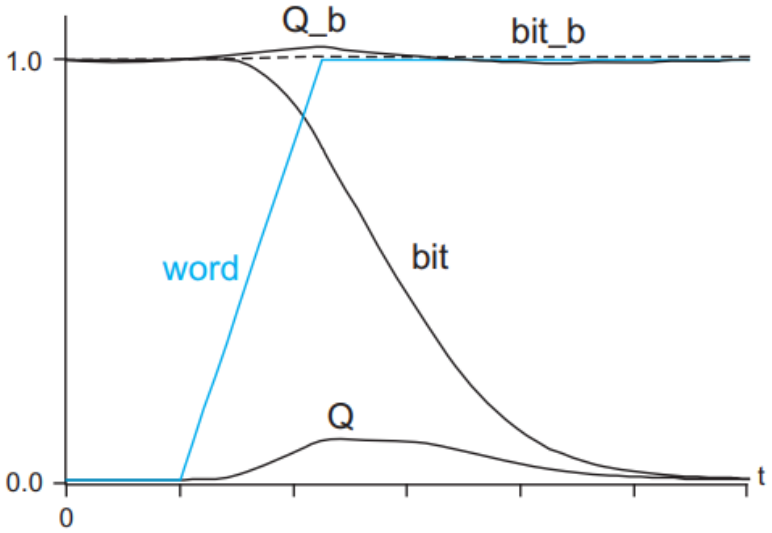


Figure 2. 3. *Read Operation Waveform [11]*

2.3.3. Write Operation

To write data to a cell, one of the bitline is used to pull the cell to VSS. Assume that, in this case, ‘1’ is written into a cell, which initially is ‘0’. Same as read operation, the bitlines are pre-charged to VDD before triggering wordline. For Q = ‘0’ and Q_bar = ‘1’ initially, M2 and M3 are on, connecting Q_bar to VDD and Q to VSS. New data, in this case is bit ‘1’, is loaded in thanks to data latch and write driver. The value of this new data decides which bitline will be tied to VSS. To write bit ‘1’ into the cell, BL_bar has to be tied to ground. Once the WL is triggered and the cell is selected, BLs are

connected to Q and Q_bar. BL_bar pulls Q_bar down to VSS and switches M4 on, connecting Q to VDD. Figure 2.4 represents the write operation waveform of a 6T SRAM cell. As can be seen from the figure, after wordline is triggered, BL_bar pulls Q_bar to VSS and flips Q value from '0' to '1'. The slight increase in Q voltage at the beginning is BL discharging through M3. However, the driver pulls BL_bar down is much stronger, this discharging process will not affect the operation.

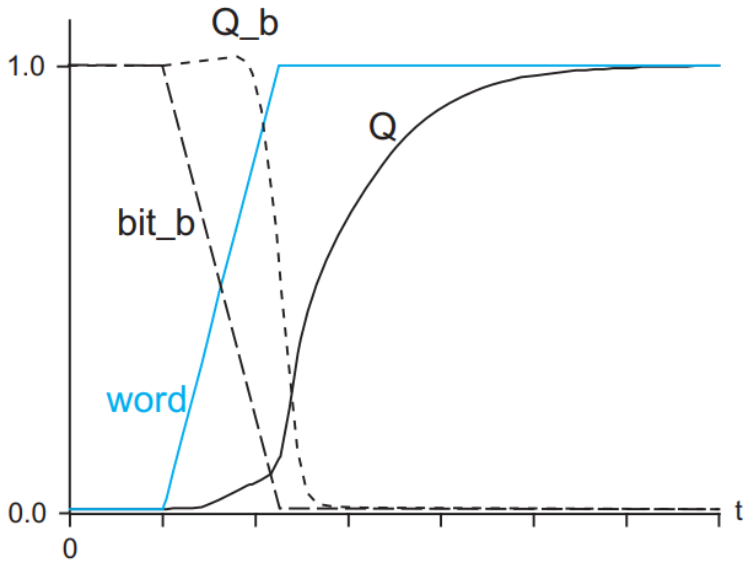


***Figure 2. 4**. Write Operation Waveform [11]*

2.4. TIMING AND POWER CONSUMPTION OF SRAM

2.4.1. Setup, Hold Time of Data and Address

SRAM operation depends on periphery such as sense amplifier, pre-charge, etc. for controlling the read/write operations towards the bitcells. Hence, timing the delay of the devices is considered to be necessary for a good design yield. The critical timing process is set up and hold timing for data input and address. In term of read operation, set up and hold time of address needs to be tuned properly for reading the correct data at the exact position. Meanwhile, in write operation, set up and hold time of both data and address needs to be checked. It is because not only address needs to be stable for writing at the exact bitcells, but also data needs to be valid and ready for pulling down the correct bitline.

SRAM works on an external clock, with each cycle is either read or write operation. On high pulse of clock, read/write operation is performed. On low pulse of clock, signals are reset, data and address are unlatched for inputting new data and address, preparing for new cycle. Set up time is the minimum amount of time a signal needs to reach a logic node before clock triggers. It can be seen as the preparing time for the signal. Set up time of address/data is the delay between address/data and clock rise at address/data latch. Hold time is the minimum amount of time a signal needs to

keep its value after clock triggering for a stable output. Hold time of address or data is the amount of time it takes for latching address/data. Once address/data is latched, the input address/data can be changed. The latched address/data is used for internal processing, hence, the change in address/data input will not affect the circuit.

Figure 2.5 describes the setup and hold time of address and data according to clock. $t_{dsu}, t_{dh}, t_{asu}, t_{ah}$ is data set up time, data hold time, address set up time, address hold time respectively. t_{dsu} is measured from data stable state (data rises to 80%VDD or falls to 20%VDD) to clock rises 50% VDD. t_{dh} is measured from clock rises 50% VDD to data already changed state (data rises to 80%VDD or falls to 20%VDD). t_{asu} is measured from address stable state (address rises to 80%VDD or falls to 20%VDD) to clock rises 50% VDD. t_{ah} is measured from clock rises 50% VDD to address already changed state (address rises to 80%VDD or falls to 20%VDD).

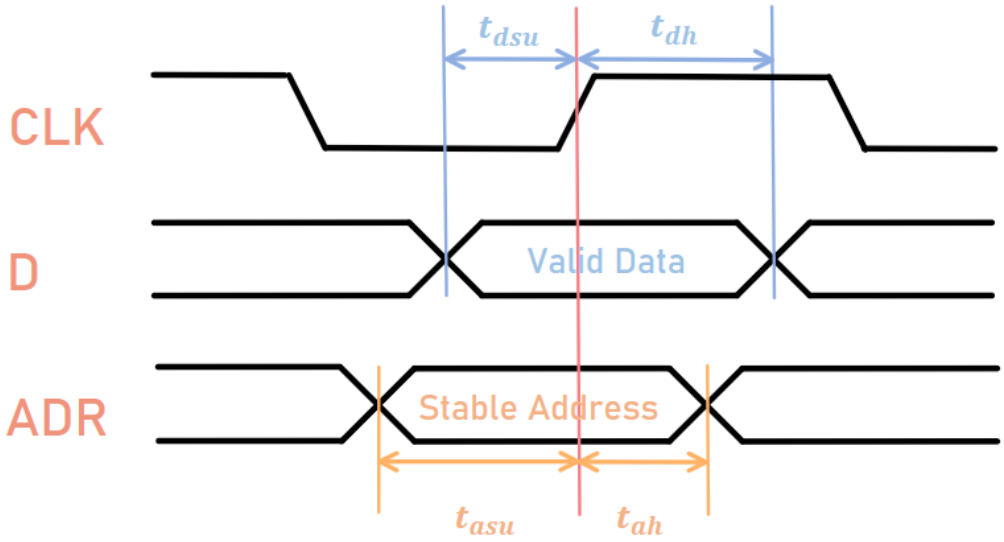


Figure 2. 5. *Data and Address Timing Waveform*

2.4.2. Read and Write Cycle Time

As mentioned in the above section, SRAM works on an external clock, with each cycle is either read or write operation. On high pulse of clock, read/write operation is performed. On low pulse of clock, signals are reset, data and address are unlatched for inputting new data and address, preparing for new cycle. For accessing the memory, a signal called ME, which is short for memory enable, has to be triggered. The read or write operation is decided by WE, which is short for write enable. When WE = 1, the write operation is performed and when WE = 0, then the read operation is performed.

Figure 2.6 represents a cycle of read operation. In low pulse width phase of the previous cycle, the signals are being prepared before triggering a new cycle (in this case is a read cycle). ME is switched on for memory access, this triggering is just for presentation, it can be switched on from the beginning of turning on the memory and keeping its value since. Meanwhile, address and WE is set up to have a stable state before clock rises. It can be seen from the figure, the output Q has the value of Q(n-1)

before clock rising and gradually change to Q(n) after some delay after clock rises. Here, Q(n-1) represents for the output data of previous read cycle and Q(n) represents for current read operation's output data. The changing process including multiple internal signals connecting periphery which will be deeply discussed later in chapter 3. After finishing reading the value of selected word, the address and WE signals are free for changing its state. Turning off WE right after successfully read selected data helps with saving power. This timing process can be done by adding self-timing circuit which also will be discussed in chapter 3.

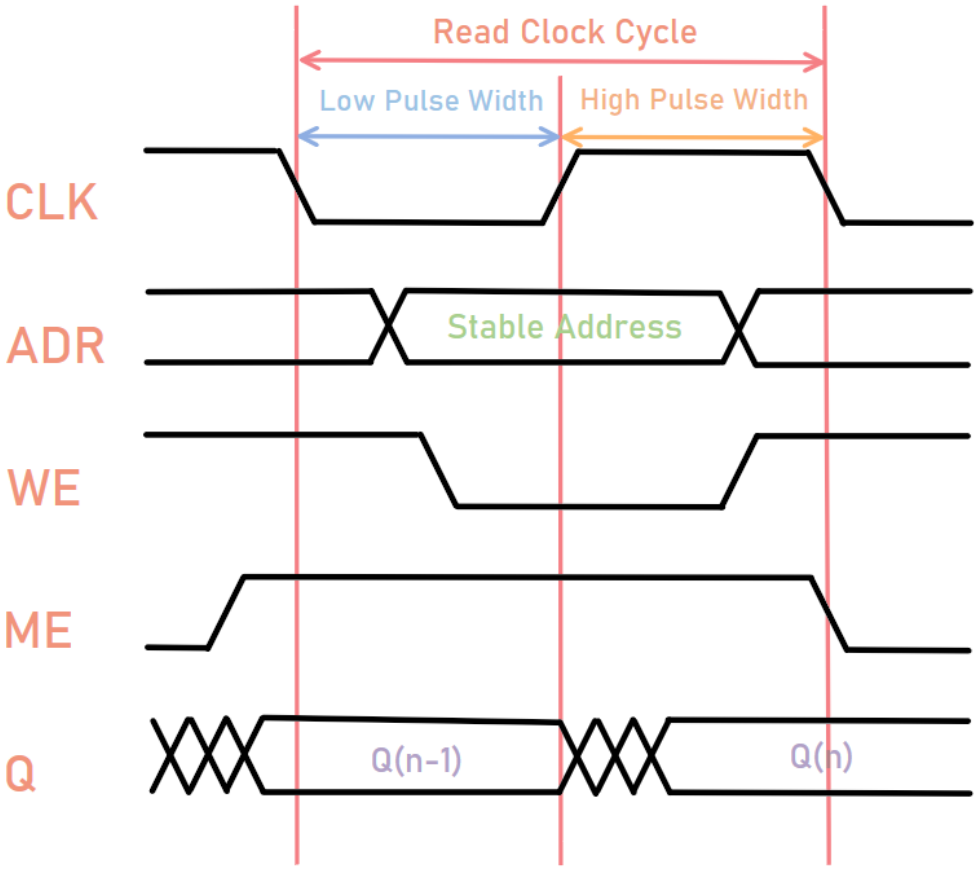


Figure 2. 6. *Read Cycle Time*

Figure 2.7 describes an SRAM cycle of write operation. Same as in read operation, the low phase of clock of previous operation will reset and change signal's value preparing for next cycle. Before clock rising which triggers a write operation, new address and new data should be stable, ready for latching. ME is also brought high to enable the memory and WE is pulled up to allow write function. All these signals will be latched in as clock triggers, which then latch address and data. The latched signals are used to write new data into bitcells through some processes of SRAM periphery. Unlike in read operation, write process does not change output value, only the value of selected word is changed. Hence, when clock rises, Q still keeps its content from previous read cycle, which is Q(n-1).

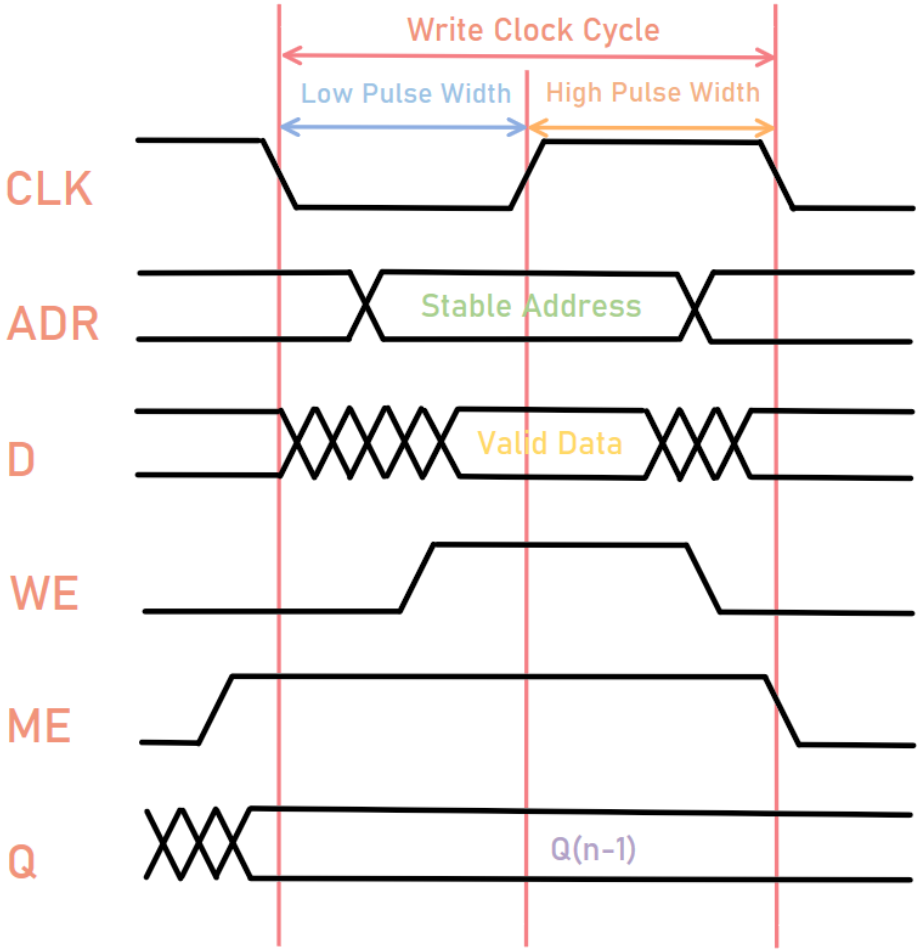


Figure 2. 7. *Write Cycle Time*

2.4.3. Power Consumption and Signal Noise Margin (SNM)

2.4.3.1. Power Consumption

SRAM memory contains billions of transistors which switches their states frequently. The power consumption in SRAM design mostly comes from the switching of transistors in read/write operation and the leakage current in its unactive state. There are many methods had been proposed to decrease the power dissipation of SRAM memory such as power gating, self-timing circuit, dual-rail memory, etc., in which, additional circuits must be inserted to the memory. These circuits have bad impact on the memory performance and its area. Hence, the power consumption optimization in an SRAM design should be significantly studied.

Instantaneous power is the energy that a circuit consumes at a certain point of time. It can be calculated by the equation (2.1) given below:

$$P(t) = I(t)V(t) \qquad (2.\ 1)$$

However, to determine the power dissipation of a circuit, the power should be observed over a period of time. Because the power consumption at different point of time is significantly different depending on the operation at that point. The average power consumption over a period of time is considered the power dissipation of a circuit.

Average power over period time T can be calculated by the given equation (2.2):

$$P(t) = \frac{I}{T}\int_0^T P(t)dt \qquad (2.\ 2)$$

The total power consumption of a circuit comes from the power consumed in its active state and the power consumed in its idle state as given in the equation (2.3). Active power is the power consumed while the chip is doing useful work [11]. As mentioned above, the dynamic power of an SRAM memory is mostly contributed by the switching of transistors in read/write operations. Meanwhile, the power consumed in idle state is the unwanted leakage which comes from the leak currents through unactive transistors. In nanometer processes with low threshold voltages and thin gate oxides, leakage can account for as much as a third of total active power [11].

$$P_{total} = P_{dynamic} + P_{static} \qquad (2.\ 3)$$

2.4.3.2. Signal Noise Margin (SNM)

In SRAM operations, there are two noise margins which should be studied carefully to ensure the bitcell stability. The two margins are the read margin and write margin. In read operation, the bitlines are precharged to VDD before wordline being triggered. The bitlines have the trend to pull up the node which stores value '0' through the access transistors. Hence, the pulldown transistors should be designed to be stronger than the access transistors. The read margin of a bitcell is the amount of noise can be inserted into that cell without changing its state. Meanwhile, in write operation, one of the bitlines is pulled down the GND level before wordline being triggered. That bitline will then pull the node which stores bit '1' down to write '0' into that node. In order to pull down the node, the access transistors must be stronger than the pull up transistors. The write margin is the amount of noise can be inserted to the bitcell without creating two same stable states in one bitcell.

To determine these margins, the feedback between the cross-coupled inverters is broken and replaced by a varied voltage source. With the read margin, the wordline and bitlines is tied to VDD and the voltage source is varied from GND to VDD. With the write margin, the wordline is tied to VDD, one of the bitlines is also tied to VDD while the other bitline is pulled down to GND and the voltage source is varied from GND to VDD. In both cases, the voltage transfer characteristics on two halves of the cross-coupled inverters are plotted. The margins are the largest squares fitted between the two plotted lines.

CHAPTER 3: SYSTEM DESIGN

3.1. SYSTEM REQUIREMENTS

The proposed system is designed to meet the below requirements:

- The system is based on the TSMC GPDK90 nanometer technology. This technology is chosen for designing because of its availability and the integration feature.

- This system allows read and write operations to be performed to memory cells. By applying the specific address to the decoder of this system, the desired memory cells will be pointed to.

- The operating frequency of this system can be up to 0.8MHz. The read or write operations can be selected by applying the write enable input. With WE = 1, the desired write data can be inserted in by data input. On the other hand, when WE = 0, the read operation is selected for the cycle. The desired read value can be observed at the data output at the end of the cycle.

- This SRAM design is expected to be small in size and stable in operation. For area efficiency, the 6T bitcells are used for a better integration. These bitcells should also be sizing well for a good SNM.

- The final design must pass all the simulation test cases, which are write '1', write '0', read '1' and read '0'. After verifying the design through simulations, the layout of the SRAM instance is constructed. The layout should be small in size while assuring the identity with the schematic. Also, layout of this SRAM design is expected to satisfy the requirements of metal rule, spacing rule and symmetric rule.

3.2. SYSTEM DESIGN

3.2.1 System Block Diagram

Figure 3.1 describes the block diagram of a full instance 8x8 SRAM memory. The design is divided into 5 main blocks including Control Block, Address Decoder Block, Array Block, Periphery Block and IO Block.

- The Control Block contains circuitry for creating internal clock pulses which are used for controlling the racing of signals. Also, latch circuitry for latching input signals is placed in this block. Lastly, it contains circuits which generate periphery control signals for read/write operation.

- The Address Decoder Block is a 3 to 8 decoder which uses address as input. The output of this block is 8-bit signal which is used to select the desired wordline for read or write operation.

- The Array Block includes 64 6T SRAM bitcells. These bitcells are placed into 8 rows and 8 columns with specific address for each row. 8 bitcells in the same column are merged together to form an 8x1 array for easy access.

- The Periphery Block consists of Precharge, Sense Amplifier, Write Driver and Read/Write Pass Gating circuit. This block is in charge of the read and write function of the memory.

- The IO Block controls the data input and data output of the memory using the Data Input Latch and Data Out Latch circuit placed inside the block.

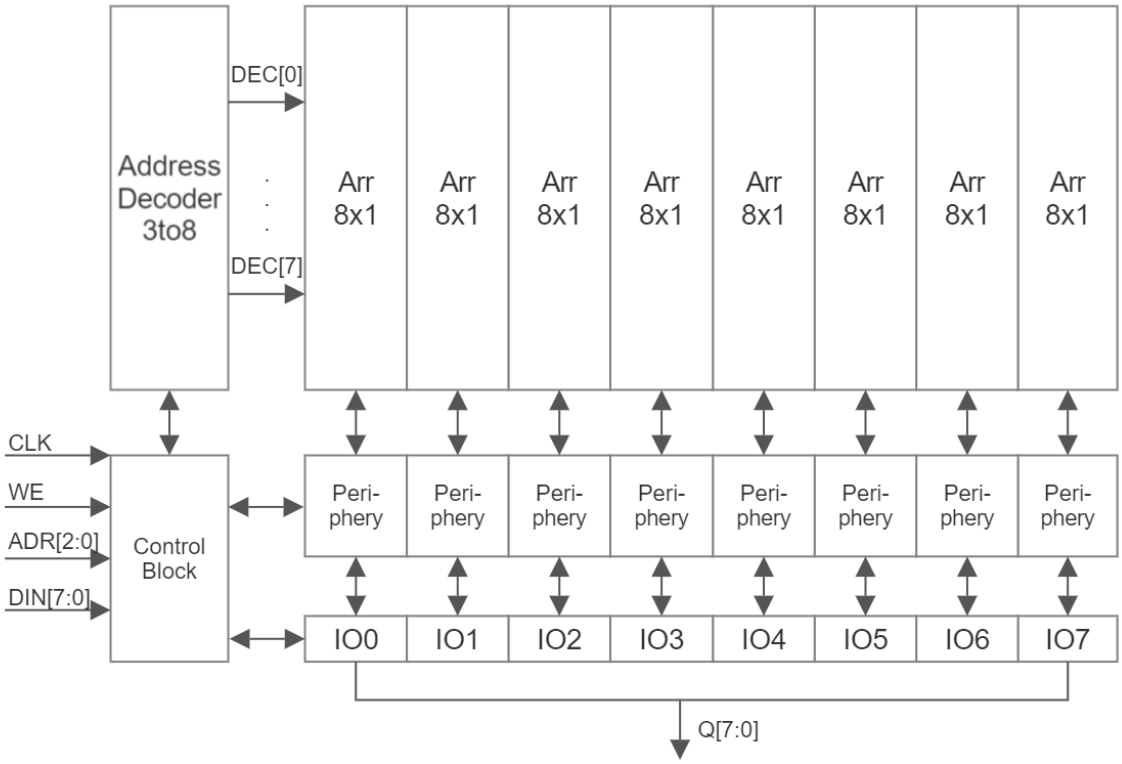


Figure 3. 1. *8x8 SRAM Instance Block Diagram*

3.2.2 System Operation

For an SRAM design, there are two main operations which are reading data contained in selected bitcells and writing data to selected bitcells. The bitcells selection is made possible by the address decoder. Meanwhile, the read/write selection is controlled by input signal WE. WE = ‘1’ triggers write operation and WE = ‘0’ triggers read operation. For read operation, the control block generates Precharge and Sense Amplifier controlling signals to read data from bitcells selected by the address decoder. The output data is latched out and can be accessed through IO block. For write operation, the control block generates the controlling signals for Precharge and Write Driver to write the data input, which is latched in by Data In Latch in IO block, into selected bitcells.

3.3. SYSTEM DESIGN

3.3.1 Array Block

An array 8x1 consists of 8 6T bitcells which share the same bitlines BT and BB as can be seen from Figure 3.2. The shared bitlines are connected to the periphery of the column. The bitcell can be selected by triggering its wordlines. The bus WL<0:7> is output of Address Decoder Block and is shared between the arrays. When a WL is selected, the bitcells from the same row of different array will be selected and read/write operation can be performed on that row.

