

**DESIGN A HIGH PERFORMANCE DUAL PORTED  
1 READ 1 WRITE CMOS SRAM**

**By**

**Yeoh Ee Ee**

**Dissertation submitted to  
UNIVERSITI SAINS MALAYSIA**

**In partial fulfillment of the requirements  
for degree with honors**

**BACHELOR OF SCIENCE (ELECTRONIC ENGINEERING)**

**School of  
Electronic and Electrical  
Engineering of University Science Malaysia**

**MAY 2006**

## **ABSTRACT**

A synchronous dual-ported high speed and low power CMOS SRAM is described. This SRAM is an 8T (transistors) architecture that has 8 transistors in every single memory cell that are capable of performing a read and write operation in one cycle, under the condition that it is not performing the read and write operation in the same decoded address simultaneously. The proposed SRAM has 4Kbit memory capacity with 33 entries and 128 size of entry and was custom designed using 90nm process. The SRAM is operating properly with supply voltage of 1.05V. The targeted operating frequency is at 125MHz and it dissipates a maximum active power of 10.5mW and consumes a maximum standby power of 2.1mW. The targeted current consumption for the SRAM is having a maximum active value of 10mA and a maximum standby current of 2mA. The functionality of the SRAM is guaranteed by running simulations over a wide range of Process, Voltage and Temperature (PVT) corners. Internal race checking for SRAM has been adopted to perform further verification to ensure that there's no failing signal in the SRAM that will cause functional error and excess power consumption in SRAM.

## ABSTRAK

CMOS SRAM yang berfrekuensi tinggi dan berkuasa rendah telah direkabentuk. Bentuk SRAM CMOS tersebut adalah bersifat dua mod yang beroperasi untuk "read" dan "write". SRAM CMOS tersebut mengandungi lapan transistor dalam setiap memori cel dan berupaya untuk menjalankan operasi "read write" dalam satu kitaran dengan syarat tidak menjalankan operasi tersebut dalam lokasi yang sama. SRAM yang direkabentuk adalah 4Kbit memori dengan 33 "entries" dan bersaiz 128 "entry". Di samping itu, SRAM yang direkabentuk beroperasi dalam 1.05V. Frekuensi yang ditetapkan adalah 125MHz dan kuasa maksimum yang didiscas adalah 2.1mW. Arus yang diperlukan adalah 10mA maksimum dan arus untuk "standby" mod adalah 2mA. Kaedah operasi SRAM ditetap kebaikannya dan sifatnya dengan menjalankan simulasi PVT, iaitu pada variasi proses, voltan dan suhu. "Internal race" untuk SRAM diidentifikasi dan dikira betapa kritikal sifat tersebut dalam operasi "read write". Oleh itu, kebaikan rekabentuk SRAM dapat diramalkan.

## ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude to the following parties for their generous help and guidance towards the completion of my thesis report. These people have made my undergraduate studies possible and have given me such a rewarding experience.

First of all, a special thanks and appreciation goes to my supervisor cum lecturer, Mr. Zulfiqar Ali Abdul Aziz, for his tireless dedication, guidance, patience and understanding. He has been supportive and dedicated throughout this FYP execution period on the welfare of his students and he has greatly contributed to the success of this project. Besides that, he is also keen to share his experience and knowledge with his student and this has given me a better insight of the field of IC design.

Thanks to Dr. Tun Zainal Azni Zulkifli, who is a great lecturer and mentor. He has taught me VLSI and analog IC design that has proven to be a great in depth insight in analog IC design. His teachings have given us a good head start on our final year project. I would also like to thank Dr. Tun Zainal Azni Zulkifli for maintaining the Cadence IC Design Tools server. Without his tireless dedication and motivation, our IC design group won't have the chance to use this IC design tools.

Special thanks to Mr. Ng Meng Thai, who is responsible for the success of collaboration of final year project between USM and Intel. His support and guidance has been a good motivation to me.

I would also like to post my up most gratitude to KL Phang, Kevin Lin, TK Goh, CL Ng and SH Tan for their guidance and motivation in Intel. Throughout this program, they have been a great role model of a mentor and a caring friend.

Lastly, I would like to thank my family members for caring and understanding throughout this time. Their patience and invaluable encouragements have given me no setbacks and pressure for all this time.

Yeoh Ee Ee

## TABLE OF CONTENTS

	<b>Page</b>
ABSTRACT.....	ii
ABSTRAK.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES.....	viii
LIST OF TABLES.....	x
<b>CHAPTER 1: INTRODUCTION</b>	
1.1 Objectives and Aims of Project.....	1
1.2 SRAM Overview.....	2
1.2.1 Types of Memory.....	2
1.2.2 Overall Architecture of SRAM Design.....	3
1.2.3 SRAM Operation.....	4
1.3 Summary.....	6
<b>CHAPTER 2: LITERATURE REVIEW</b>	
2.1 Different SRAM Memory Cell Design Topologies.....	7
2.2 Summary.....	10
<b>CHAPTER 3: HIGH PERFORMANCE SRAM DESIGN</b>	
3.1 Design Specifications.....	11
3.2 Top Level System View.....	13
3.2.1 Pin List.....	14
3.2.2 Modes of Operation.....	14
3.3 Summary.....	15
<b>CHAPTER 4: SYSTEM ARCHITECTURE DESIGN</b>	
4.1 SRAM Architecture Design.....	16
4.2 Summary.....	18

## **CHAPTER 5: CUSTOM CIRCUIT DESIGN OF SRAM**

5.1	Memory Cell .....	19
5.2	Memory Array.....	20
5.3	Pre-charge Circuitry / Bit-line Keeper .....	21
5.4	Row Decoders for Read and Write.....	23
5.5	Latches.....	26
	5.5.1 Input Latch.....	26
	5.5.2 Output Latch.....	28
5.6	Controller Block.....	30
5.7	Summary.....	34

## **CHAPTER 6: CIRCUIT IMPLEMENTATION**

6.1	Sub-blocks Integration.....	35
	6.1.1 Memory Cell.....	35
	6.1.2 Row Decoders.....	37
6.2	LVS Check.....	38
6.3	Neutral Schematic Conversion Work.....	38
6.4	Test Bench.....	38
6.5	What is OCEAN Scripting? .....	39
6.6	Using Scripting to Run Simulation.....	39
6.7	Process Skewing Variation.....	40
6.8	Simulation Settings.....	41
6.9	Summary.....	41

## **CHAPTER 7: SIMULATION RESULTS**

7.1	Pre-layout Simulation.....	42
	7.1.1 Read Operation.....	42
	7.1.2 Read Access Time.....	45
	7.1.3 Write Access Time.....	47
7.2	Summary.....	47