3.3.2 Control Block

The control circuit is in charge of generating periphery control signals. Figure 3.3 describes the schematic diagram of a control circuit. The inputs of the circuit are CLK and WE. The outputs of the circuit are:

- PRCHB: This signal is used for turn on Precharge.
- DCLK: Clock signal used for latching new data input.
- WCLK: Clock signal used for triggering address decoder to generate WL signal.
- SAPR: This signal is used for turn on Sense Amplifier Precharge.
- SAE: This signal is used for enable Sense Amplifier, reading the data out.
- WPASS/RPASS: These signals are used for selecting read or write operation.

The control circuit is designed base on the desired signal's states in read operation, write operation and in low clock cycle where the memory is reset for new cycle. The truth table for the states is given in Table 3.1. This table does not include DCLK and WCLK because those are two delayed version of CLK. From the truth table, the circuit is designed from different logic gates in order to generate desired signal states based on the given inputs.

There are two special cases, in which one signal is the delayed version of another signal. The first case is WCLK, which is in fact CLK pulse delayed by the same amount of time it takes for the decoder to process information. This signal allows wordline to be turned on. Hence, it should be switched off when the cells are precharging in case the precharge process may change the cell's value. For this purpose, this signal is gated with CLK and PRCHB by a 3-input NAND gate as can be seen in Figure 3.3. The second case is similar, SAE is the delayed version of SAPR. After switching SAPR high for turning off Sense Amplifier Precharge, the SAE signal is brought high for reading out the cell's value. This signal should also be pulled down when SAPR turn the SA Precharge back on. Hence, SAE is gated with SAPR by an AND gate for this purpose.

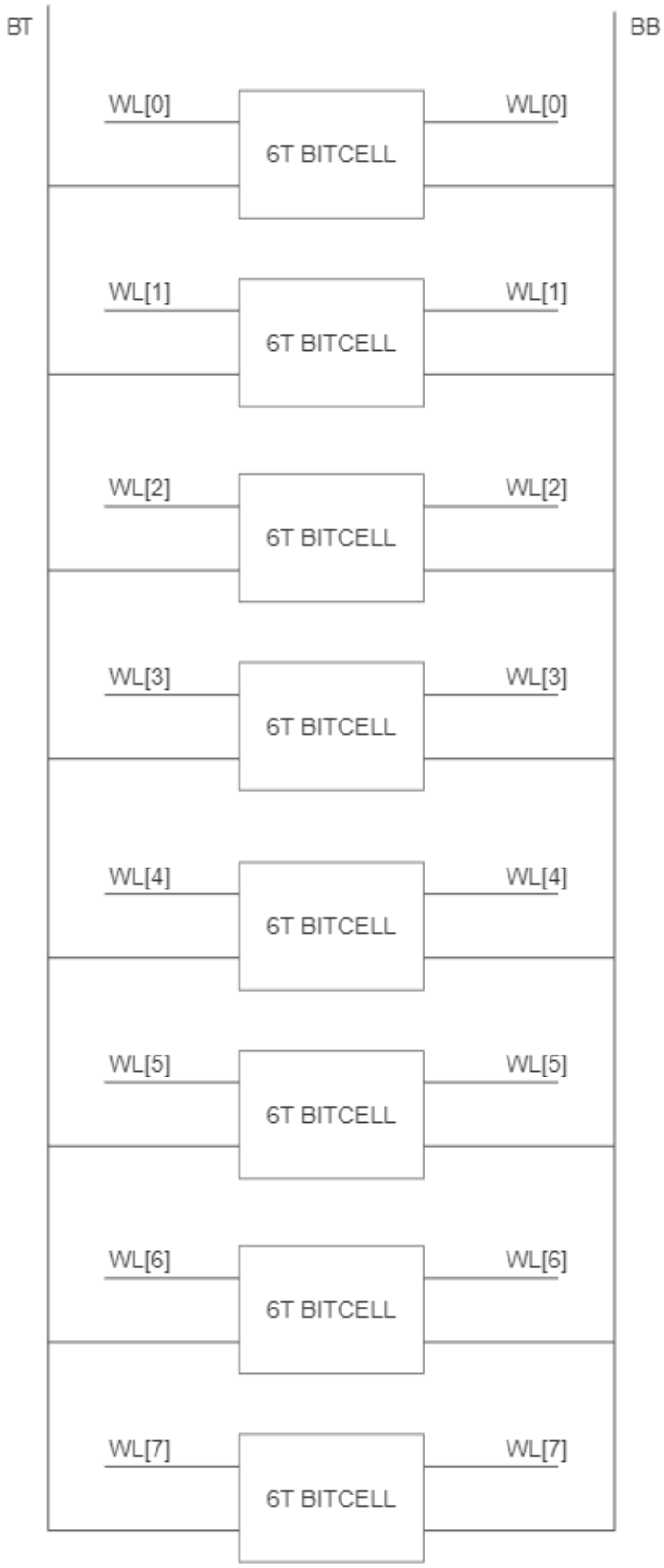


Figure 3. 2. *Schematic diagram of 8x1 Array*

Table 3. 1. *Control Circuit Truth Table*

CLK	**WE**	PRCHB	SAPR	SAE	WPASS	RPASS
0	**0**	0	0	0	0	1
0	**1**	0	0	0	0	1
1	**0**	1	1	1	0	0
1	**1**	1	0	0	1	1

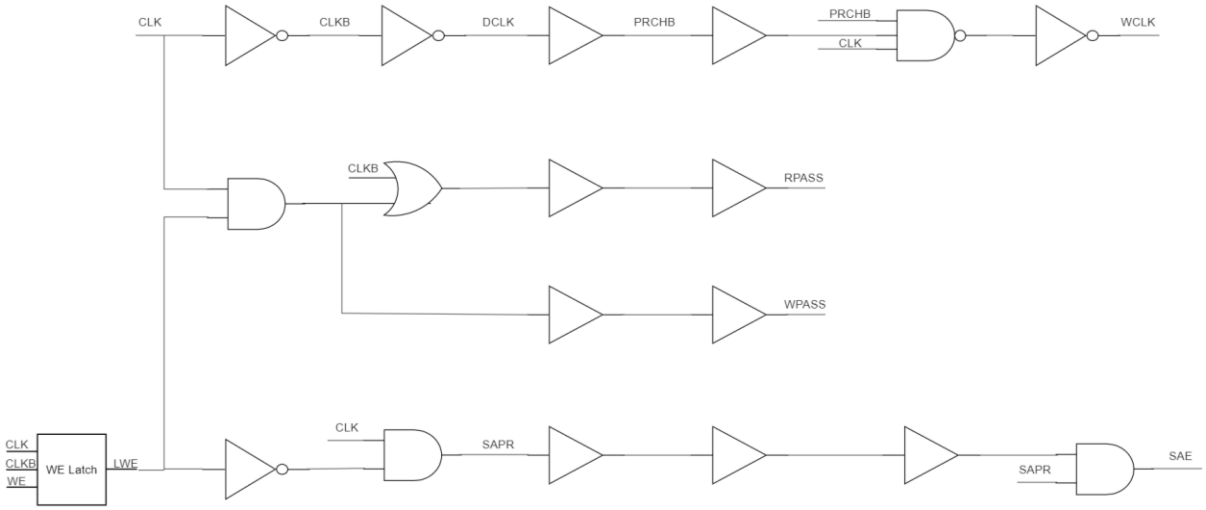


Figure 3. 3. *Schematic Diagram of Control Circuit*

3.3.3 Address Decoder Block

For selecting desired row to read or write from 8 bitcell rows, the address of that row needs to be inputed. For saving the number of input pins, address decoder is used for decoding 3-bit address inputs into 8-bit wordline selecting signals. Figure 3.4 represents the schematic of 3 to 8 decoder with ADR<0:2> as input and DEC <0:7> as ouput. In this design, the 3 to 8 decoder is made of eight 3 inputs AND gates, which actually are made from eight 3 inputs NAND gates and eight inverters. Depending on the input value, one of the outputs will be selected and pulled up. The truth table of this decoder is given in Table 3.2.

The outputs of the decoder are then gated with WCLK signal which permits wordline selection as can be seen from Figure 3.5. The reason of the existence of WCLK is making a better slew of WL. Because it takes some time for the decoder to decode the address, the slew of WL might be bad. The WCLK signal is generated from control circuit, which in reality is CLK signal delayed by the same amount of time to decode

address. The address decoder is still active even when in low pulse of WCLK. However, only the DEC<0:7> bus carry the address information in the low phase of WCLK. This period is the setup period of address which is mentioned in Chapter 2. When WCLK rises, the address information carried in DEC<0:7> bus is transparent to WL<0:7> bus and the selected wordline is turned on. Then, read or write operation can be performed on the selected bitcells.

Table 3. 2. *Address Decoder 3-to-8 Truth Table*

ADR <0:2>	WL <0>	WL <1>	WL <2>	WL <3>	WL <4>	WL <5>	WL <6>	WL <7>
000	**1**	0	0	0	0	0	0	0
001	0	**1**	0	0	0	0	0	0
010	0	0	**1**	0	0	0	0	0
011	0	0	0	**1**	0	0	0	0
100	0	0	0	0	**1**	0	0	0
101	0	0	0	0	0	**1**	0	0
110	0	0	0	0	0	0	**1**	0
111	0	0	0	0	0	0	0	**1**

3.3.4 Periphery Block

3.3.4.1. Precharge Circuit

The precharge circuit is used for precharging the bitlines to VDD. Before read/write operation, bitlines need to be at VDD level. For read operation, bitlines need to be precharged so that the voltage between bitlines after sensing the bitcell can be compared. In other case, the write operation will drive one bitline down. Hence, the bitlines have to be precharged so that after pulling down, there will be one bitline stay as VDD. The schematic diagram of precharge circuit is given in Figure 3.6.

The circuit is built up from 3 PMOS controlled by the PRCHB signal. When the PRCHB signal pulled down, the 3 PMOS are turned on. The 2 PMOS on the sides are in charge of pulling BT and BB node to VDD level. While the middle PMOS is used for stabilizing the voltage between the two nodes. When PRCHB signal brought high again,

BT and BB can be discharged or changed voltage level for read or write purposes. After finishing read/write operation, the precharge is turned on again for resetting bitlines.

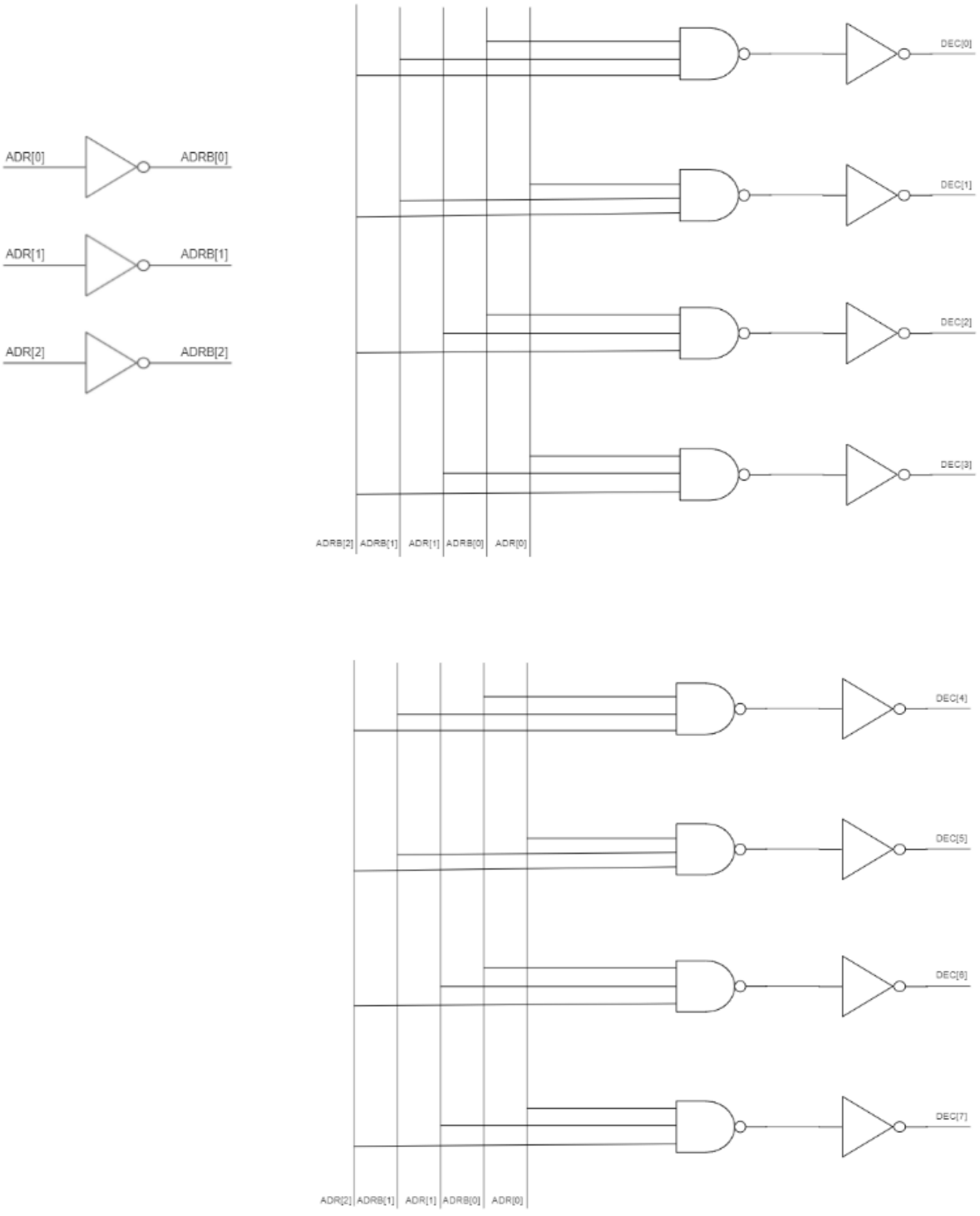


Figure 3. 4. *Schematic Diagram of 3-to-8 Address Decoder*

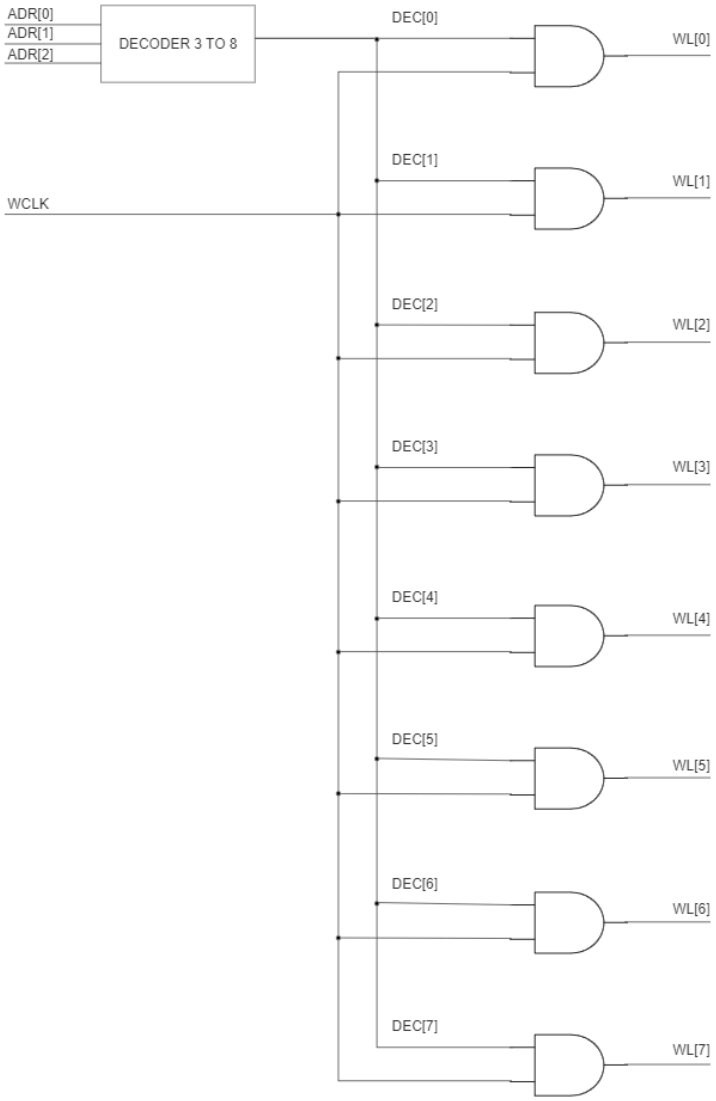


Figure 3. 5. *Schematic Diagram of Address Decoder with Gating WCLK*

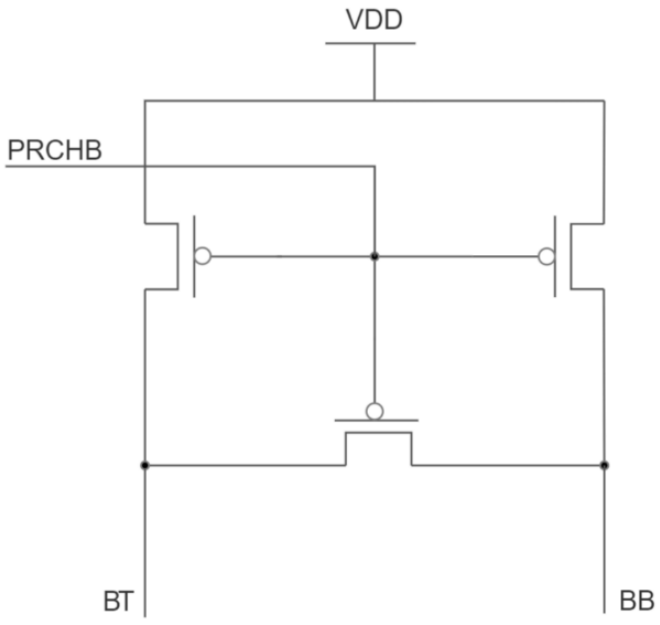


Figure 3. 6. *Schematic Diagram of Precharge Circuit*

3.3.4.2. Sense Amplifier

For sensing such a small voltage difference between the bitlines, a sense amplifier is necessary for reading bitcell's value. For this design, a latch type voltage sense amplifier is used and its schematic diagram is given in Figure 3.7. This sense amplifier replicates the behavior of a real bitcell with two bitlines on the sides. The cross-coupled inverters are the replica of the bitcell while two sensing bitlines (XT and XB) acts the same as the real bitlines (BT and BB).

A latch type voltage sense amplifier contains a precharge circuit controlled by an SAPR signal. This precharge is used for precharging the sensing bitlines (XT and XB) before read operations. On low clock cycle, SAPR is low and the sensing bitlines are precharged to VDD level. When clock triggers for a new read cycle (when WE = '0'), the SAPR signal is brought high which allows the sensing bitlines to discharge.

In this circuit, SAE signal is in charge of enabling the Sense Amplifier when the voltage difference between the two actual bitlines is sufficient. This signal is inputted into the sense amplifier as two separate inputs which are SAEP and SAEN. When SAE is low, the SAEP signal trigger the two PMOS connected to the read bitlines (RT and RB). This transparency lets XT and XB discharging through the PMOS to RT and RB. By this method, the sensing bitlines can mirror the voltage difference between the read bitlines. When the difference is sufficient, the SAE signal is pulled up which leads to the activation of the cross-coupled inverters. The SAEN signal is switched to '1', unblocking the path to VSS. From the voltage difference between XT and XB, the cross-coupled inverters can pull the lower voltage sensing bitline to VSS and the value can be latched out to data output.

3.3.4.3. Write Driver

The write driver circuit simply is two big inverters as can be seen from Figure 3.8. The inputs of the circuit are the outputs WTI and WBI of the Data Input Latch in IO Block. The outputs of the circuit (WT and WB) are the write bitlines which are gated with the actual bitlines at Read Write Pass Circuit. The size of the inverters is much larger than the normal inverters so that they are strong enough to pull down swiftly the bitlines down.

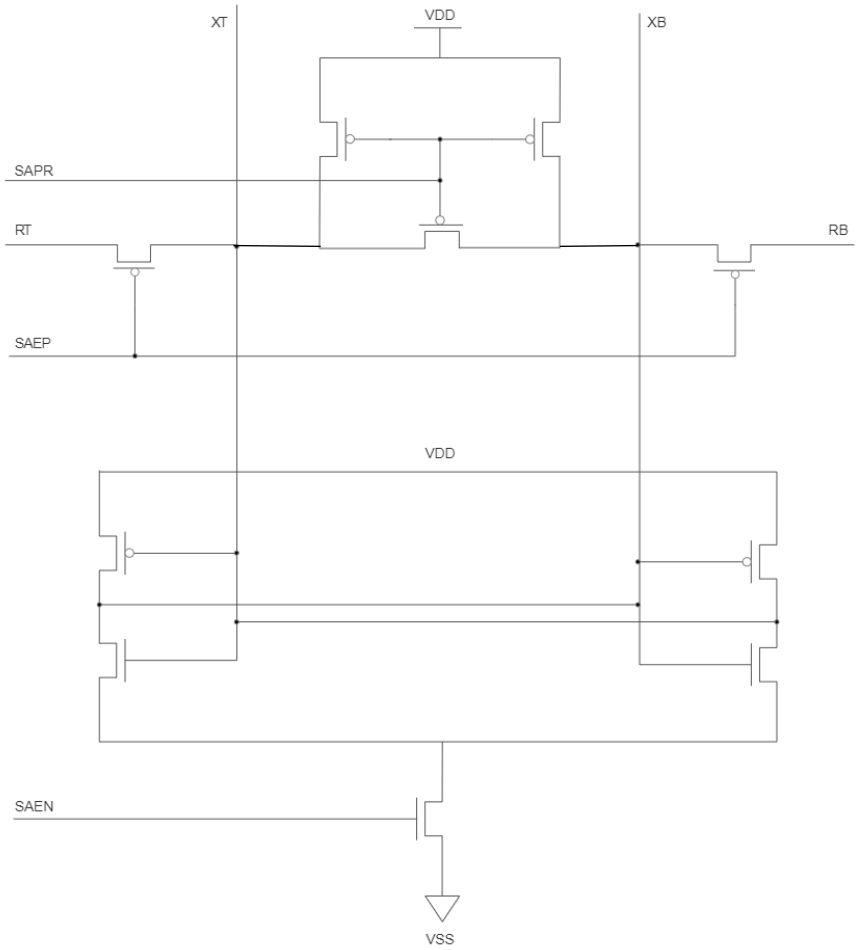


Figure 3. 7. *Schematic Diagram of Sense Amplifier*

After latching the new data for write operation, the WTI and WBI carry the invert values of bitlines. For instance, if ‘1’ is written to a bitcell, WTI will be ‘0’ and WBI will be ‘1’. These signals pass through the driving inverters, pulling up WT and pulling down WB. By pulling down WB while keeping WT at VDD level, the bitline bar (BB) is pulled down and ‘1’ is written to the bitcell. The same thing happens to the bitline (BT) when ‘0’ is written.

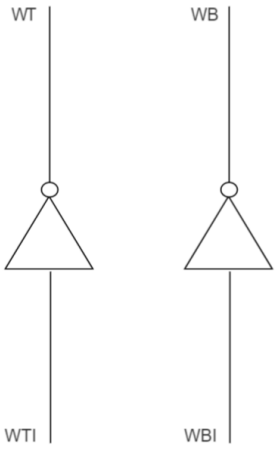


Figure 3. 8. *Schematic Diagram of Write Driver*

3.3.4.4. Read/Write Pass Circuit

Figure 3.9 describes the schematic diagram of the Read/Write Pass Circuit. The main function of this circuit is to direct the actual bitlines to read bitlines or write bitlines depending on the selected operation. The inputs of this circuit are RPASS and WPASS. These two signals have the same value to each other and to WE signal on high pulse of CLK. On the low phase of CLK, RPASS is kept high while WPASS is kept low to reject read/write operation.

With WPASS = '1', the NMOS transistors are turned on, connecting actual bitlines (BT and BB) with write bitlines (WT and WB). This connection allows the data value on write bitlines to be transparent to the actual bitlines. In other case, when RPASS = '0', the PMOS transistors are switched on, connecting the actual bitlines (BT and BB) with read bitlines (RT and RB). This allows the read bitlines to inherit the voltage level of the actual bitlines.