**CHAPTER 8: INTERNAL RACE**

8.1 What is a Race? ..... 48  
8.2 Races Found in SRAM Circuitry..... 49  
8.3 Summary.....51

**CHAPTER 9: POWER SIMULATION**

9.1 Simulation Techniques..... 52  
9.2 Power Simulation Results.....53  
9.3 Summary.....53

**CHAPTER 10: POST-LAYOUT ESTIMATION**

10.1 1<sup>ST</sup> Order Calculation.....54  
10.2 Summary.....55

**CHAPTER 11: CONCLUSION**

10.1 Overall Summary.....56  
10.2 Recommendations for Future Work..... 57

REFERENCES..... 58

## LIST OF FIGURES

	<b>Page</b>
Figure 1.1 : Semiconductor Memory.....	2
Figure 1.2 : SRAM Top Level Architecture.....	4
Figure 1.3 : Read Operation Timing Diagram.....	5
Figure 1.4 : Write Operation Timing Diagram.....	5
Figure 2.1 : 6T SRAM Memory Cell.....	7
Figure 2.2 : 4T SRAM Memory Cell.....	8
Figure 2.3 : 8T SRAM Memory Cell.....	9
Figure 2.4 : 12T SRAM Memory Cell.....	10
Figure 3.1 : SRAM Top Level System View.....	13
Figure 4.1 : SRAM Overall Block Diagram.....	16
Figure 5.1 : 8T Dual-Ported Bit-Cell.....	20
Figure 5.2 : PMOSs Type 1 Pre-charged Circuitry.....	21
Figure 5.3 : Full Keeper.....	22
Figure 5.4 : Half Keeper.....	22
Figure 5.5 : 2 - 4 Decoder.....	23
Figure 5.6 : 3 - 8 Decoder.....	24
Figure 5.7 : Word-line Driver.....	25
Figure 5.8 : Input Latch.....	26
Figure 5.9 : Write Driver.....	27
Figure 5.10 : Pre-charged Circuitry.....	28
Figure 5.11 : Output Latch.....	29
Figure 5.12 : Circuitry of in_latch_drive.....	30
Figure 5.13 : Latch.....	31
Figure 5.14 : Driver.....	31
Figure 5.15 : Controller Block.....	32
Figure 5.16 : Write Column Circuitry.....	33
Figure 5.17 : Delay Write Enable.....	33
Figure 5.18 : Write in_latch_drive Circuitry.....	33
Figure 5.19 : Write Enable Driver.....	34
Figure 6.1 : Memory Cell 8x8.....	36
Figure 6.2 : Memory Cell 8x1.....	36



Figure 6.3 : Decoder 1x8.....	37
Figure 6.4 : Distribution of Channel Length Variation.....	40
Figure 7.1 : Simulation Done at $6\sigma$ Fast Corner.....	42
Figure 7.2 : Read Operation for FFFF Corner.....	43
Figure 7.3 : Simulation Done at $6\sigma$ Slow Corner.....	43
Figure 7.4 : Read Operation for SSSS Corner.....	44
Figure 7.5 : Write Operation for FFFF Corner.....	46
Figure 7.6 : Write Operation for SSSS Corner.....	46
Figure 8.1 : Race of Input 1 & 2.....	48
Figure 9.1 : Power Simulation.....	53

## LIST OF TABLES

	<b>Page</b>
Table 3.1 : SRAM Design Specifications.....	12
Table 3.2 : Pin List.....	14
Table 6.1 : PVT Simulation Requirements.....	41
Table 7.1 : Read and Write Access Time.....	47
Table 8.1 : Race Identification.....	50

## **CHAPTER 1 – INTRODUCTION**

### **1.1 Objectives and Aims of Project**

More often than not, large portion of modern digital chips are occupied by memory and its capacity is forecasted to further increase in the new era of System on Chip (SoC). Hence, high density while maintaining high speed memory design is urgently needed by the semiconductor industry especially due to a great demand for cache applications in very fast processors. Concurrently, analog circuit designers also have to take power consumption problem into consideration due to the increased integration and operating frequency. In addition, portable equipment such as laptop computers, PDAs and cellular phones are more widely used nowadays and this raises the importance of low power design for longer battery operation.

In this fast paced, competitive generation, people are aiming for a long lasting, high speed product that can still fully operate after hours or even days. Thus, this project aims at exploring and implementing high speed memory design to overcome speed degradation caused by large memory capacity. The circuit is also required to operate properly at low power supply voltage, and the actual voltage supply value for this project is set at 1.05V. The power estimation for this project is at 11mW while the current target is at 10mW. In addition, the read and write access time is targeted at 2ns for worst case corners. From the data measured, it is obvious that the read access time needs more time compared to write access time as the global bit-line needs time to be discharged before it can be read out.

Technology scaling and more sophisticated fabrication processes have enabled transistors to be fabricated with much shorter minimum channel length, or so called feature size. Thus, larger volume of integrated circuit can be produced with the same size of silicon wafer and eventually reduces overall production cost. SRAM has to be design with caution else it will offset the advantage of technology scaling as area of the ram's memory array becomes much larger and it will be even larger when additional features is implemented. Thus, we have to be careful at trading-off area, speed and power consumption.

In general, this project aims to produce a high performance CMOS SRAM. With the aid from Cadence IC Design Tools, a front to end custom design tool from schematic, simulation to layout, DRC and LVS are performed. In addition to that, multiple simulations and validation to characterize the performance of the SRAM is carried out in order to ensure that the particular architecture meets all the timing constraints and is good in terms of functionality, robustness and reliability.

## 1.2 SRAM Overview

This section provides a general idea of Static Random Memory (SRAM) and how it operates.