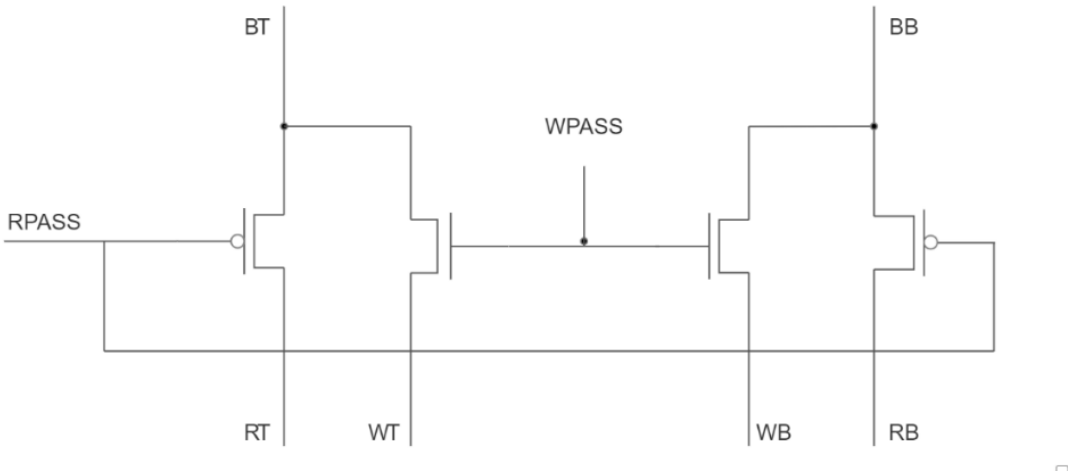


Figure 3. 9. *Schematic Diagram of Read/Write Pass Circuit*

3.3.5 IO Block

3.2.5.1. Data Input Latch

Figure 3.10 represents the schematic diagram of the Data Input Latch which is designed based on the truth table given in Table 3.3. This circuit is used for latching the input data in write operation. For such function, clock inverters are used for performing latching data.

The clock inverter is an inverter which can be accessede or unaccesed depending on the clock signal. The first clock inverter is used as a gate of new data. On low pulse of DCLK, the value of DIN is transparent to WTI and WBI. On high pulse of DCLK, this clock inverter blocks new data from DIN. The second inverter works on the reverse cycle of DCLK. On low clock pulse, this clock inverter is disabled. Then, on high phase of DCLK, it turns into a latching inverter which keeps the node not floating while DIN

is blocked. This operation helps keeping the outputs WTI and WBI stable and unchangeable when write operation is triggered.

Table 3. 3. *Data Input Latch Truth Table*

DCLK	WTI	WBI
0	DIN_b	DIN
1	Latched DIN_b	Latched DIN

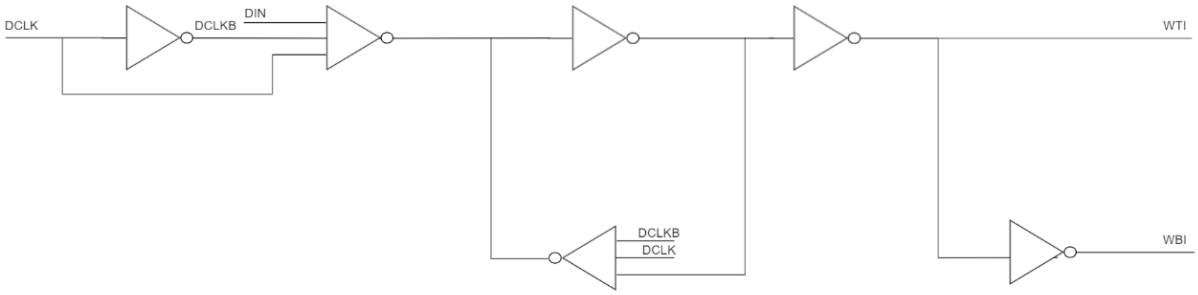


Figure 3. 10. *Schematic Diagram of Data Input Latch*

3.2.5.2. Data Output Latch

Figure 3.11 illustrates the schematic diagram of the Data Output Latch. As its name, the circuit function is to latch out the data read from Sense Amplifier. The truth table used to design this circuit is given in Table 3.4. The inputs of this circuit are QT and QB, which are in fact sensing bitlines (XT and XB) after passing through a buffer. The output of this circuit is QOUT, which is also the final read output of the memory.

When there is read operation performed, QT and QB will have opposite value and that value will be transparent to QOUT. For instance, in read ‘0’ operation, QT will have value ‘0’ and that will trigger the PMOS transistor, pulling the gate of the inverters up. This will trigger the QOUT to fall to ‘0’.

In case there is no read operation, QT and QB is kept high because the sensing bitlines are not discharged. With QT = ‘1’, the PMOS transistor is inactive. QBI is also pulled down because QB = ‘1’. This leads to the NMOS transistor is also inactive. For this case, the QOUT value should continue to keep its last value. If the previous value is ‘1’, then the NMOS transistors will be active with QT = ‘1’ and PMOS transistors will be inactive with QOUT = ‘1’. This will pull the gate of inverters down, keeping QOUT at ‘1’. In case the last QOUT value is ‘0’, the PMOS transistors are active instead of NMOS transistors. This will pull the gate of inverters up, keeping QOUT at ‘0’.

Table 3. 4. *Data Output Latch Truth Table*

QT	QB	QOUT
0	1	0
1	0	1
1	1	0

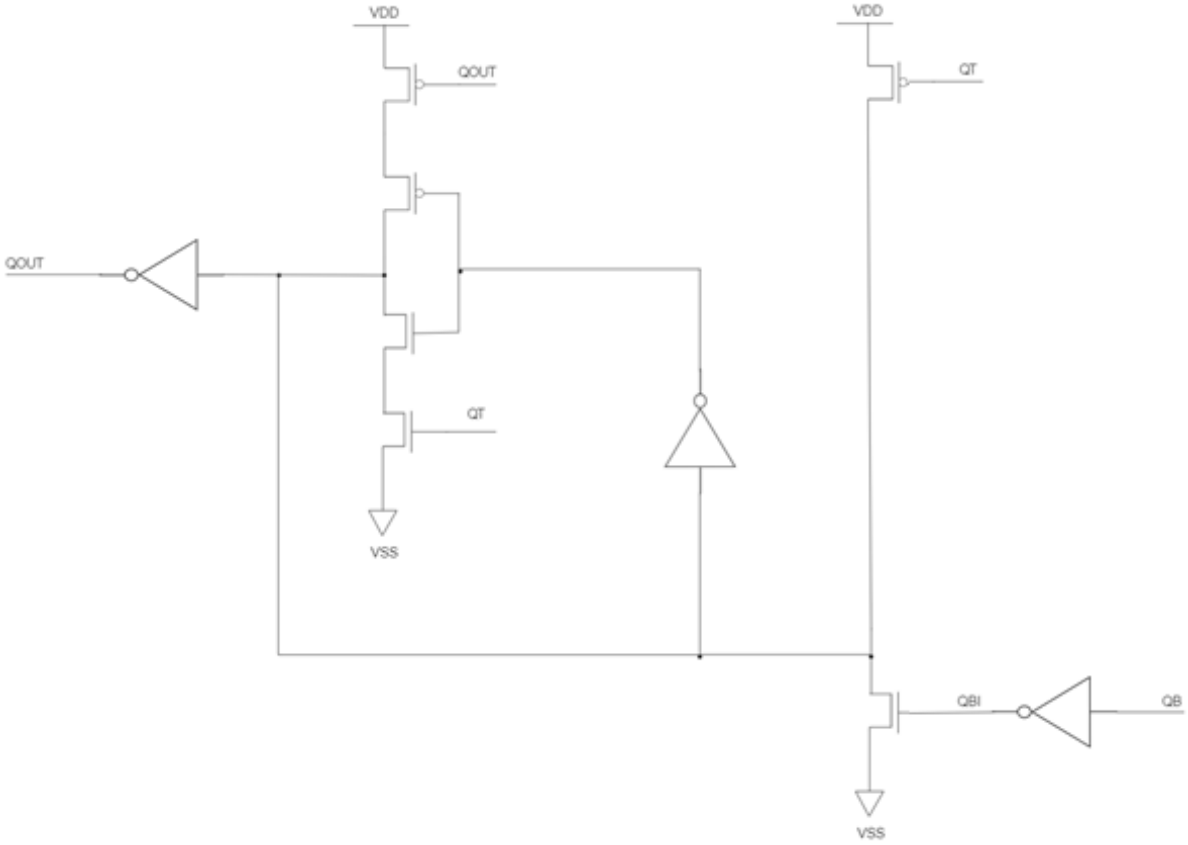


Figure 3. 11. *Schematic Diagram of Data Output Latch*

CHAPTER 4: RESULTS

4.1. SRAM 8X8 INSTANCE

From the schematics and layouts of separating devices, the 8x8 SRAM memory is formed by connecting all the bitcells, the control circuit, the address decoder, the IO block and other peripheries together. Figure 4.1 illustrates the schematic of a complete 8x8 SRAM instance. The first row includes one address decoder connected with 8 columns of bitcells through WL<0:7> bus. Each column of bitcells is connected with 1 set of peripheries (which includes a precharge, a read/write pass circuit and a sense amplifier) and 1 IO block (which includes a data input latch and a data output latch) through bitlines. The peripheries and IO block are controlled by the signals which are generated by the control circuit in the far left of the memory. There are 4 external inputs including CLK, WE, DIN<0:7> and ADR<0:2> and 1 external output bus which is QOUT<0:7>.

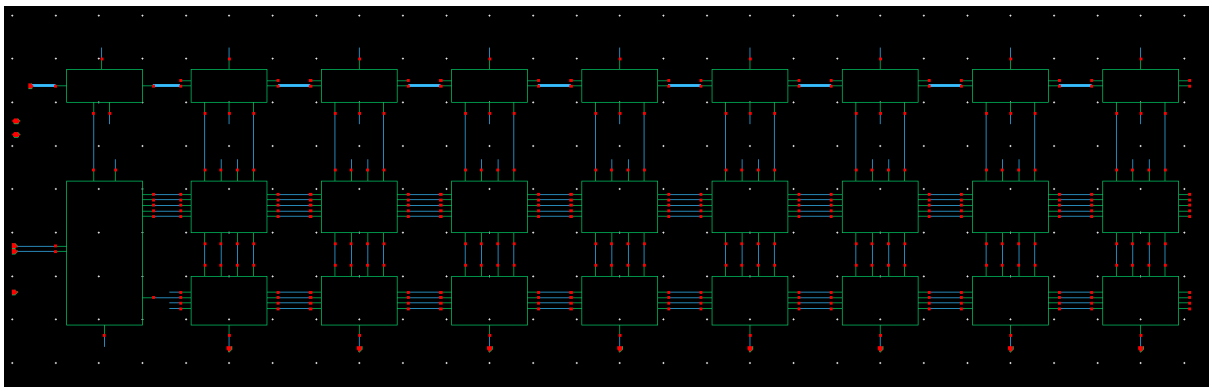


Figure 4. 1. *SRAM 8x8 Instance Schematic*

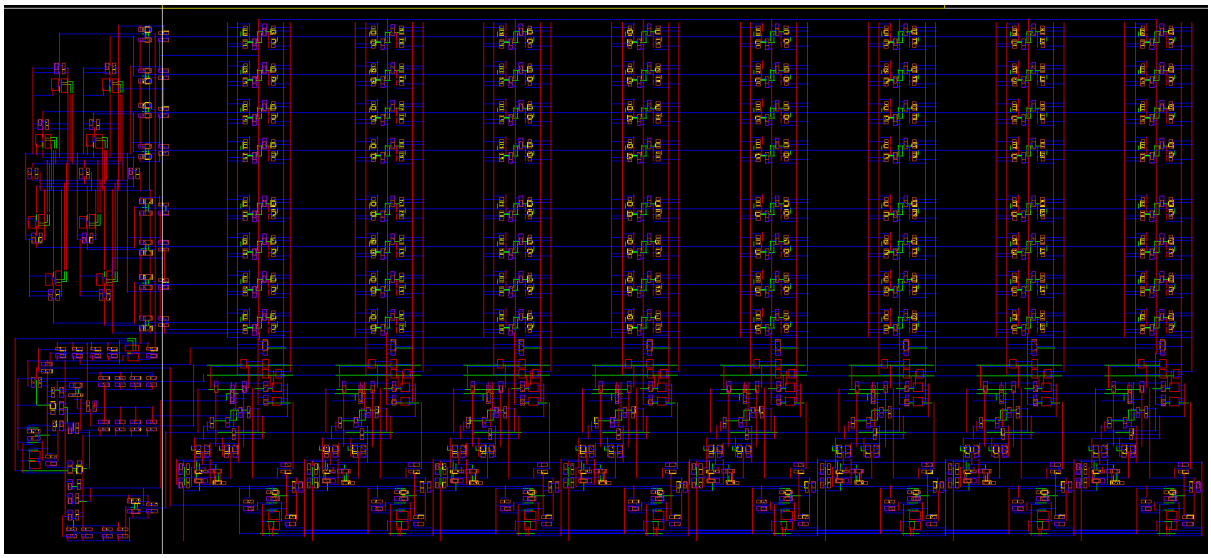


Figure 4. 2.*SRAM 8x8 Instance Layout*

After creating the full instance schematic and testing its operation, the layout of 8x8 SRAM memory is drawn using the available layouts. All of the layouts including 8x8 Array, Address Decoder, Control Circuit, Precharge, Read/Write Pass Circuit, Sense Amplifier, Data Input Latch and Data Output Latch are merged and connected to each other. The metal direction rule is still applied with metal 1 for horizontal signals and metal 2 for vertical signals. The layout result is given in the Figure 4.2 and is fully verified by LVS and DRC check.

4.2. SRAM ARRAY

4.1.1. SRAM 6T Cell Schematic and Layout Result

From the schematic diagram presented in Figure 4.3, the 6T memory cell schematic and layout are formed using 4 NMOS transistors and 2 PMOS transistors. Figure 4.3 illustrates the schematic of the cell which is created by Virtuoso Schematic Editor L. The memory cell can be accessed by triggering the input pin WL to '1' and read/write operations can be performed. To read the cell's value, the precharged bitlines sense the voltage at RT and RB nodes. To write new data to the cell, one of the bitlines is tied to ground in order pull down RT or RB node, depending on the data's value. To assure the stability of read and write operation to the cell, the ratio between pull up transistors, pull down transistors and pass gate transistors must be scaled properly. Specifically, the pulldown transistors (MN0 and MN3) must be stronger than the pass gate transistors (MN1 and MN2) and the pass gate transistors must be stronger than the pull up transistors (MP0 and MP1). The size of pull up transistors, pull down transistors and pass gate transistors of this design are 120nm, 400nm and 350nm, respectively.

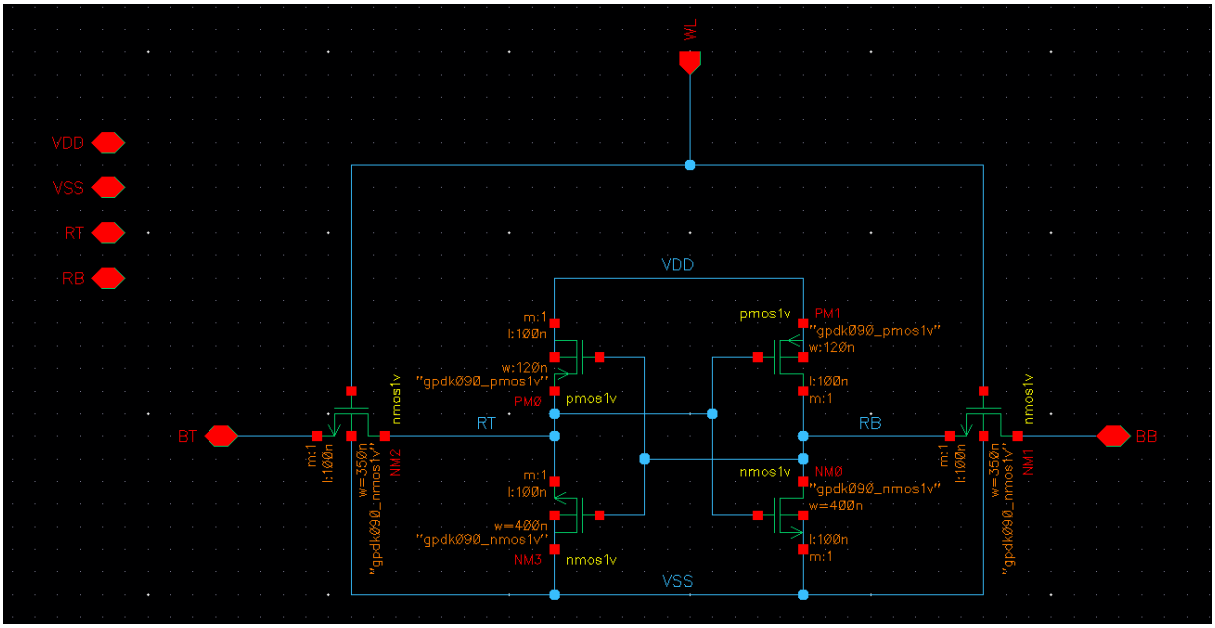


Figure 4. 3. *SRAM Memory Cell Schematic*

From the schematic, the memory cell's symbol which contains all cell's inputs and outputs is created and is presented in Figure 4.4. The cellview is named ram6t and has 8 pins:

- Bitlines pins: BT and BB
- Cell's value node pins: RT and RB
- Wordline pins: WL (2 pins)
- Power pins: VDD and VSS

The WL pin is intentionally duplicated for connectivity purpose. Since memory cells in the same row share the same wordline, the WL pin is placed in both the left side and the right side of the symbol.

After creating the cell and verifying its operation, the layout of the 6T SRAM cell is drawn using Virtuoso Layout Suite XL. Figure 4.5 describes the layout of the memory which is built based on the 6T SRAM thin cell layout scheme. In this layout, metal-1 runs strictly in the horizontal direction and metal-2 runs strictly in the vertical direction. Metal-1 is used for the WL signal while metal-2 is used for the power rails and the BLs. The result layout is fully verified by running DRC and LVS check using Assura.

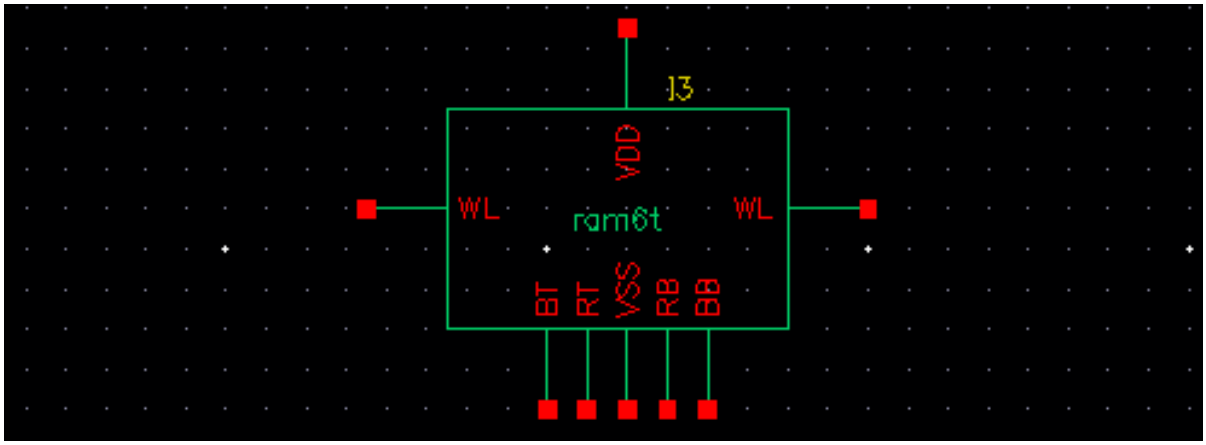


Figure 4. 4*. SRAM Memory Cell Symbol*

4.1.2. 8x1 6T SRAM Cell Array Schematic and Layout Result

Figure 4.6 shows the schematic of an 8x1 6T SRAM Cell Array, or can be simply seen as an 6T SRAM cell column. SRAM array contains 8 6T memory cells in the same column, which also means they share the same bitlines, BT and BB. The memory cells can be selected by triggering one bit of bus WL<0:7>. For instance, if the desired cell is in row 4, the WL<4> is brought high while all the other WL are kept low. For observation purpose, bus RT<0:7>, RB<0:7> are made to view the value of each cell in the column.

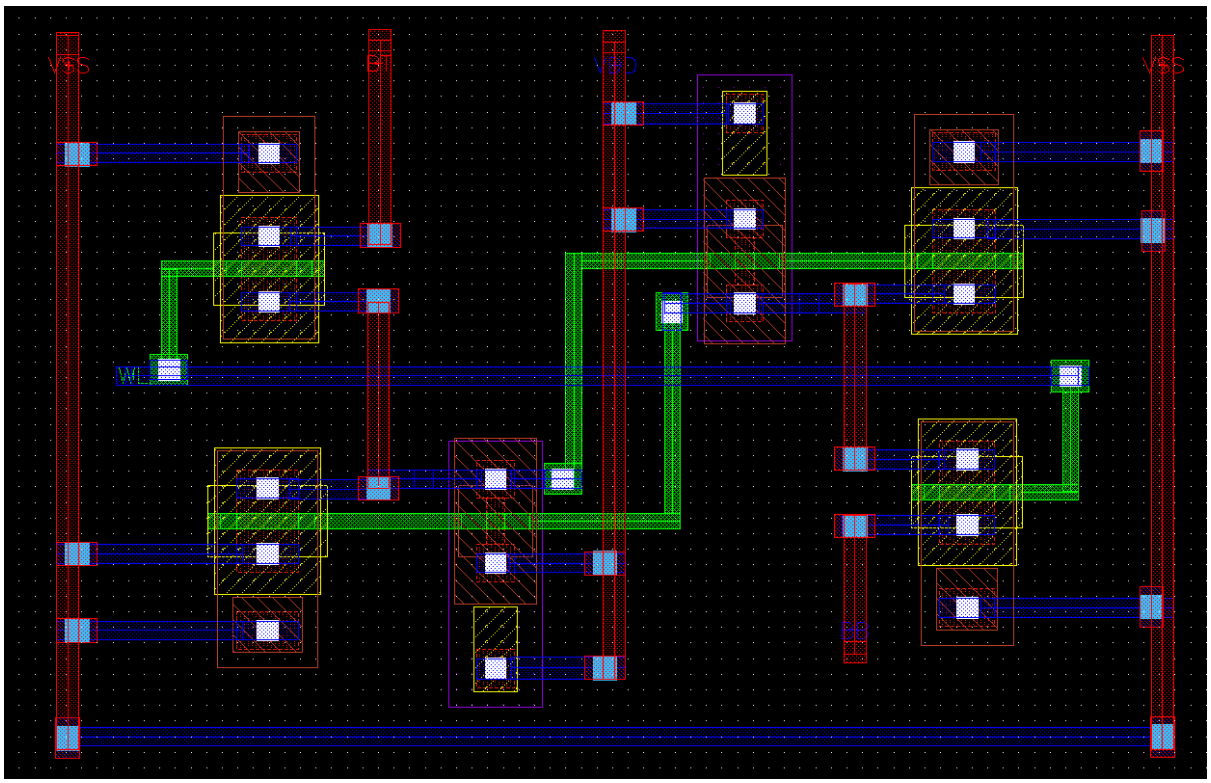


Figure 4. 5.*6T SRAM Cell Layout*

From the schematic, the memory cell's symbol which contains all cell's inputs and outputs is created and is presented in Figure 4.7. The cellview is named arr8x1 and has 8 pins:

- Bitlines pins: BT and BB
- Wordline bus: WL<0:7>
- Cell value buses: RT<0:7> and RB<0:7>
- Power: VDD and VSS

After creating the schematic and verifying the array's operation, the layout of 8x1 6T SRAM Cell Array is created using Virtuoso Layout Suite XL. The layout of 8x1 6T SRAM Cell Array is formed using 8 6T SRAM thin cell layout which have the bitlines connected with each other and share the same power net. Figure 4.8 represents the layout result, this layout is fully verified by LVS and DRC check.