### 1.2.1 Types of Memory

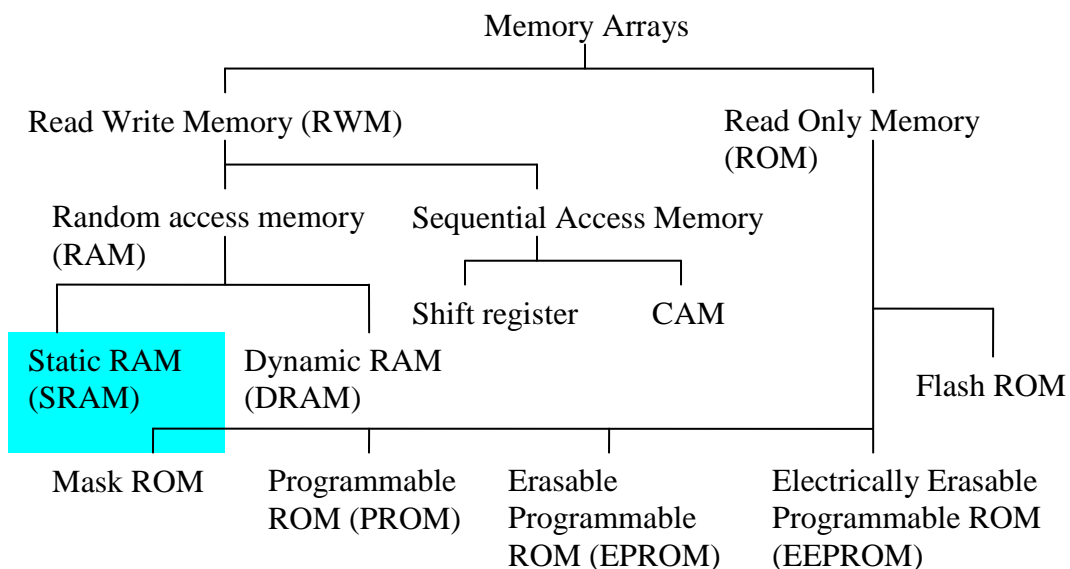


Figure 1.1: Semiconductor Memory

Static Random Access Memory (SRAM) is a variant of read-write random access memory (RWRAM) which can access any of the memory cells within the memory cells plane with the same delay and perform read or write operation to it, in contrast to the sequential access memory which access time is varied by the location in memory cells plane. Compare with Dynamic Random Access Memory (DRAM) which uses capacitive charge to store data and needs to be refreshed periodically in order to compensate for charge loss, SRAM stores data as long as electrical power is supplied and faster than DRAM as it stores data using positive feedback. In addition, SRAM stores complement of the data and allows certain techniques to be used which further

improves the memory access time. Although the data retention power needed for SRAM is lower, area consumed per memory cell is much larger than DRAM nevertheless due to different memory cell design that requires more transistors.

SRAM can be used as embedded block (EBB) memory circuitry such as level 1 or level 2 caches in the microprocessor operating based on the Principle of Locality or as standalone memory chip which needs higher output load driving capability and insusceptible to greater external noise. However, SRAM is playing more important role in embedded memory especially caches in microprocessors and digital signal processing (DSP) circuitry because of its fast access time in nature. Also current technology trend is System on Chip (SoC) which is about putting everything on a single chip. When the processing strength of microprocessor these days becomes more and more powerful, speed of other circuitry within the present computer system becomes a bottleneck for achieving higher performance. Particularly DRAM that is used as main memory to store instructions, raw data and information, its long access delay will unquestionably slow down the overall processing speed. At this point, SRAM has been used as a buffer between the Central Processing Unit (CPU) and main memory (DRAM) to accelerate data fetching speed.

### **1.2.2 Overall Architecture of SRAM Design**

Figure 1.2 shows the general architecture of an SRAM with  $M \times N$  size of memory cell array. This structure utilizes two-way decoding method that is suitable for large memory size and is random access architecture. Each memory cell is connected to a horizontal line that is called "word line" and two vertical lines that are called "bit line". Word line is used to select a memory row and bit line is used to convey data from or into the memory cell. A memory cell in the array is selected by both word line and will be activated according to the input address signals. The function of pre-charge circuitry is to keep the bit lines close to voltage levels for reading operation.