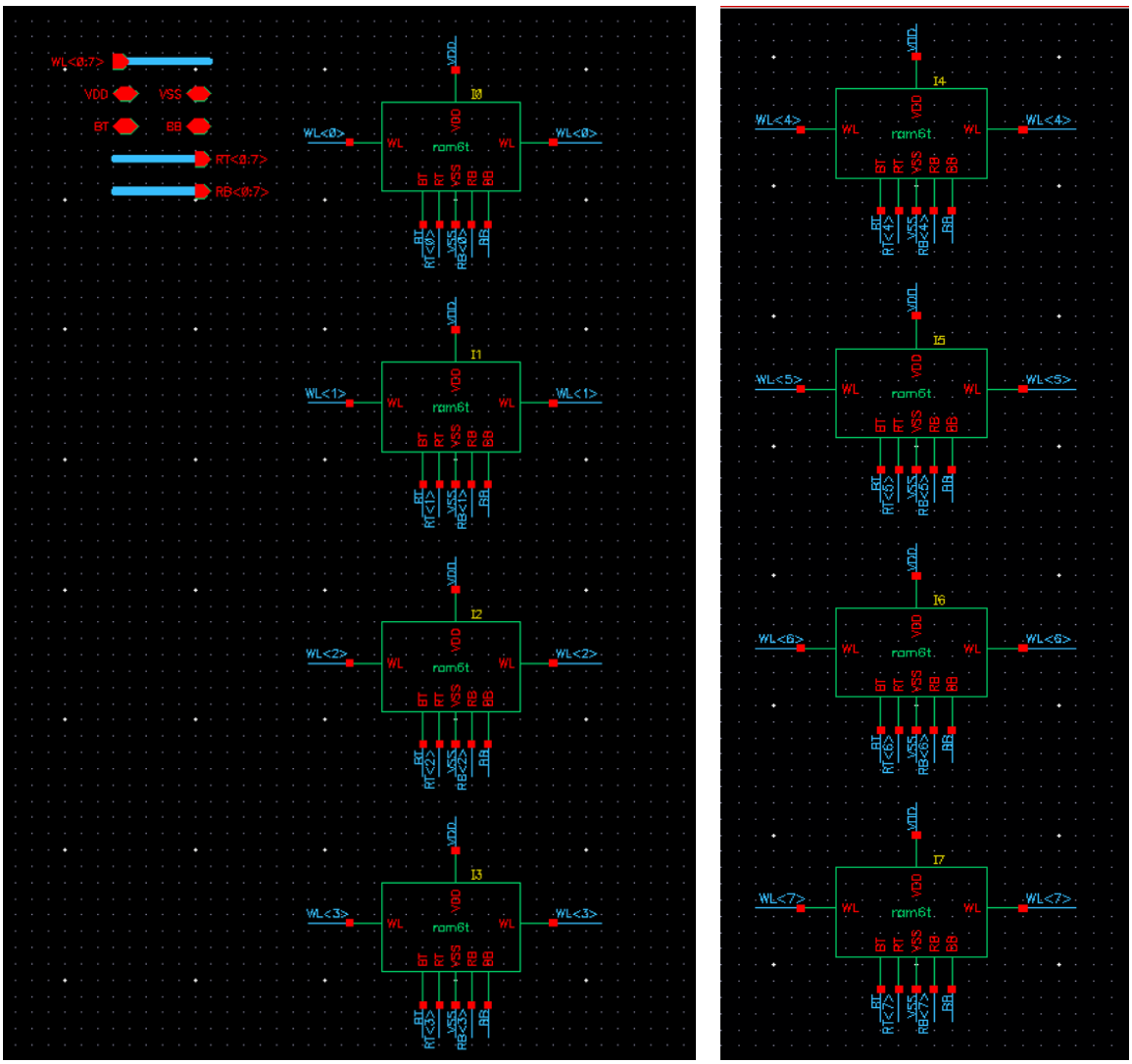


Figure 4. 6.*SRAM Array 8x1 Schematic*

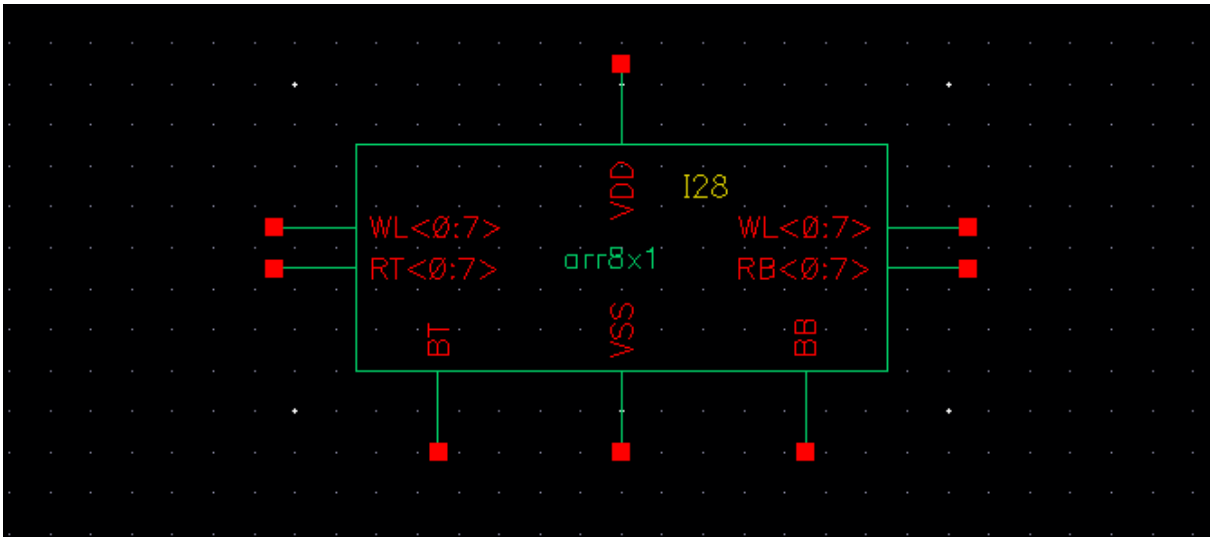


Figure 4. 7.*8x1 6T SRAM Cell Array Symbol*

4.3. SRAM PERIPHERY

4.2.1. Control Circuit

For creating signals used in controlling the peripheries of the memory, multiple logic gates are used to form a control circuit. Figure 4.9 shows the schematic of the control circuit. As can be seen from the figure, most of the gates are buffers, which are used for delaying the signals. The purpose of these delays are adjusting proper internal signals racing. The WE_lat circuit is in fact a latching circuit with the same operation as the Data Input Latch. It is used for latching the write enable signal while read/write operations are performed. The circuit is designed based on the truth table given in Table 3.1.

From the schematic, control circuit's symbol which contains all cell's inputs and outputs is created and is presented in Figure 4.10. The cellview is named CLKGEN and has 11 pins:

- Clock pins: CLK
- Write enable pin: WE
- Wordline clock pin: WCLK
- Data clock pin: DCLK
- Precharge control pin: PRCHB
- Sense Amplifier control pin: SAPR and SAE
- Read/Write Pass pin: RPASS and WPASS
- Power: VDD and VSS

After creating the schematic and verifying the circuit's operation through simulation, the layout of the Control Circuit is made using Virtuoso Layout Suite XL. Figure 4.11 illustrates the layout result, this layout is fully verified by LVS and DRC check using Assura LVS and DRC Check.

4.2.2. Precharge

From the schematic diagram of Precharge shown in Figure 3.6, a schematic view of Precharge circuit is generated. Figure 4.12 shows the schematic of Precharge circuit which contains 3 PMOS transistors. The input of this circuit is PRCHB which controlls the access to the 3 PMOS. When PRCHB switched to '0', PM0 and PM1 pull the outputs (BT and BB) up to VDD. PM2 is used for neutralyzing the voltage between the two bitlines. When PRCHB rises to '1', all of the transistors are turned off and the outputs are free to discharge.

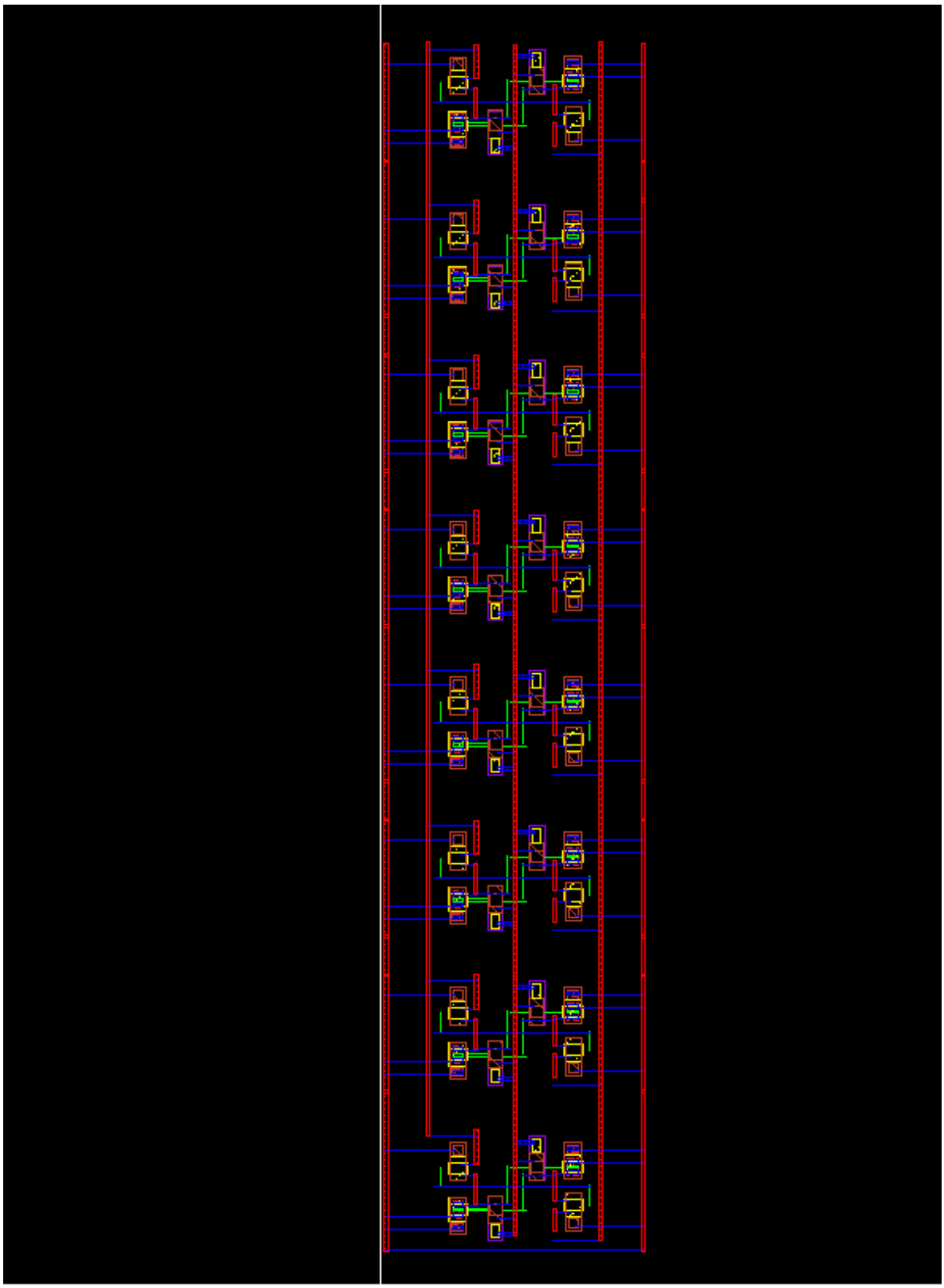


Figure 4. 8.*8x1 6T SRAM Cell Array Layout*

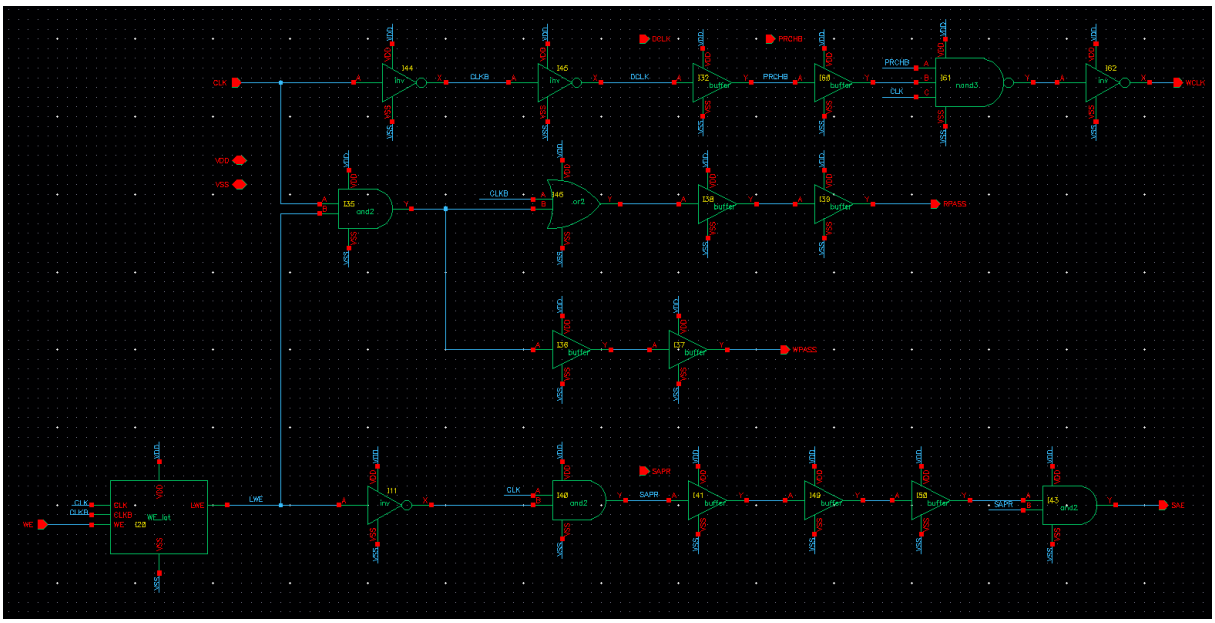


Figure 4. 9.*Control Circuit Schematic*

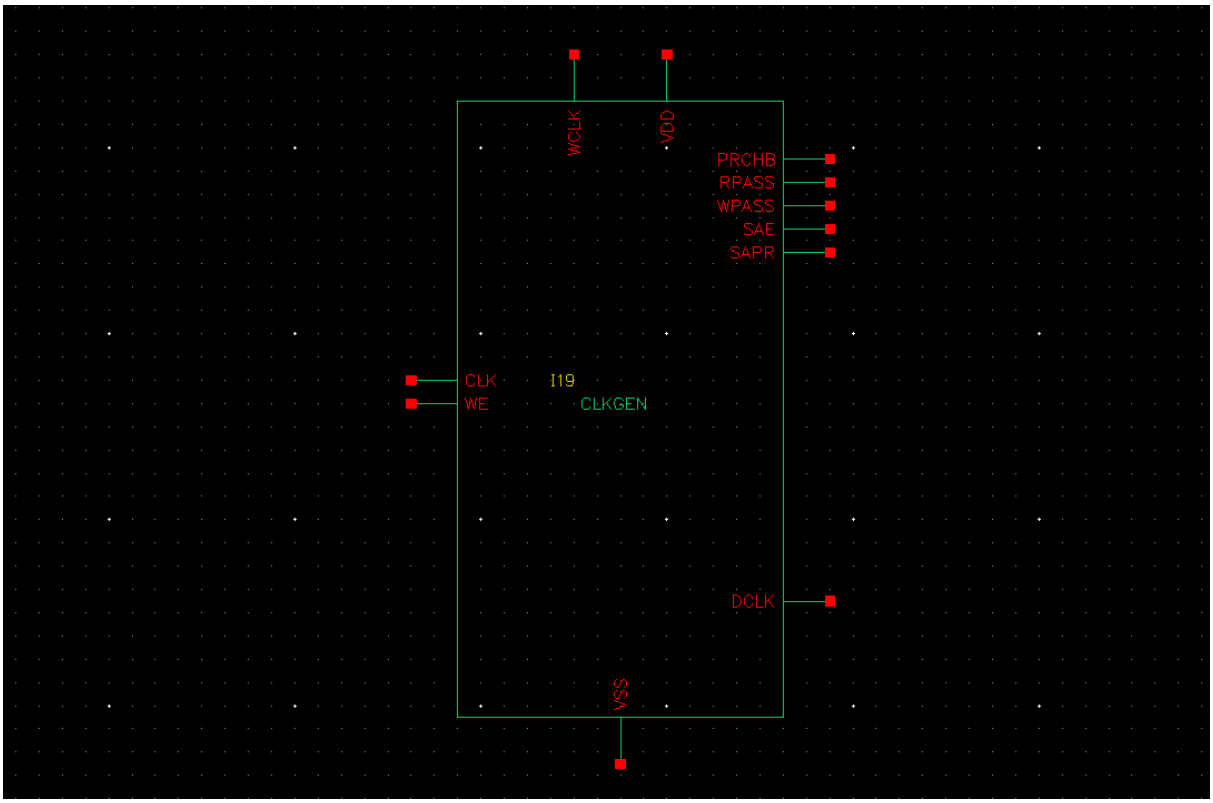


Figure 4. 10.*Control Circuit Symbol*

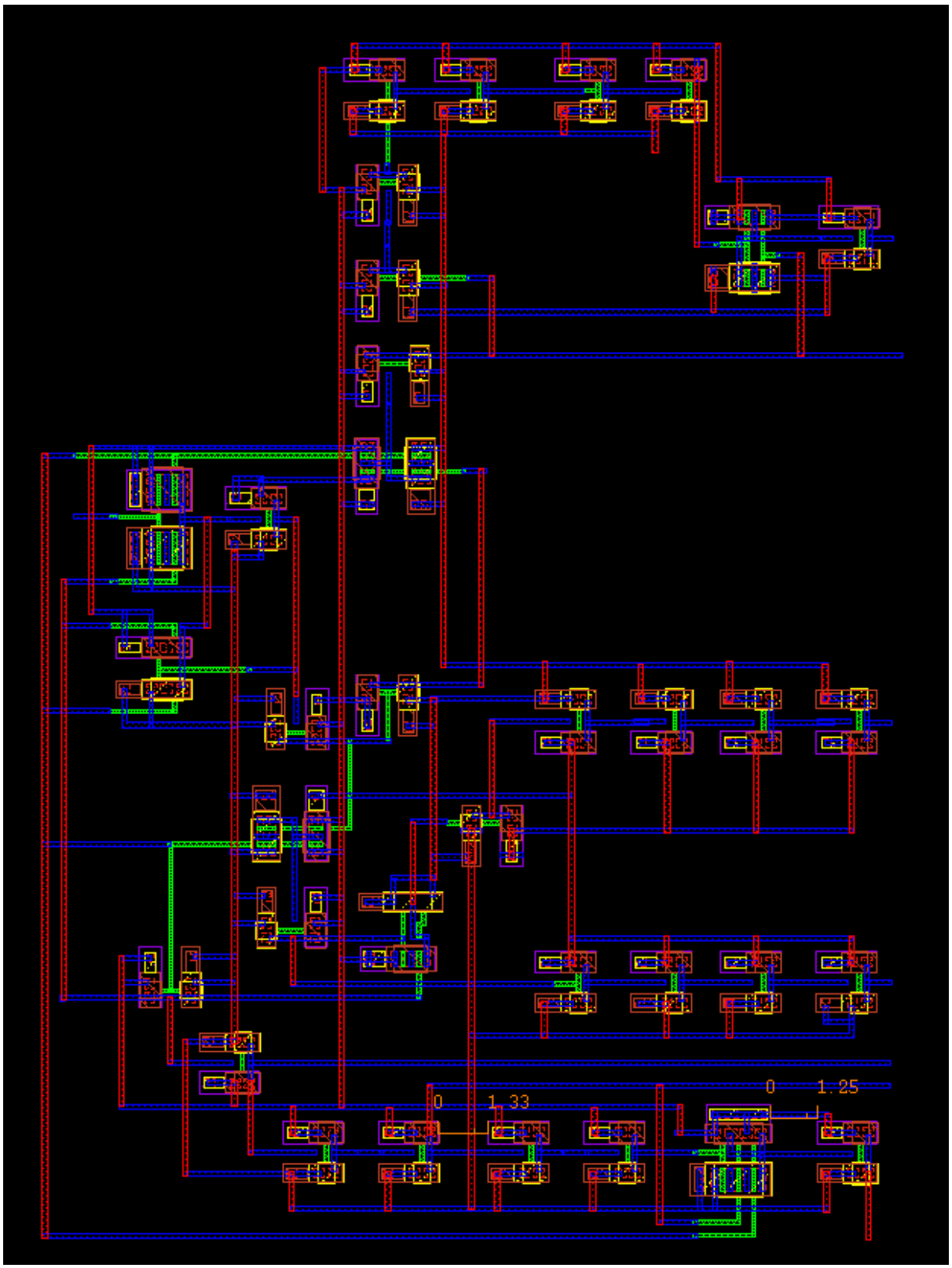


Figure 4. 11.*Control Circuit Layout*

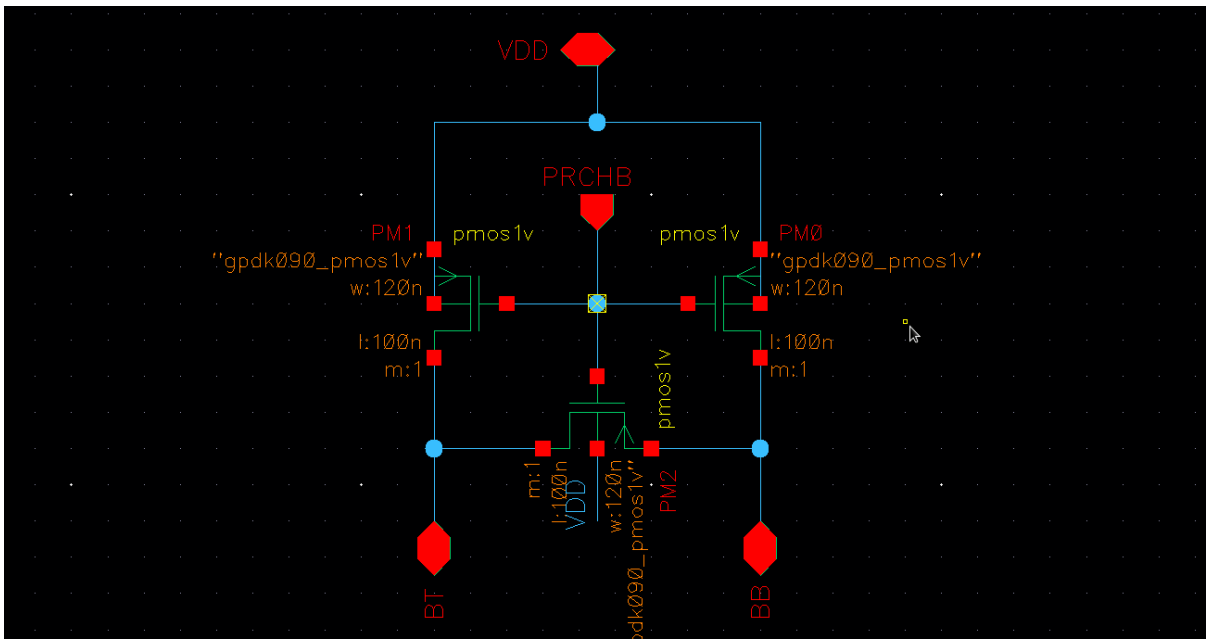


Figure 4. 12.*Precharge Schematic*

From the schematic, the precharge's symbol which contains all input and outputs is created and is shown in Figure 4.13. The cellview is named prch and has 4 pins:

- Bitlines pins: BT and BB
- Precharge signal pin: PRCHB
- Power pin: VDD

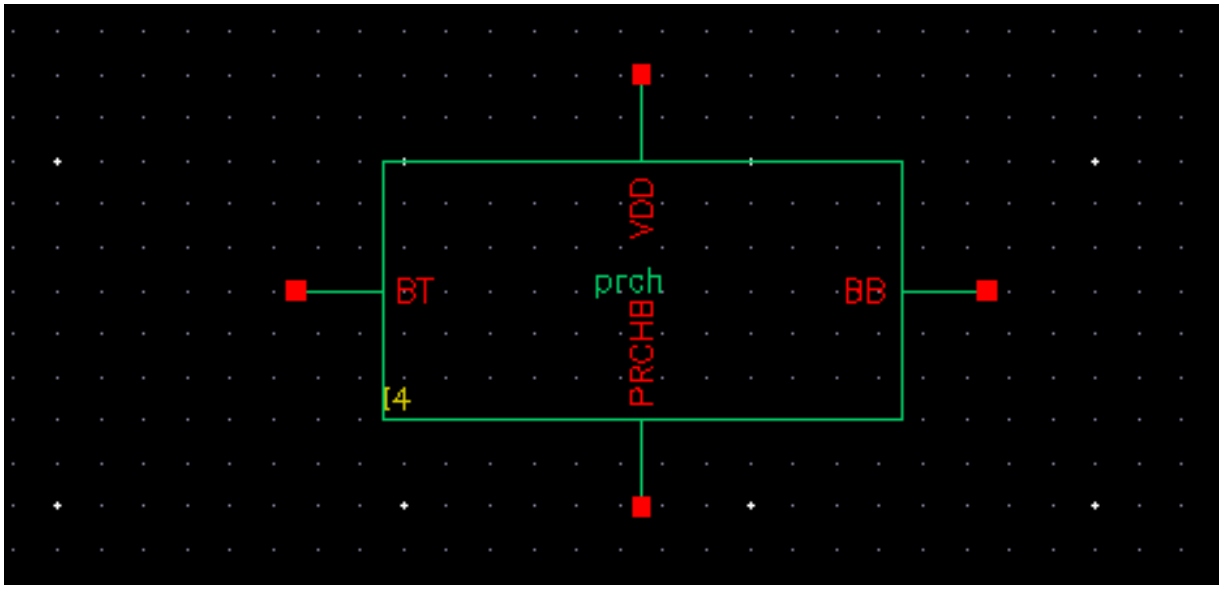


Figure 4. 13.*Precharge Symbol*

Same as other circuit, after the schematic is created and its operation is verified through simulation, the layout of the Precharge Circuit is made using Virtuoso Layout Suite XL. Figure 4.14 shows the layout of the Precharge Circuit result. As can be seen, one 3 fins PMOS is used instead of three 1 fin PMOS for area saving. The metal 1 is

used for horizontal connections while metal 2 is used for vertical connection to avoid congestion.

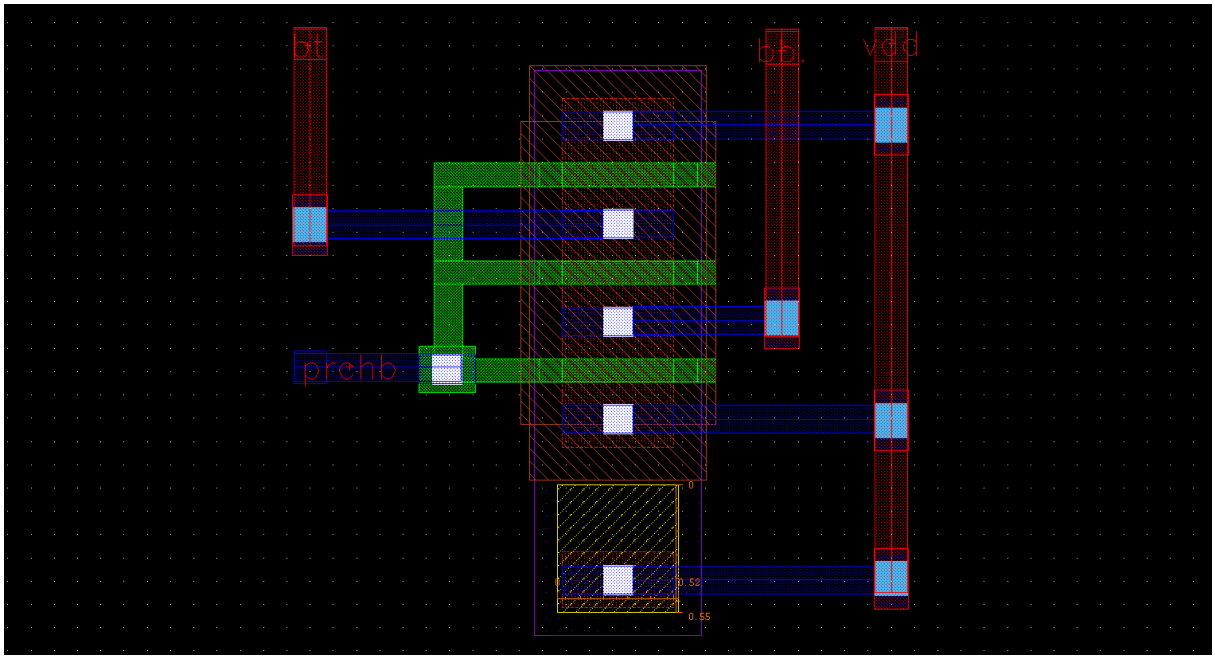


Figure 4. 14.*Precharge Layout*

4.2.3. Sense Amplifier

In this circuit, one cross-coupled inverters, one precharge circuit, 2 pass gates PMOS transistors and 1 ground gating NMOS transistor are used. The schematic of the Sense Amplifier is given in Figure 4.15. The precharge circuit is controlled by input signal SAPR. This precharge is used to precharge the sensing bitlines, XT and XB. The sensing bitlines are discharging after turning off the precharge and SAEP is low. The voltage difference is then transformed to contrary value by the cross-coupled inverters when SAEN is switched high, unblocking the ground gating. The output of this circuit is the contrary value of the sensing bitlines, XT and XB.

From the schematic, the sense amplifier's symbol which contains all input and outputs is created and is shown in Figure 4.16. The cellview is named sa and has 9 pins:

- Sensing bitlines pins: XT and XB
- Read bitlines pins: RT and RB
- Sense Amplifier enable pins: SAEP (triggered by '1') and SAEN (triggered by '0')
- Sense Amplifier precharge pin: SAPR
- Power: VDD and VSS

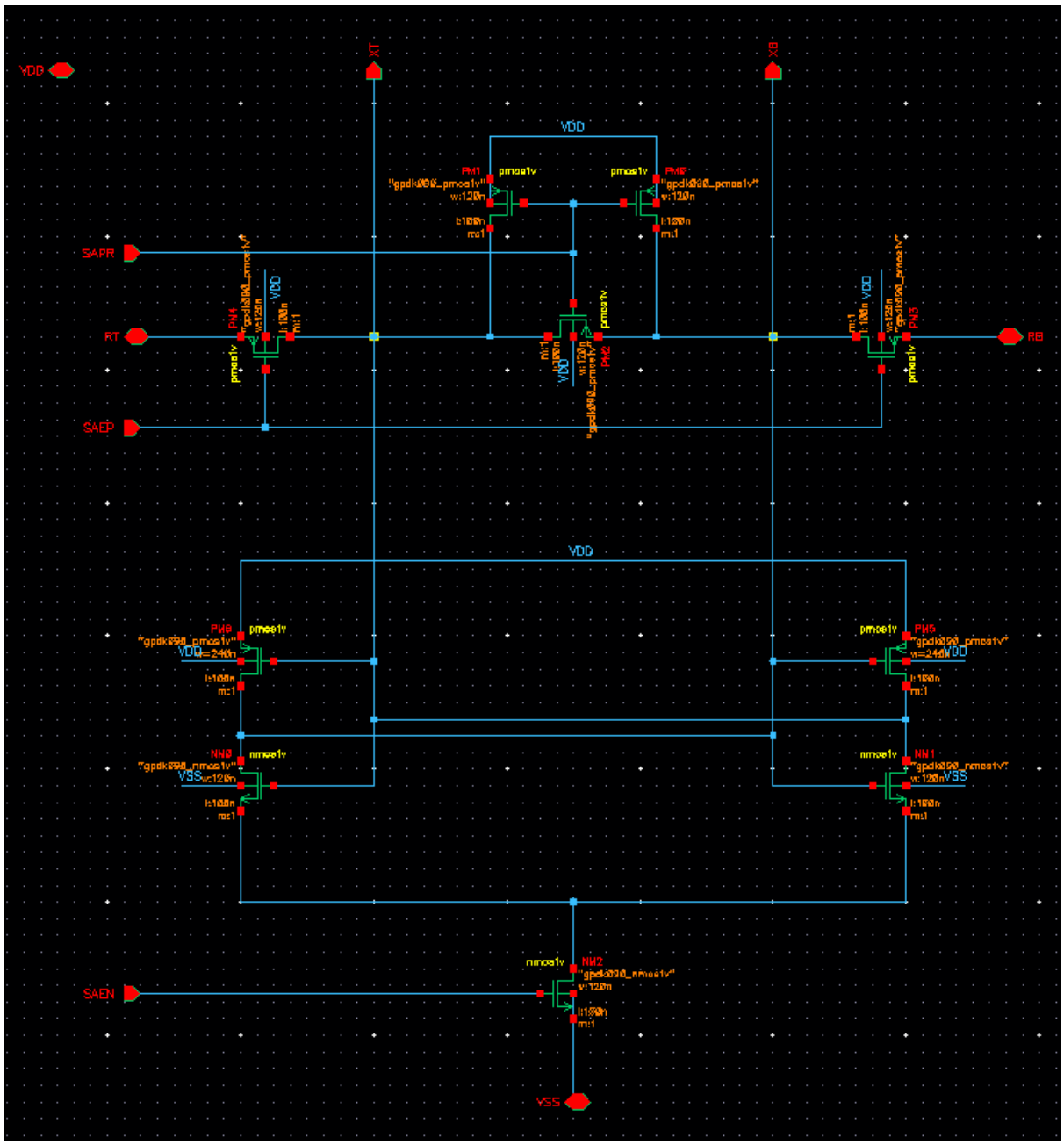


Figure 4. 15.*Sense Amplifier Schematic*

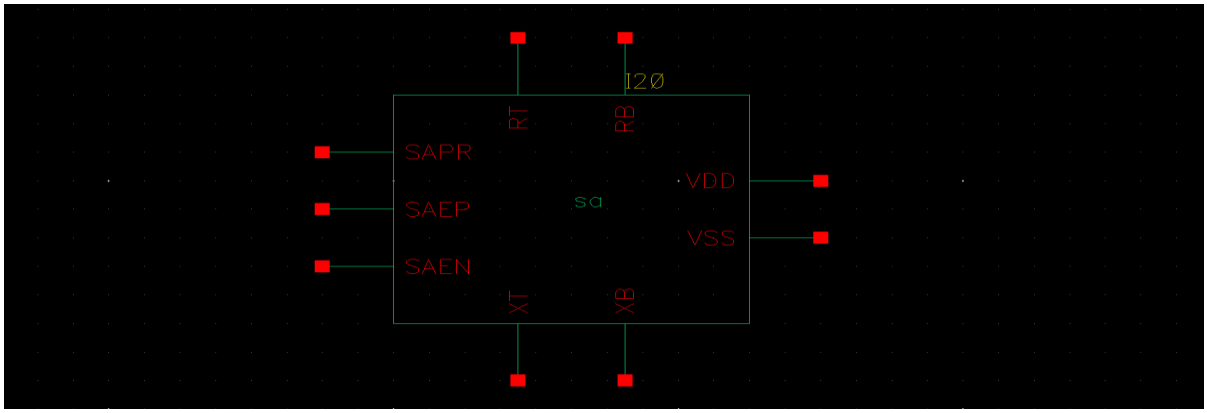


Figure 4. 16.*Sense Amplifier Symbol*

After creating the schematic and verifying the circuit's operation through simulation, the layout of the circuit is created. Figure 4.17 illustrates the layout result of the Sense Amplifier. The metal 1 is used for horizontal connections while metal 2 is used for vertical connection to avoid congestion. This layout is fully verified by LVS and DRC check using Assura LVS and DRC Check.

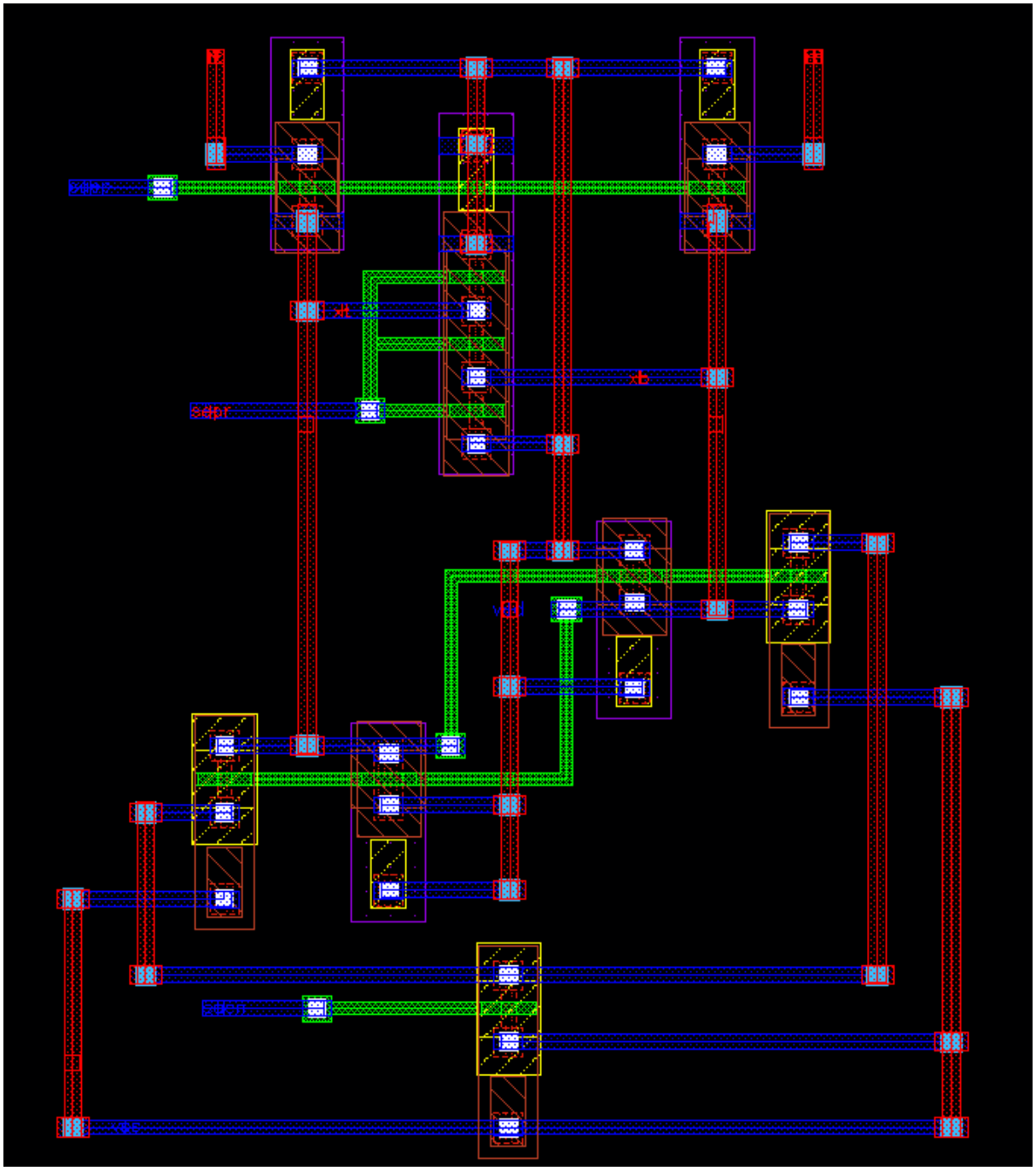


Figure 4. 17.*Sense Amplifier Layout*

4.2.4. Write Driver

The Write Driver circuit is made of two large inverters. A normal inverter has pullup's size of 240nm and pulldown's size of 240nm. When the inverters used in Write Driver circuit has the size of pullup transistors of 360nm and pulldown transistors of

1.8um. The Write Driver is used for pulling down the bitlines to write data to the cells. Hence, the pulldown transistors must be much stronger. The schematic of the Write Driver is given in Figure 4.18.

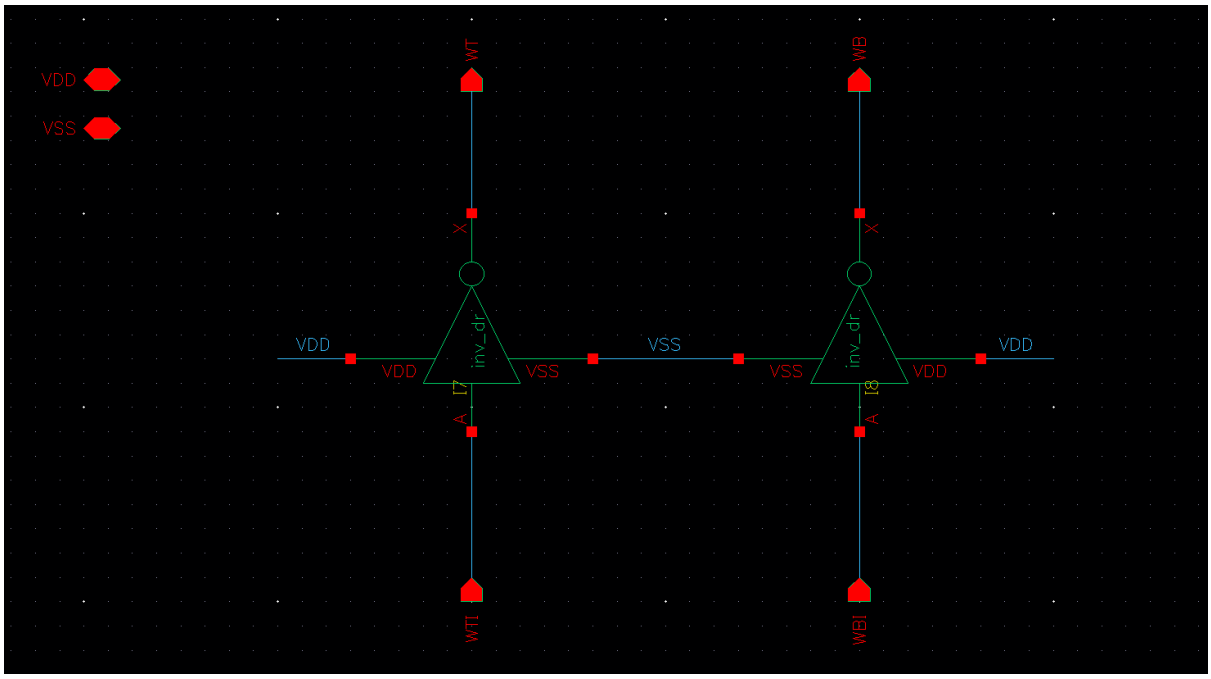


Figure 4. 18.*Write Driver Schematic*

From the schematic, the write driver's symbol which contains all input and outputs is created and is shown in Figure 4.19. The cellview is named wrdrvr and has 6 pins:

- Write bitlines pins: WT and WB
- Write data input pins: WTI and WBI
- Power pins: VDD and VSS

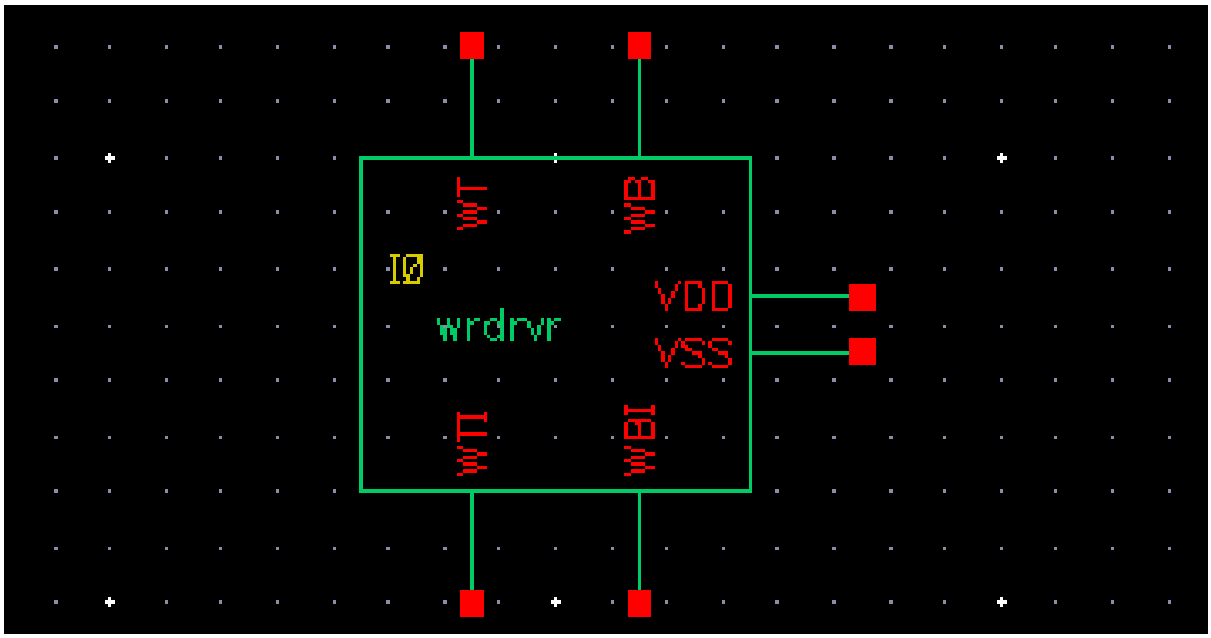


Figure 4. 19.*Write Driver Symbol*

Same as other circuit, after creating the schematic and verifying the circuit's operation through simulation, the layout is created. The layout is formed by layouts of 2 inverters with adjusted size. Figure 4.20 illustrates the layout result of the Write Driver. The metal 1 is used for horizontal connections while metal 2 is used for vertical connection to avoid congestion. This layout is fully verified by LVS and DRC check using Assura LVS and DRC Check.

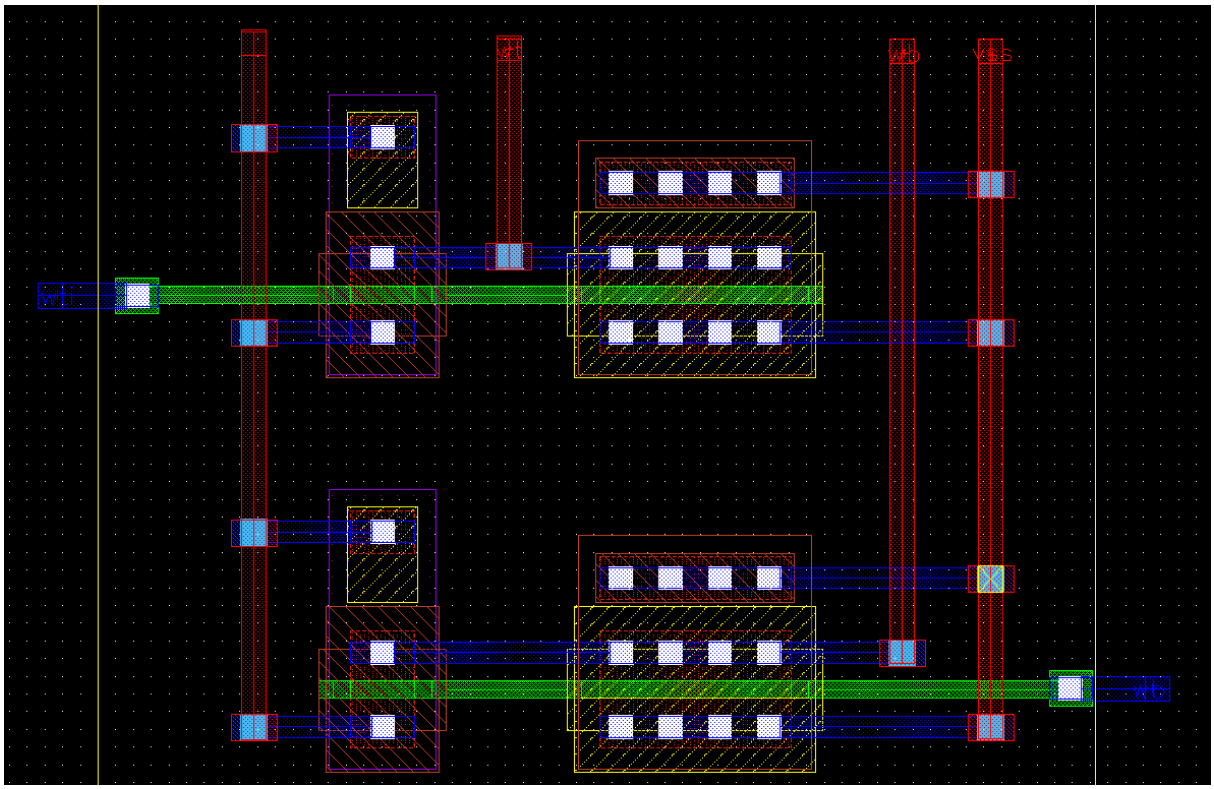


Figure 4. 20.*Write Driver Layout*

4.2.5. Data Input Latch

From the schematic diagram presented in Figure 3.10, the Data Input Latch schematic is formed using 2 clock inverters and 3 inverters. Figure 4.21 illustrates the schematic of the cell which is created by Virtuoso Schematic Editor L. The input is the external data input while the clock signal is generated from the control circuit. The output WTI and WBI will carry contrary values in which WBI has the same value as DIN.