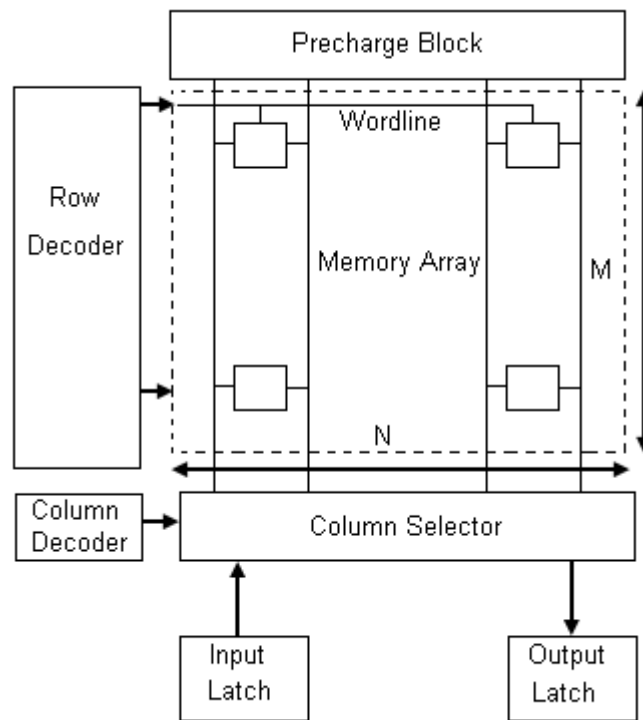


Figure 1.2: SRAM Top Level Architecture

### 1.2.3 SRAM Operation

SRAM read and write operations can be thought of as dynamic CMOS logic operations with two phases – Pre-charge and Evaluate. Before read operation begins, input address has to be ready and it is placed on the address bus, meaning read address has to have a setup time before starts reading. This also applies to read enable signal. Before the read operation starts, bit lines are pre-charged. Then, the evaluate phase commences by activating word line and one of the selected bit lines will start discharging according to the data stored in the selected memory cell. Data that has been read out will be latch out from the SRAM.

Figure 1.3 and Figure 1.4 below show the simplified timing diagram for read and write operation. Each operation includes the setup time, hold time and time from clock to output.