From the schematic, the write driver's symbol which contains all input and outputs is created and is shown in Figure 4.22. The cellview is named di_lat and has 6 pins:

- Data clock pin: DCLK
- Data input pin: DIN
- Write data input pins: WTI and WBI
- Power pins: VDD and VSS

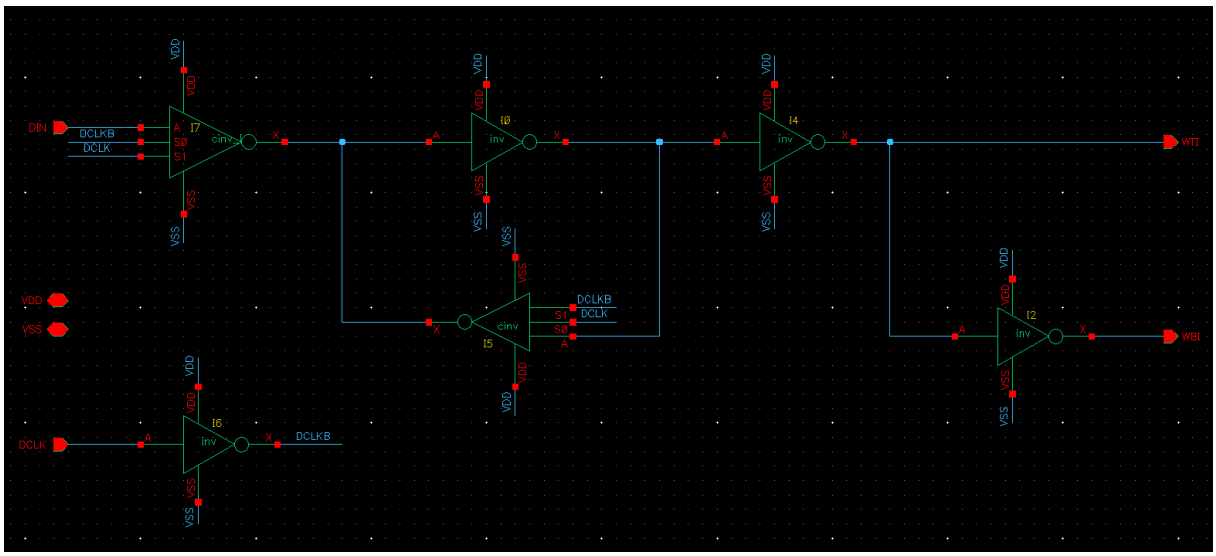


Figure 4. 21.*Data Input Latch Schematic*

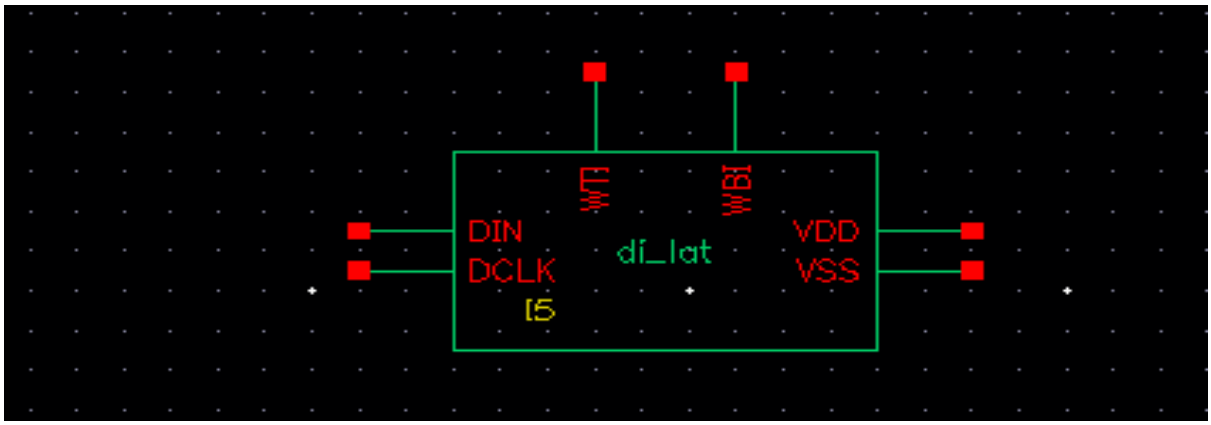


Figure 4. 22.*Data Input Latch Symbol*

After creating the schematic and verifying the circuit's operation through simulation, the layout is created. The layout is formed by layouts of 4 inverters and 2 clock inverters. Figure 4.23 shows the layout result of the Data Input Latch which is fully verified by LVS and DRC check using Assura LVS and DRC Check.

4.2.6. Data Output Latch

From the schematic diagram presented in Figure 3.11, the Data Output Latch is formed. Figure 4.24 illustrates the schematic of the cell which is created by Virtuoso Schematic Editor L. The inputs are the data read from the cell by Sense Amplifier. The output of the circuit is the final output to external – QOUT.

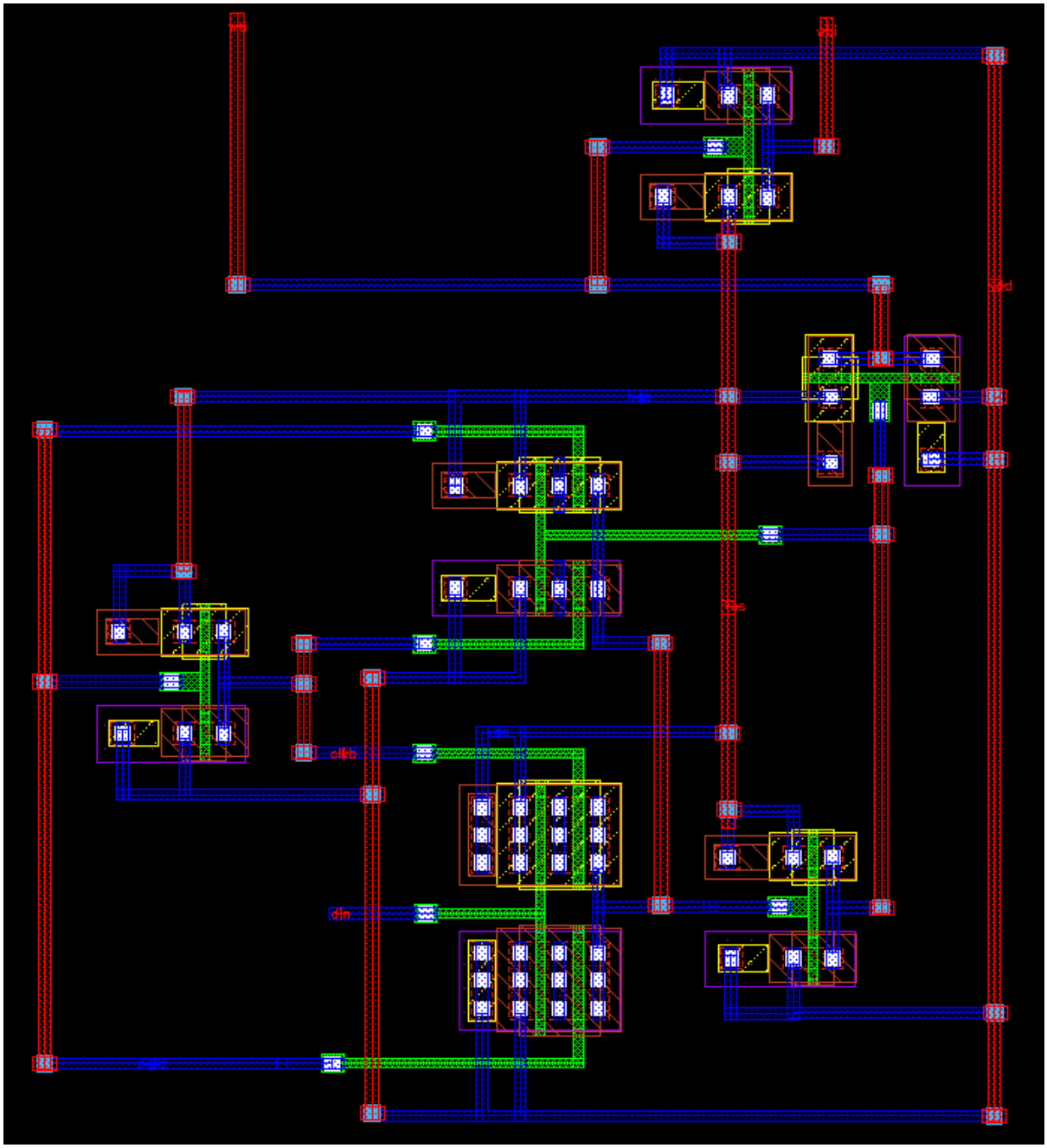


Figure 4. 23.*Data Input Latch Layout*

From the schematic, the write driver's symbol which contains all input and outputs is created and is shown in Figure 4.25. The cellview is named di_lat and has 6 pins:

- Read data pins: QT and QB
- Data output pin: QOUT
- Power pins: VDD and VSS

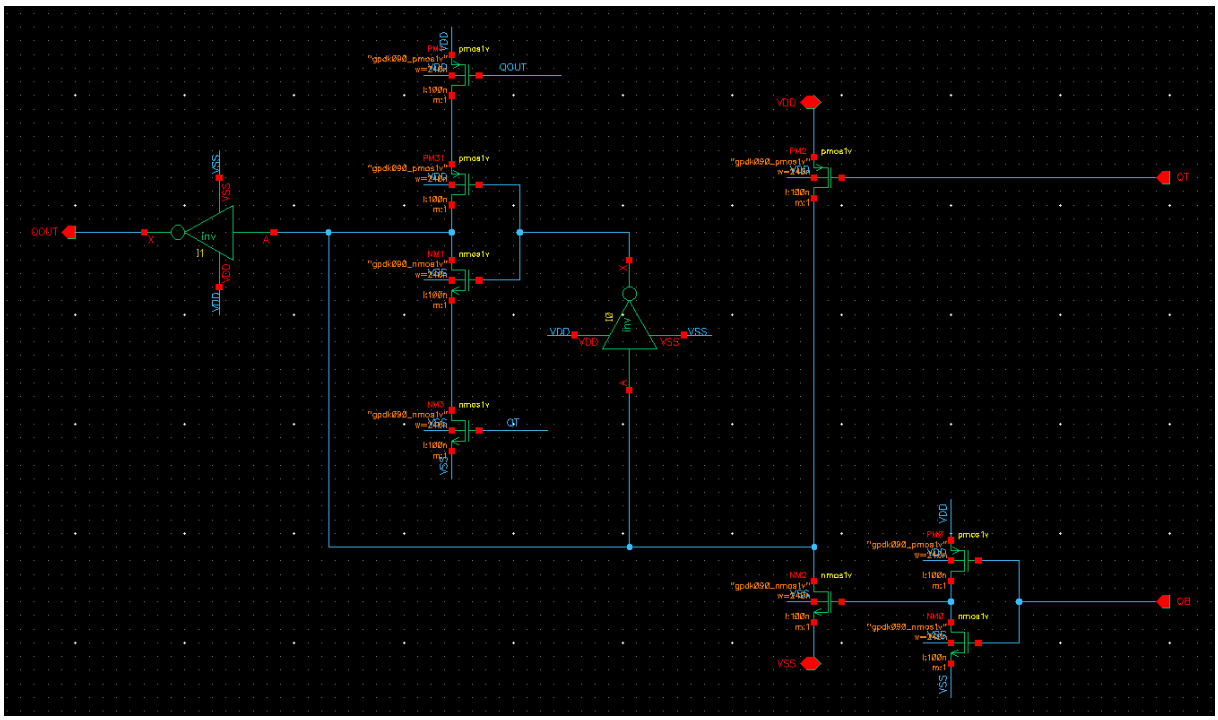


Figure 4. 24.*Data Output Latch Schematic*

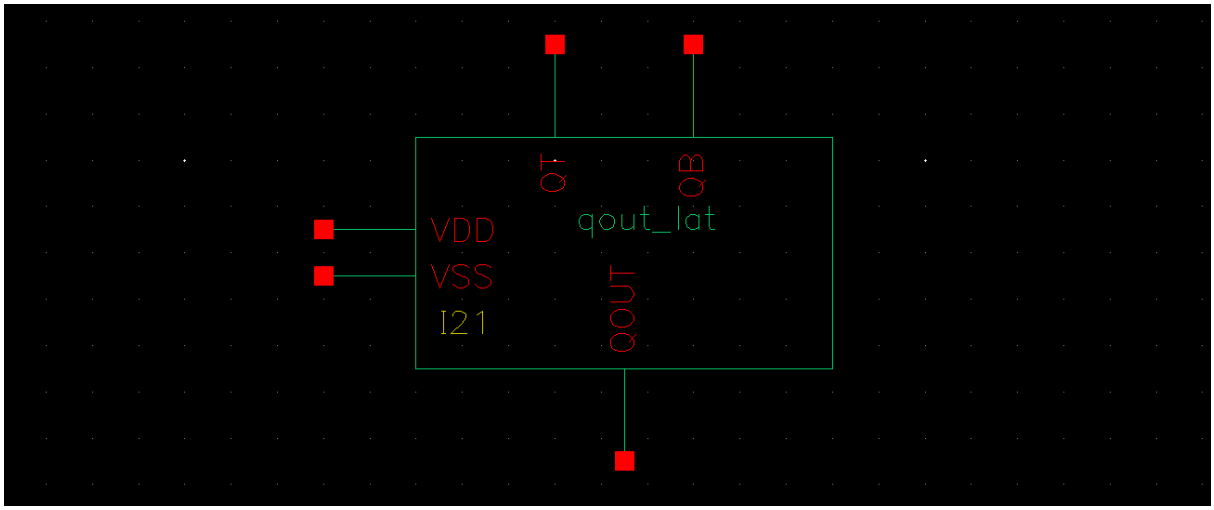


Figure 4. 25.*Data Output Latch Symbol*

After creating the schematic and verifying the circuit's operation through simulation, the layout is created. Figure 4.26 shows the layout result of the Data Output Latch. The metal 1 is used for horizontal connections while metal 2 is used for vertical connection to avoid congestion. This layout is fully verified by LVS and DRC check using Assura LVS and DRC Check.

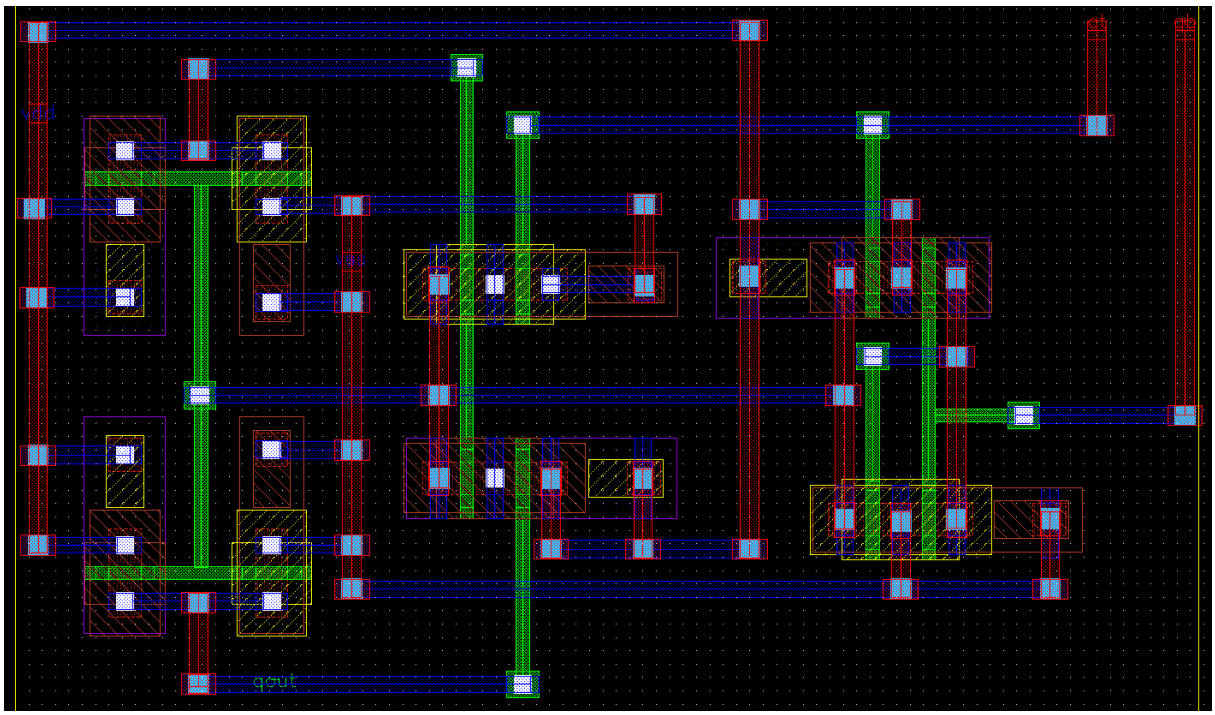


Figure 4. 26.*Data Output Latch Layout*

4.2.7. Read/Write Pass Circuit

In this circuit, 2 PMOS and 2 NMOS transistors are used to formed the schematic. The schematic of the Read/Write Pass Circuit is represented in Figure 4.27. The inputs RPASS and WPASS are inputted to choose between read and write operations. The bitlines BT and BB is connected with the bitlines of the bitcells. Depends on the selected value, the bitlines will be connected with read bitlines, RT and RB or write bitlines, WT and WB.

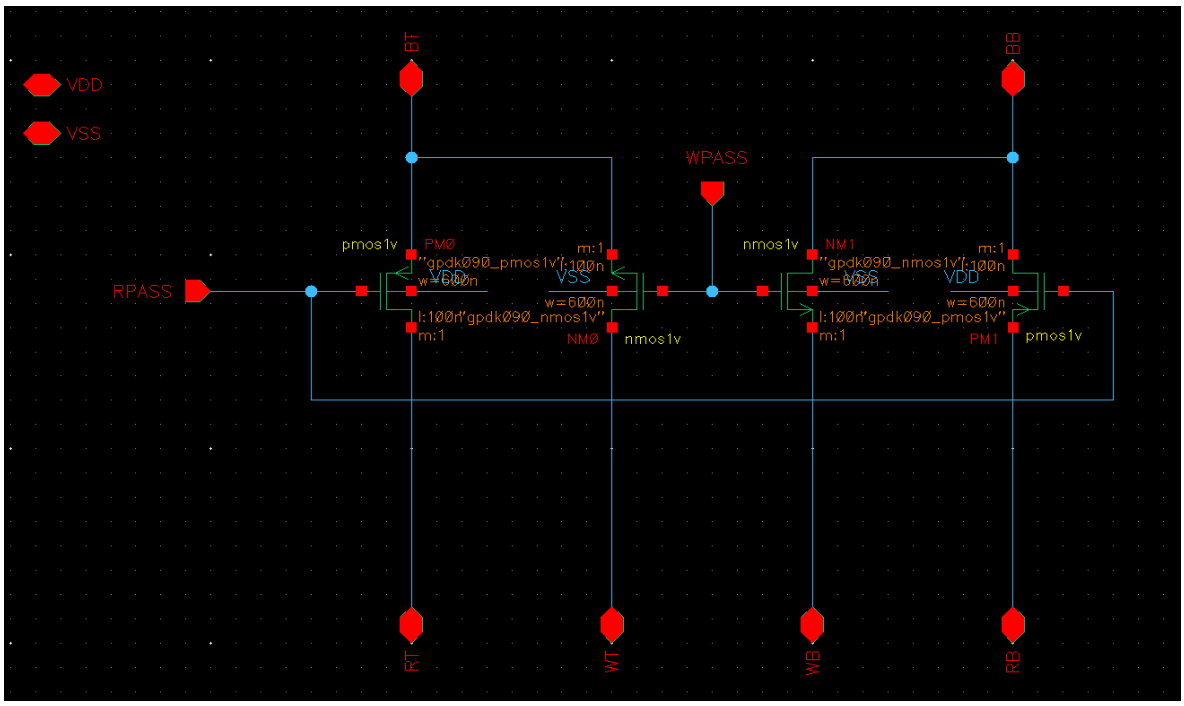


Figure 4. 27.*Read/Write Pass Circuit*

From the schematic, the read/write pass circuit's symbol which contains all input and outputs is created and is shown in Figure 4.28. The cellview is named rw_gating and has 10 pins:

- Read/Write enable pins: RPASS and WPASS
- Bitlines pins: BT and BB
- Read bitlines pins: RT and RB
- Write bitlines pins: WT and WB
- Power pins: VDD and VSS

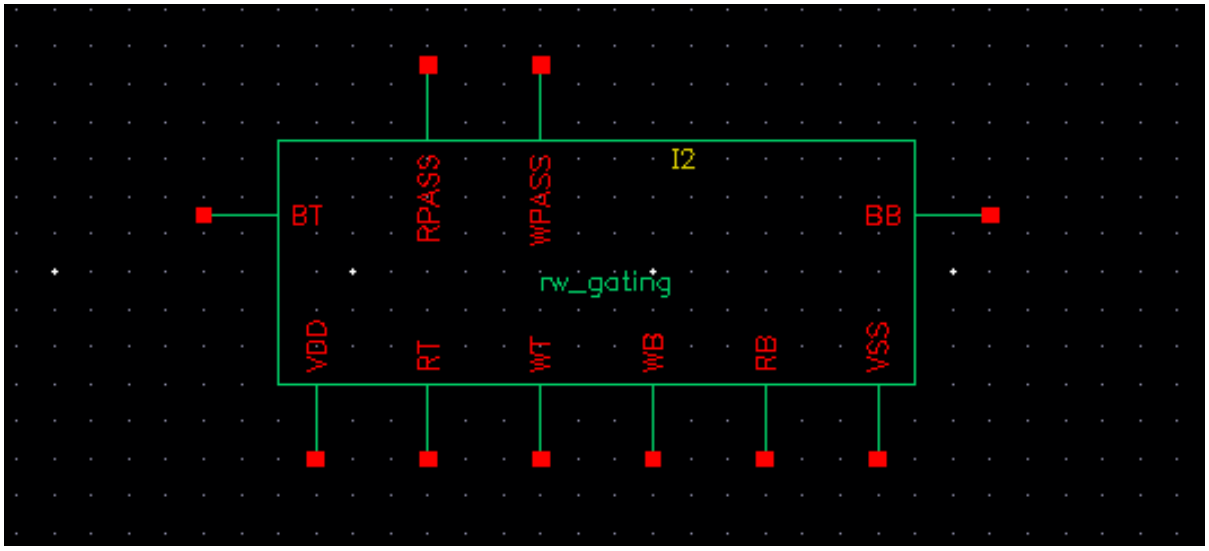


Figure 4. 28.*Read/Write Pass Circuit Symbol*

Same as other circuit, after the schematic is created and its operation is verified through simulation, the layout of the Read/Write Pass Circuit is made using Virtuoso Layout Suite XL. Figure 4.29 illustrates the layout result of the circuit. The metal 1 is used for horizontal connections while metal 2 is used for vertical connection to avoid congestion. The layout circuit is fully verified by LVS and DRC check using Assura LVS and DRC Check.

4.2.6. Address Decoder 3 to 8:

Figure 4.30 represents the schematic of Address Decoder with Clock Gating. This decoder consists of a 3 to 8 decoder and 8 AND gates to gate the decode value by WCLK signal. When WCLK is low, the decode values (DEC<0:7>) will be blocked at the AND gates. One wordline from bus WL<0:7> is triggered on only if WCLK is brought high depending on ADR<0:2> inputs. the schematic of the 3 to 8 decoder is given in Figure 4.31. This decoder is made of 8 AND gates and 3 inverters.

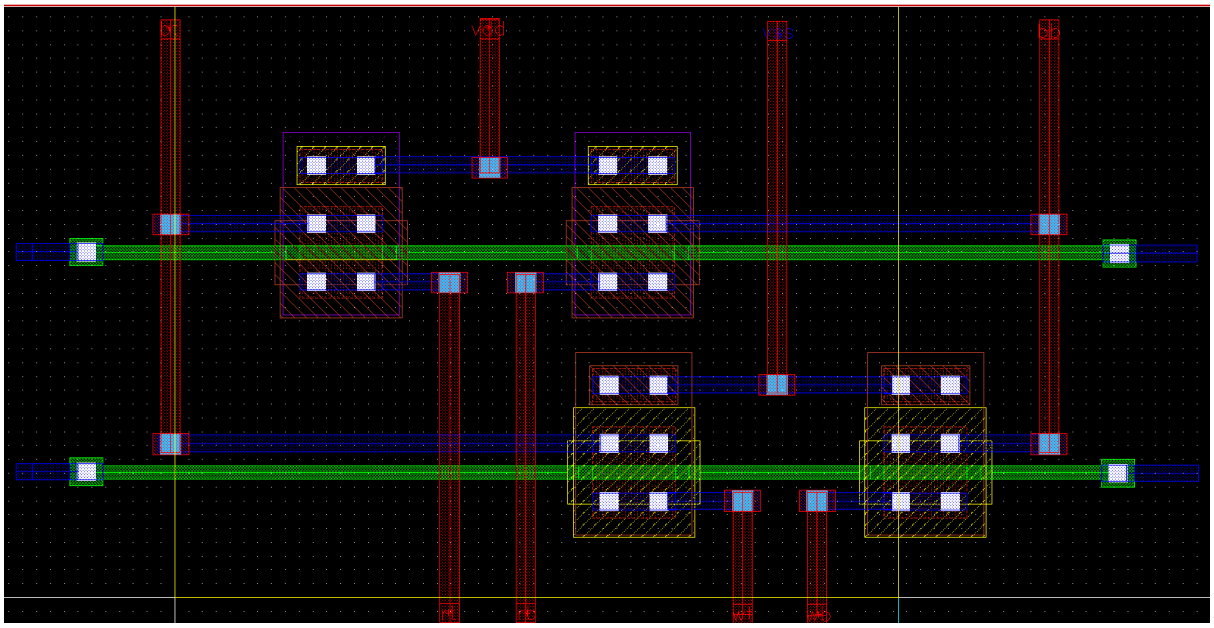


Figure 4. 29.*Read/Write Pass Circuit Layout*

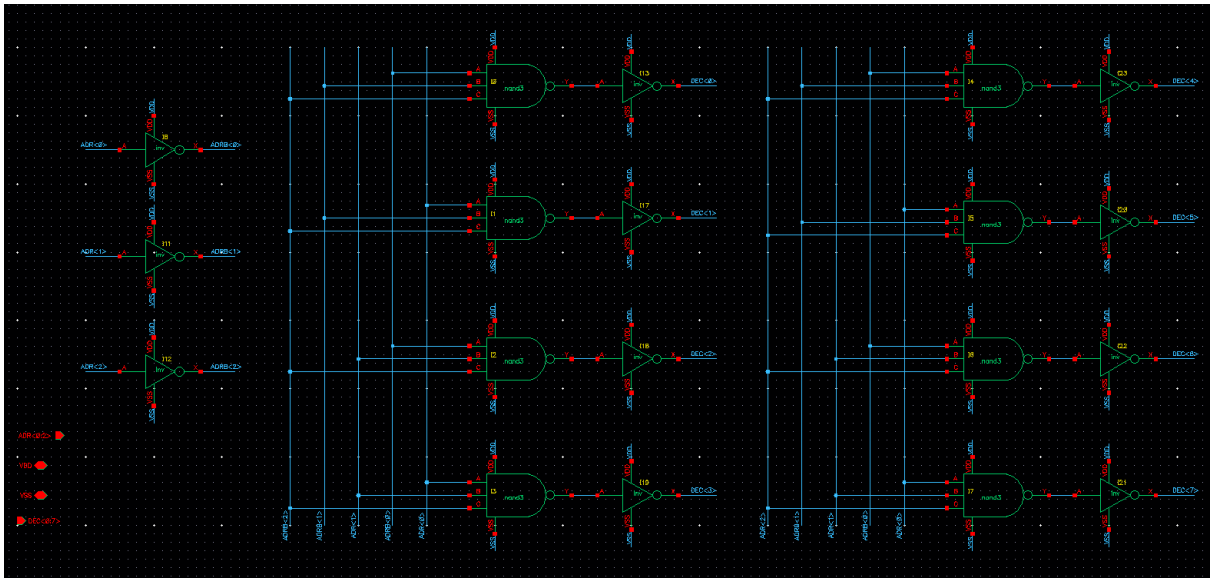


Figure 4. 30.*3 to 8 Decoder Schematic*

From the schematic, the address decoder's symbol which contains all input and outputs is created and is shown in Figure 4.32. The cellview is named adr_wclk and has 5 pins:

- Address bus: ADR<0:2>
- Wordline bus: WL<0:7>
- Wordline clock pin: WCLK
- Power pins: VDD and VSS

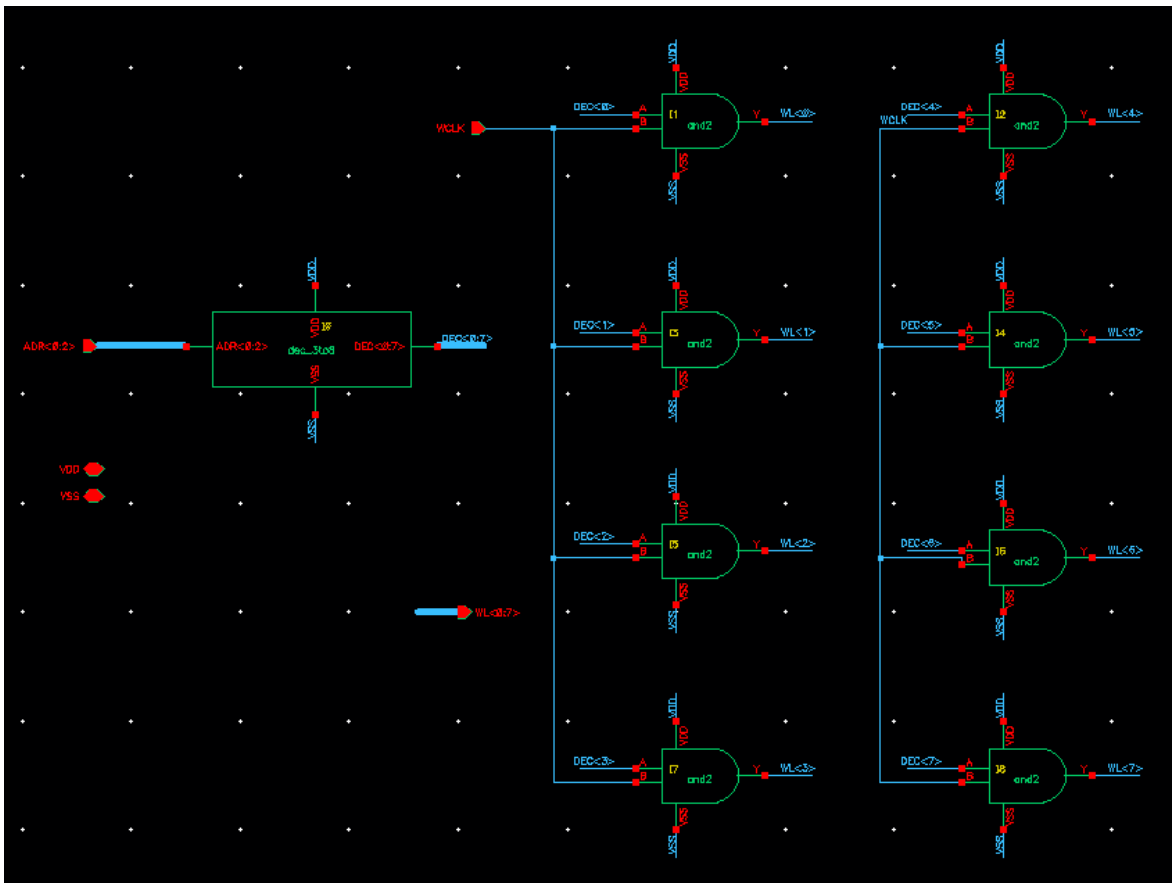


Figure 4. 31.*Address Decoder with Clock Gating*

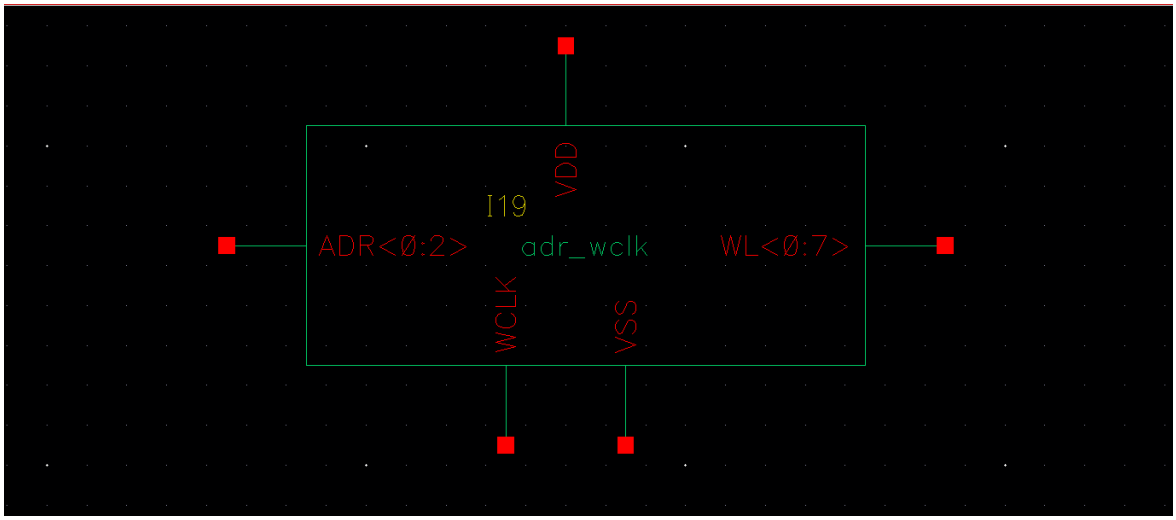


Figure 4. 32.*Address Decoder Symbol*

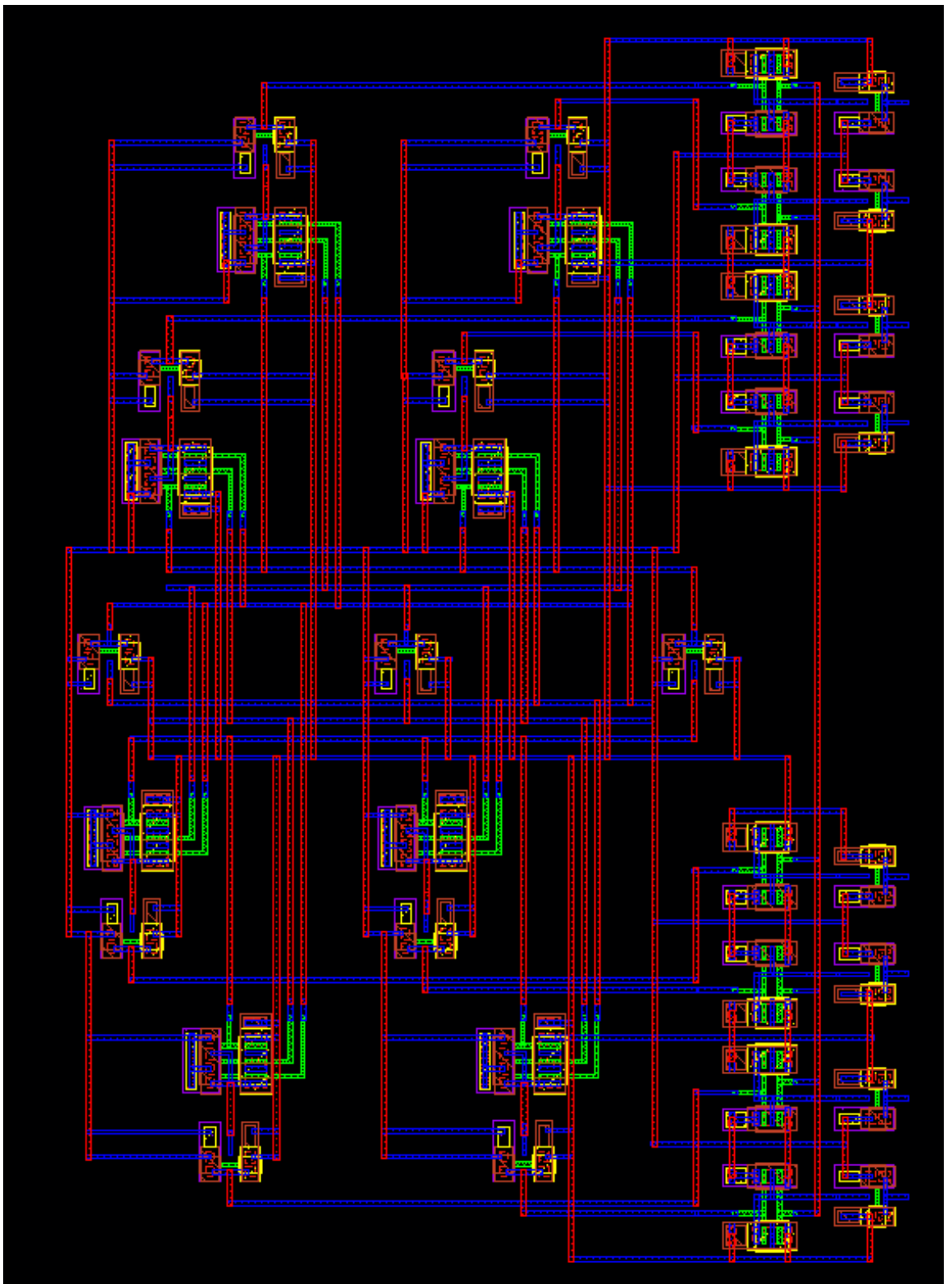


Figure 4. 33.*Address Decoder Layout*

4.4. SIMULATION RESULTS AND MESUREMENTS OF 8X8 SRAM MEMORY.

For validating the operations of the proposed 8x8 SRAM design, test patterns are inputted in. Test patterns are generated including 4 inputs: CLK, WE, DIN<0:7> and ADR<0:2>. The outputs are observed through QOUT<0:7> for read operation and through RT<0:7>, RB<0:7> (which are the data nodes of bitcells) for write operation.

Figure 4.34 represents the simulation results of the proposed SRAM in different read and write operation test cases. The simulation result shows the waveforms of the inputs and outputs of the memory as well as its intern signals in four cases. In each clock cycle, one read or write operation will be performed depeding on the input signal WE. There are four clock cycles which perform four testcases accordingly. The same address is used for these testcases which are ADR<0:7> = 00000000. For better observation, the address inputs and the wordline signal at that position (WL<0>) are hidden in the result waveform. Also, only the first bitcell of the row are obeserved for better view.

- **Cycle 1:** Write ‘0’ into bitcells which has previous value of ‘1’ at address ‘0’.
- **Cycle 2:** Read ‘0’ from bitcells which is previously written ‘0’ at address ‘0’.
- **Cycle 3:** Write ‘1’ into bitcells which has previous value of ‘0’ at address ‘0’.
- **Cycle 4:** Read ‘1’ from bitcells which is previously written ‘1’ at address ‘0’.

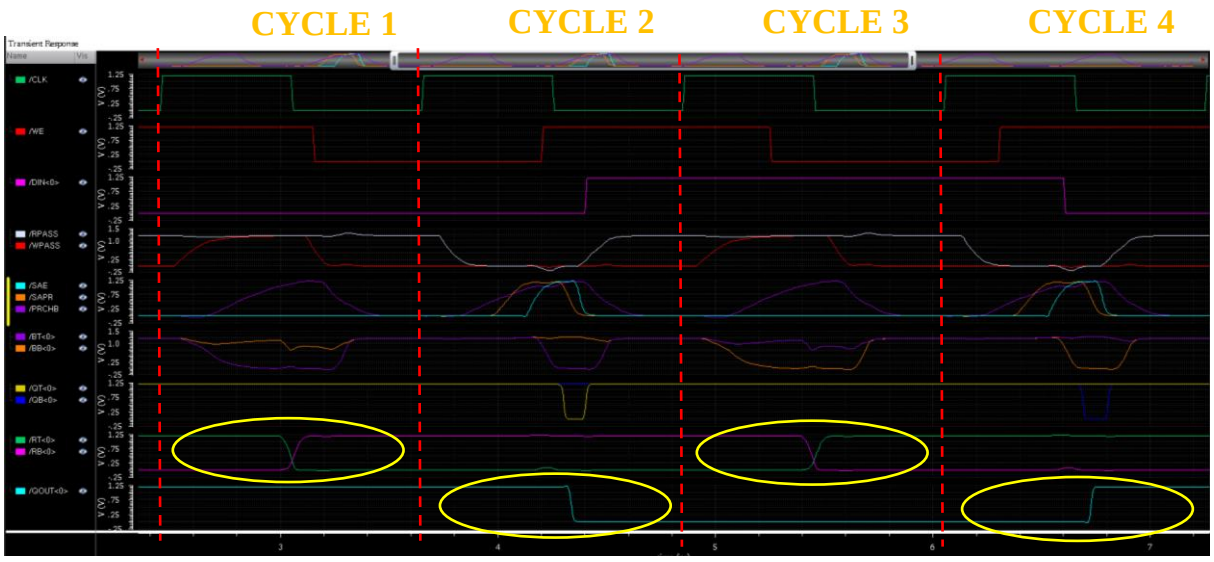


Figure 4. 34.*Test Cases For Read And Write Operations*

In write operations, the WE, DIN<0:7>, ADR<0:7> must be inputted before clock rises for setup time. The PRCHB signal is kept low in low phase of clock to keep BT<0> and BB<0> at VDD level before the write operation. To enable the write function, WE signal must be set to ‘1’. With WE = ‘1’, the Sense Amplifier is kept unactive. When clock rises, the precharge is turned off by rising PRCHB, allowing the BT<0> and BB<0> to discharge. At the same time, the WCLK signal allows the decoded address to

go in and trigger WL<0>. The WPASS signal is also turned on, allowing the new input data to be inserted in through BT<0> and BB<0>.

- In case of writing ‘0’ to the bitcell (cycle 1), the input data DIN<0> is ‘0’. With DIN<0> = ‘0’, BT<0> will quickly be pulled down as can be seen from the figure. This process will flip the value stored in the bitcell, RT<0> and RB<0>. After this write operation, the value of the bitcell changes from RT<0> = ‘1’ to RT<0> = ‘0’. Hence, the write ‘0’ operation is finished successfully.

- Alternately, when ‘1’ is written to the bitcell (cycle 3) with the input data DIN<0> = ‘1’. For this case, the bitline bar BB<0> is pulled down instead of BT<0>. The value of RT<0> and RB<0> is flipped once again. After this write operation, the value of the bitcell changes from RT<0> = ‘0’ to RT<0> = ‘1’. Hence, data ‘1’ is successfully written into the bitcell.

For read operation, the inputs WE and ADR<0:7> must also be inputted before clock rises for setup time. Same as write operation, the PRCHB signal is kept low in low phase of clock to keep BT<0> and BB<0> at VDD level before the read operation. For triggering the read operation, write enable signal (WE) must be set to ‘0’. When clock rises, the precharge is turned off by rising PRCHB, allowing the BT<0> and BB<0> to discharge. Meanwhile, the RPASS is pulled low for connecting the real bitlines BT<0> and BB<0> to the Sense Amplifier. With high pulse clock and WE = ‘0’, the signal SAPR is brought high for turning off Sense Amplifier Precharge, which allows the sensing bitlines to mirror the behavior of the actual bitlines. When the voltage difference between the bitlines is sufficient, the Sense Amplifier enable signal SAE is switched on, allowing the values of the bitcell to be outputted.

- In case of reading ‘0’ from the bitcell (cycle 2), after the precharge is turned off by switched on PRCHB, the bitlines start to discharge. The value stored in the bitcell (RT<0>) at this point is ‘0’ so BT<0> will be discharged. After that, the Sense Amplifier Precharge is also turned off by bringing SAPR to VDD level, which allows the voltage difference between the bitlines BT<0> and BB<0> to be mirrored. As can be seen from the figure, after the voltage of BT<0> drops to a certain amount (typically more than 20mV), the SAE signal is switched on and the values of bitlines are read out to QT<0> and QB<0> with QT<0> = ‘0’ and QB<0> = ‘1’. Then, QT<0> is latched out to output, flipping QOUT<0> value and ‘0’ can be observed at QOUT<0>. Hence, the read operation is performed successfully.

- Alternately, in case of read operation performed on bitcell which stores the value of ‘1’, BB<0> will be discharged after signal PRCHB being brought high and switch off the Precharge. Then, SAPR is also pulled up to turn off the Sense Amplifier Precharge. The voltage difference is sensed by the Sense Amplifier and the value stored in the bitcell will be read through QT<0> and QB<0>. In this case, BB<0> is discharged so QB<0> value is flipped to ‘0’ when SAE triggered. Hence, QT<0> is flipped to ‘1’ and

latched out to QOUT<0>. As can be seen from the figure, ‘1’ is observed at QOUT<0> after SAE being pulled up. This means that the read operation is finished successfully.

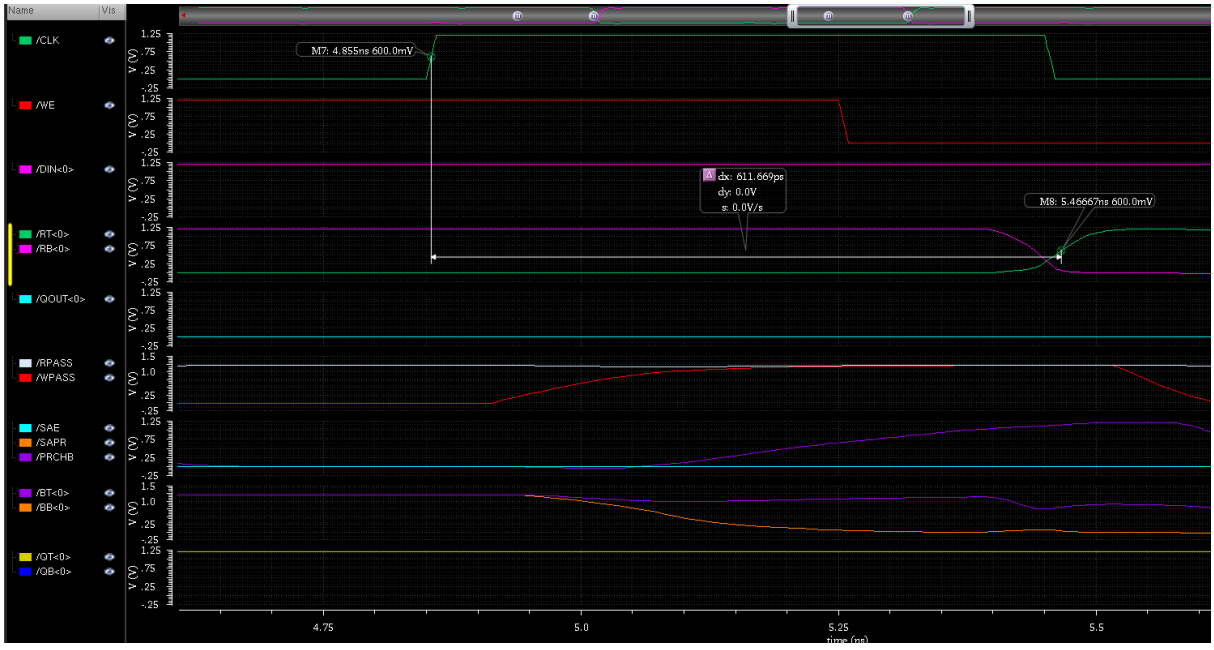
After the read/write operations are verified by simulations, the delay of read/write process is measured in all cases. For write operation, the delay is measured from the beginning of the operation to the point where the value stored in the bitcell flipped. With read operation, the delay is measured from the beginning of the operation to the point where the value stored in the bitcell observed at QOUT.

Figure 4.35a shows the delay time to write value ‘1’ into a bitcell. As mentioned earlier, the delay is measured from clock signal rises to 50% of VDD to RT<0> rises to 50% of VDD. Meanwhile, the delay time to write bit ‘0’ into a bitcell is described in Figure 4.35b. In this case, the delay is measured from clock signal rises to 50% of VDD to RT<0> falls to 50% of VDD. The delay is calculated by equation 4.1:

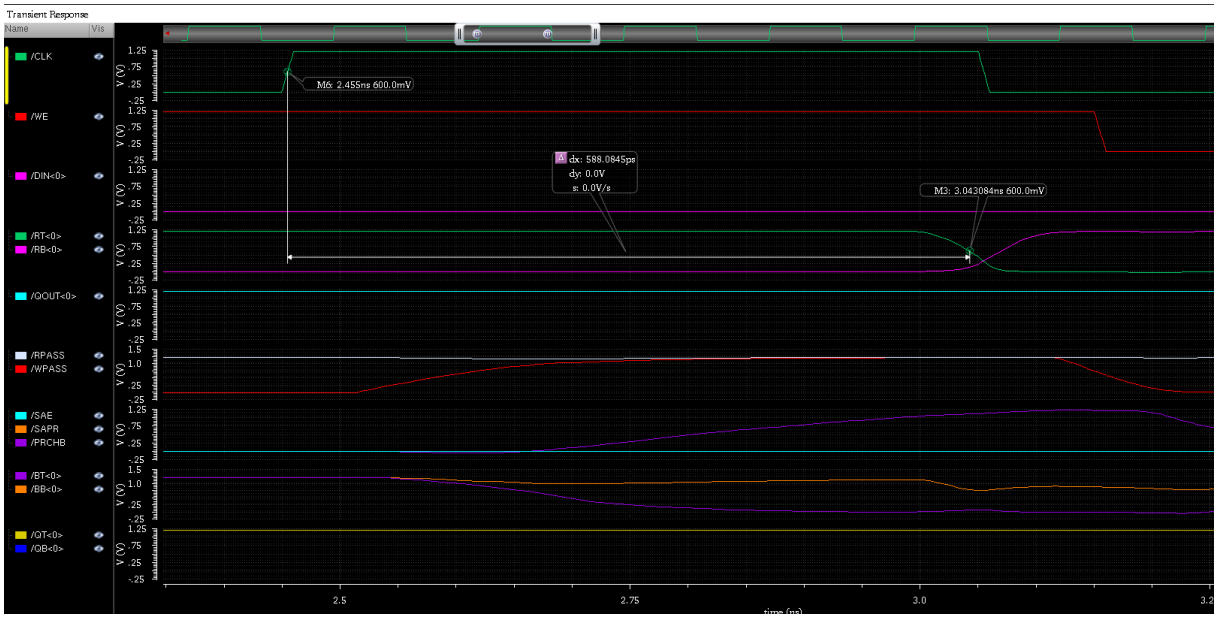
$$T_{PD} = (T_{PDLH}/2) + (T_{PDHL}/2) \qquad (4.1)$$

$$T_{PD} = (611.669ps/2) + (588.0845ps/2) = 599.87675ps$$

Hence, it takes about 599.87675ps to write new data to the farthest row of bitcells.



(a)



(b)

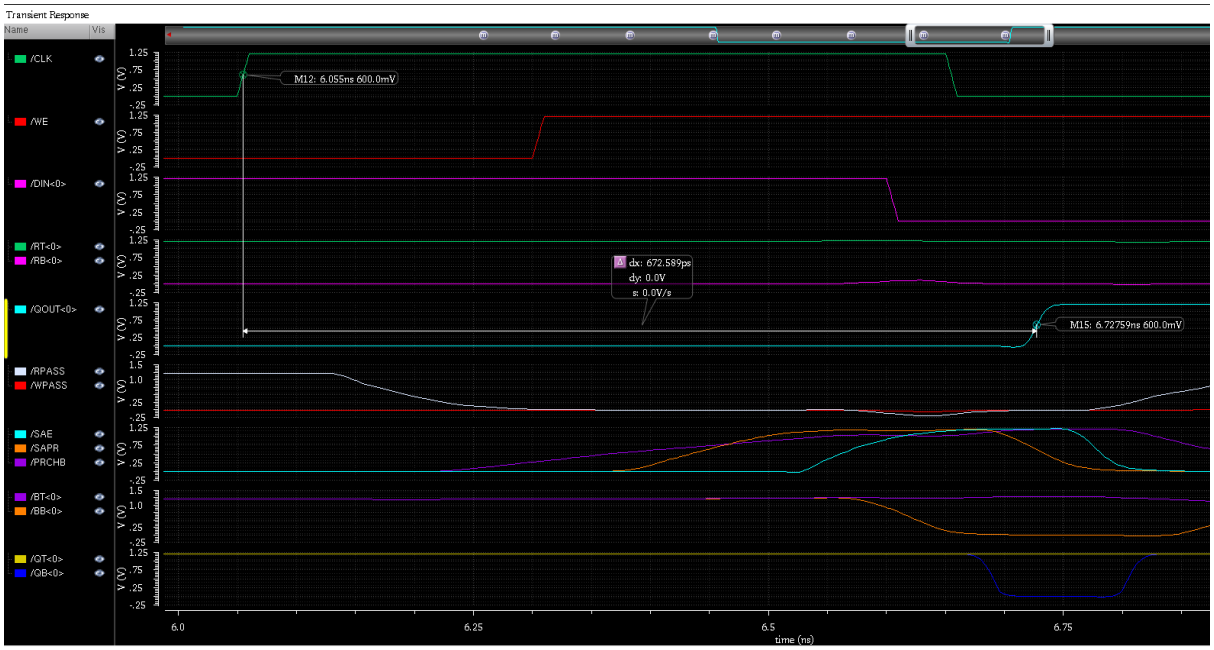
Figure 4. 35. (a) *Write '1' Delay Time* ***(b)*** *Write '0' Delay Time*

Figure 4.36a shows the access time of reading ‘1’ from a bitcell. The delay is measured from clock signal rises to 50% of VDD to QOUT<0> rises to 50% of VDD. Meanwhile, the access time to read bit ‘0’ from a bitcell is described in Figure 4.36b. In this case, the delay is measured from clock signal rises to 50% of VDD to QOUT<0> falls to 50% of VDD. The delay is calculated by equation 4.2:

$$T_{PD} = (T_{PDLH}/2) + (T_{PDHL}/2) \qquad (4.\ 2)$$

$$T_{PD} = (672.589ps/2) + (683.3052ps/2) = 677.9471ps$$

Hence, it takes about 677.9471ps to read the stored data from the farthest row of bitcells.

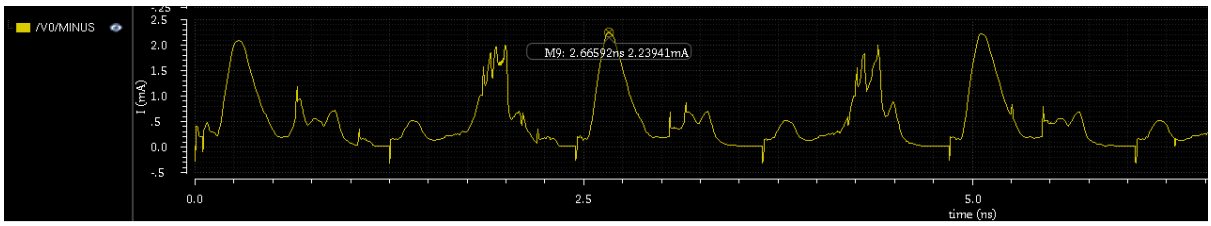


(a)

(b)

Figure 4. 36. (a) *Read '1' Delay Time* ***(b)*** *Read '0' Delay Time*

After determining the delay of each read/write operation cases, the power of the 8x8 SRAM memory is evaluated. Figure 4.37a represents the waveform of current flows through the memory in multiple read/write operations. Using the equation given in equation (2.2) and the current measured in Figure 4.37a, the average power is calculated.



(a)

From Figure 4.37a, the peak current of the memory is measured to be 2.2394mA. The average power consumed by the 8x8 SRAM memory is shown in Figure 4.37b.

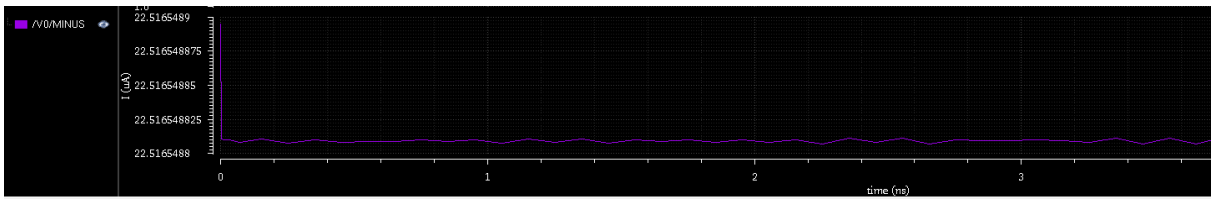
$$P_{dynamic} = 625.7\ nW$$

(b)

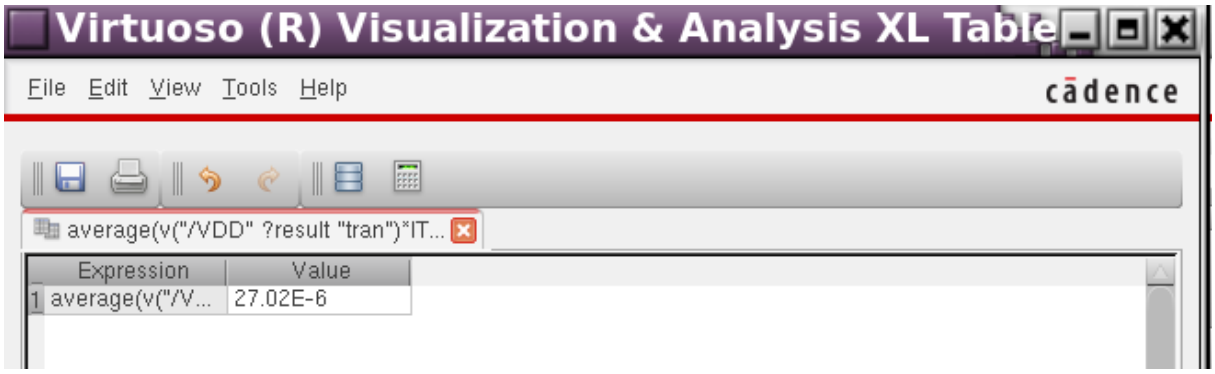
Figure 4. 37. (a) *Dynamic Power Waveform* ***(b)*** *Average Power Result*

Beside the dynamic power, the static power of the memory is also evaluated and given in Figure 4.38a. The static power is measured when the memory is in its idle state. To create such state, the inputs are kept stable and the power is observed. The average static power is also calculated by equation (2.2) and the result is shown in Figure 4.38b.

$$P_{static} = 27.02\ nW$$



(a)



(b)

Figure 4. 38. (a) *Static Power Waveform* ***(b)*** *Static Power Result*

The total power consumption of the 8x8 SRAM is then calculated by equation (2.3):

$$P_{total} = P_{dynamic} + P_{static} = 625.7 + 27.02 = 652.72\ nW \qquad (4.\ 3)$$

After determining the power dissipation of the memory in equation 4.3, the SNM of the bitcell is further studied. The SNM test circuit is built based on the description in Chapter 2. The test circuit is given in Figure 4.39.

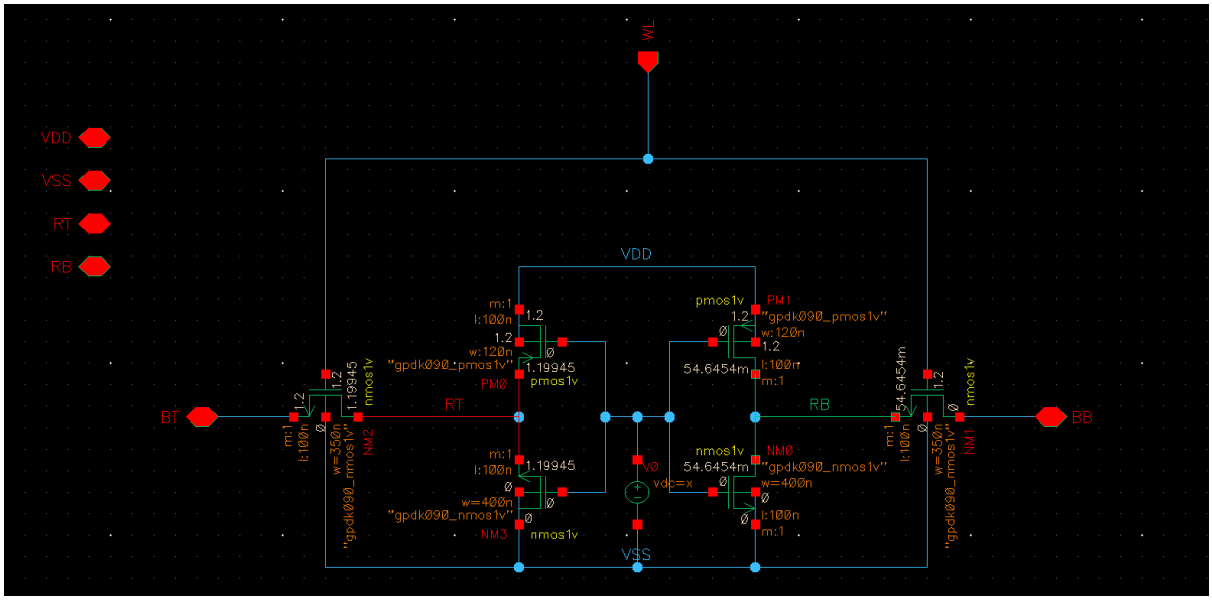
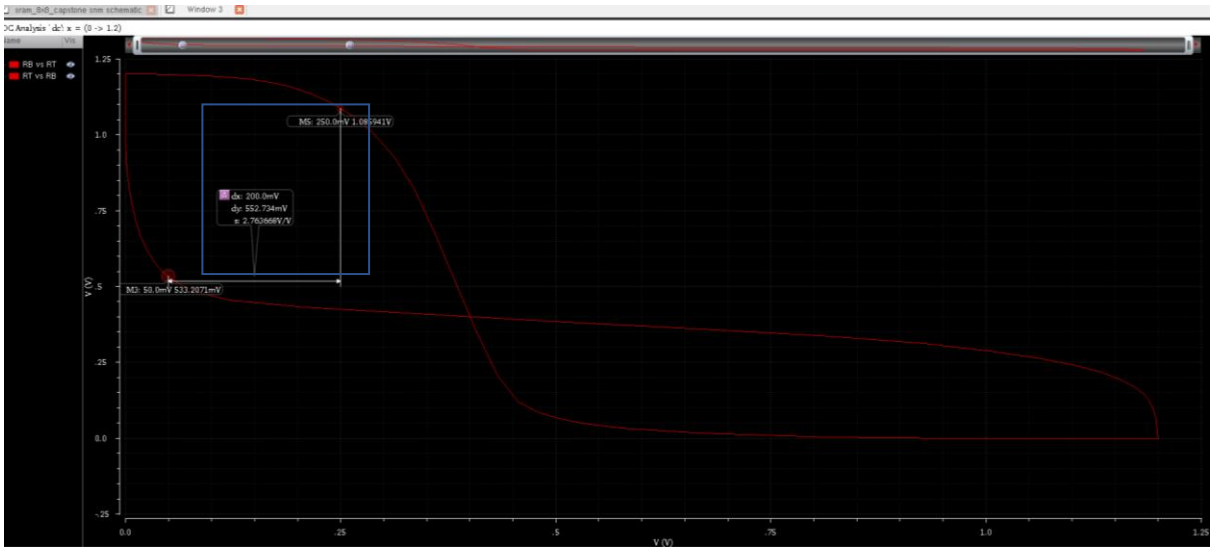


Figure 4. 39. *SNM Test Circuit*

With the read margin, WL, BT and BB is set to be at VDD level. The replicate noise source ‘vdc’ is varied from 0 to VDD. The voltage transfer characteristics of RT vs RB and RB vs RT is plotted to create the butterfly curve which is given in Figure 4.40a. The margin is determined by the side of the largest fitted square between the two halves. The read margin of the proposed bitcell is 200mV.



(a)

With the write margin, WL and BT is set to be at VDD level while BB is tied to GND. The replicate noise source ‘vdc’ is varied from 0 to VDD. The voltage transfer characteristics of RT and RB is plotted and given in Figure 4.40b. The margin is determined by the side of the largest fitted square between the two halves. The read margin of the proposed bitcell is 895.3617mV.

(b)

Figure 4. 40. (a) *Read Noise Margin* ***(b)*** *Write Noise Margin*

Hence, the Static Noise Margin of both read and write operation has been measured. The large margin value represents a stable memory cell, which is not fragile to noise. There is trade-off between a good noise margin with the cell’s area. The margin can be varied by changing the bitcell’s transistors ratio, depending on the purpose of the memory.

CHAPTER 5: CONCLUSIONS AND FUTURE WORKS

5.1. CONCLUSIONS

In this paper, the proposed SRAM design is implemented and function-verified using the technology library TSMC gpdk90nm and the EDA tool Cadence Virtuoso. The proposed SRAM memory is 64 bits arranged into 8 words, with 8 bits in each word. The design features 6T memory arrays, which are widely used in modern SRAM designs because of their simplicity, symmetrical structure, and compatibility with peripherals. The desired word for a read or write operation is determined using the 3 to 8 clock-gating Address Decoder. The bitcell array is controlled by the peripherals, which assist in performing read and write processes. The peripherals consist of Precharge, Write Driver, Sense Amplifier, Read/Write Pass and I/O Latches. Various test cases are implemented to thoroughly investigate the memory operation, as demonstrated by the timing diagram. Also, Read and Write Noise Margin Analysis which assures the stability of the memory cell, was performed. The memory's power dissipation, which encompasses both static and dynamic power, is also measured and calculated. The memory layout is then formed, and each component has been validated using DRC and LVS checks by Assura.

- **Strengths**

Instead of utilizing a single set of peripherals for each bitcell, the design uses one set of peripherals that are shared by eight bitcells in the same column. As a result, the memory size has been substantially reduced. The Voltage Latch Sense Amplifier is used to detect the voltage differences at an extremely small offset. On top of that, with fewer peripherals, the load on peripherals has been lowered. This plays a role in minimizing the design's delay time and power dissipation. Plus, the layout has been optimized in terms of area, symmetry, and metal rule.

- **Weaknesses**

Since the design uses a single set of peripherals shared between the bitcells in one column, the path from bitcells to peripherals is longer. Without any performance boost method, read and write operations time is not optimized. Hence, the delay time is quite large compared to other small-capacity 90 nm-based designs.

Another downfall of the proposed system is the large power dissipation. In spite of using fewer peripherals and latched IOs, the power consumption of this design remains inefficient.

5.2. FUTURE WORKS

Because of the limited time I have to accomplish this capstone project, many high-performance, low-power consumption schemes for the memory cannot be evaluated. In my upcoming study, I plan to concentrate on enhancing memory performance while

reducing power consumption in terms of functional power and leakage currents. The objectives of these advancements can be achieved by incorporating various techniques that are currently used in SRAM design.

The first bottleneck of the proposed design is the performance of read and write operations. This performance issue can be properly solved by using a technique called self-timing. The self-timing approach can be used as an automatic tracking feature that aids in the design's ability to keep track of its own operation and determine when to switch on and off the peripherals. This can dramatically decrease both the delay time and power consumption. I also plan on developing SRAM memory with a larger capacity. With a high number of bitcells, alternative SRAM structures such as Bank and Center Decode can be used to enhance the performance of the bitcells that are located further away from the control unit. For the same purpose, buffers and re-buffers can also be applied to the design to boost the strength of internal signals.

The other downfall of this design is the large amount of power consumed in both the active and inactive states of the memory. Hence, many techniques will be incorporated into my future design to improve the power consumption of the memory. A technique called Power Gating will be used to prevent leakage current in the unactive state. Along with managing the static power dissipation, the dynamic power can also be lowered through Dual Rail Power method. This technique uses two different voltage sources for the memory array and the peripherals. This can help reduce the amount of power required during read and write operations.

REFERENCES

[1] S. I. A. (SIA), International Technology Roadmap for Semiconductors

(ITRS), 2012.

[2] Lim W, Chin H C, Cheng L S and Tan MLP 2014 Performance evaluation of 14nm

FinFET-based 6T SRAM cell functionality for DC and transient circuit analysis

Journal of Nanomaterials 820763.

[3] N. D. T. P. V. Khoa, "Phân tích hiệu năng của các thiết kế SRAM trên công nghệ TSMC 90nm CMOS," *Tạp chí khoa học và công nghệ - Đại học Đà Nẵng,* vol. 20, no. 10/01/2022, pp. 26-31, 2021.

[4] L.B.S.B.T.T. Võ Thanh Trí, "Thiết kế bộ nhớ SRAM 32KB kết hợp kỹ thuật dự trữ hàng và cột," *Tạp chí khoa học và công nghệ đại học Đà Nẵng,* vol. 5, no. 05/05/2015, pp. 161-165, 2015.

[5] T. N. Phu Phu, D. P. Gia Han, N. C. Luong, and N. van Cuong, "Design A Synchronous Single-port Sram 1024x32xMUX4 Using 28NM Technology*," Int. J. Comput. Digit. Syst.,* vol. 10, no. 1, 2021, doi: 10.12785/ijcds/100110.

[6] M. Agarwal, T. Tevatia, "Design & Implementation Of Self Time Dummy Replica Technique In 128x128 Low Voltage SRAM*" Int. J. of Novel Research And Development*, vol. 2, no. 4, 2017.

[7] Sreerama Reddy G M, P Chandrasekhara Reddy, "Design and Implementation of 8K-bits Low Power SRAM in 180nm Technology" *Int. Multi Conf. of Eng. and Comp. Sci.*, vol. 2, 2009.

[8] Garima Jain, "Design and Simulation Low Power SRAM Circuits" *International Journal for Scientific Research & Development*, vol. 1, 2013.

[9] commons.wikimedia.org. (2014). *File: SRAM Cell Inverter Loop.png - Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/File:SRAM_Cell_Inverter_Loop.png [Accessed 9 Jun. 2023].

[10] Kunal Dhawan. (n.d.). *SRAM Design with Differential Voltage Sense Amplifier*. [online] Available at: https://kunal-dhawan.weebly.com/sram-design-with-differential-voltage-sense-amplifier.html [Accessed 9 Jun. 2023].

[11] N. H. E. W. a. D. M. Harris, CMOS VLSI Design A Circuits and Systems Perspective, United States of America: Edwards Brothers, 2011.