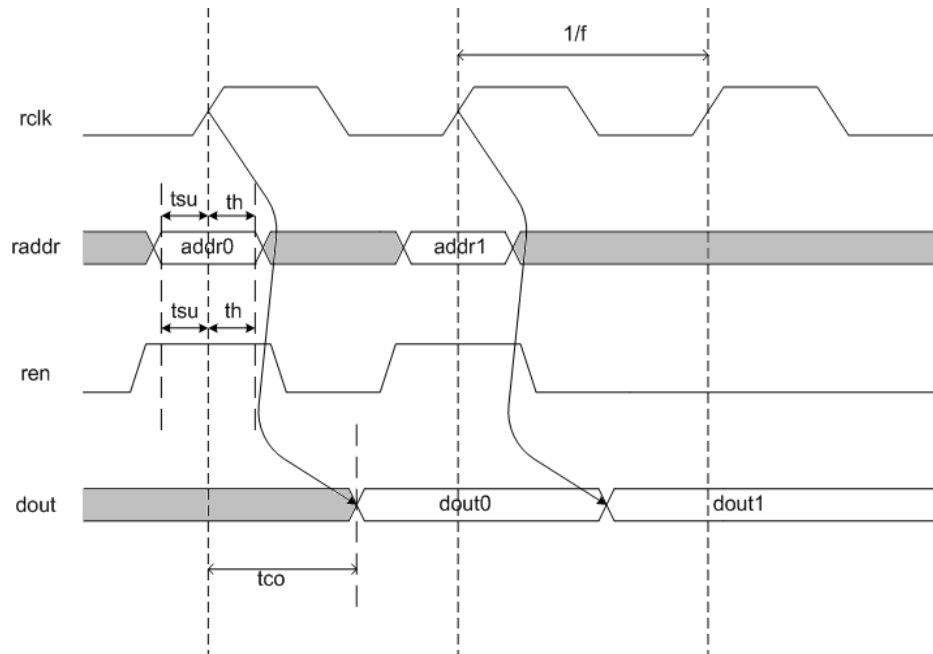


Figure 1.3: Read Operation Timing Diagram

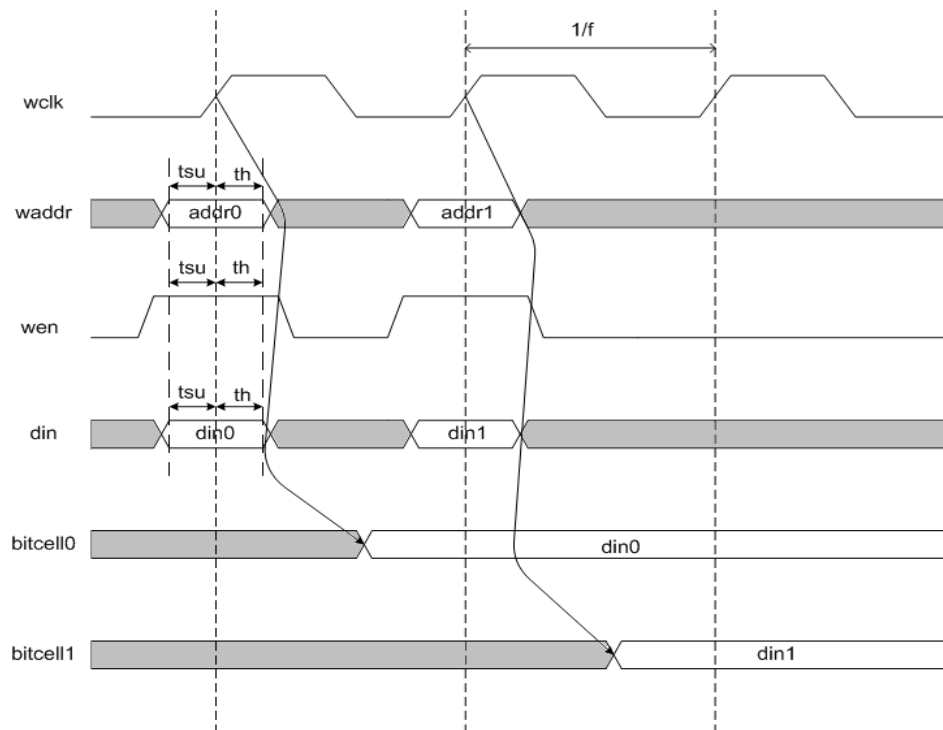


Figure 1.4: Write Operation Timing Diagram

At the beginning of a write operation, both the input address and input data are placed on address bus and data bus, byte enable and write enable signal are selected. The

selected bit cell according to its write address signal will be storing the input data after going through the latch in circuitry.

Referring to the timing diagram above, the setup time is the time that the selected signal needs to be ready before the rising/falling edge of the clock. This SRAM is a synchronous architecture, meaning that all the signals in the SRAM are referring to the clock as the reference signal. As for the hold time, it is the time the data or input address needs to hold to ensure the validity of the output.  $T_{co}$  is the time from clock to output. This is known as the time that the SRAM needs in order to read out the output.

### **1.3 Summary**

This chapter illustrates on the SRAM generic architecture. It gives a basic understanding of the meaning of register files and the focus of its design. Objectives and aims of the project are further elaborated with types of memory shown. A general top level architecture is shown to provide a clearer perspective to the readers' understanding. Furthermore, the timing diagram for the read and write operation are drawn and explained to give a comprehensive view of how the SRAM functions.



## CHAPTER 2 – LITERATURE REVIEW

This chapter will cover for different SRAM memory cell design available with its tradeoff in terms of speed, power and stability for 4T, 6T, 8T and 12T.

T – transistor in the memory cell.

### 2.1 Different SRAM Memory Cell Design Topologies

There are several ways to design a SRAM core. Traditionally, 6-transistor memory cell is used in normal SRAM design that forms the most basic storage element – cross-coupled inverter. In this topology, sense amplifier can be absent if speed is not an issue or else it must be inserted into the design. This memory cell design is also called a single-port memory cell. It can perform either read or write operation at any time but not simultaneously. Figure 2.1 shows the circuit topology of 6T memory cell.

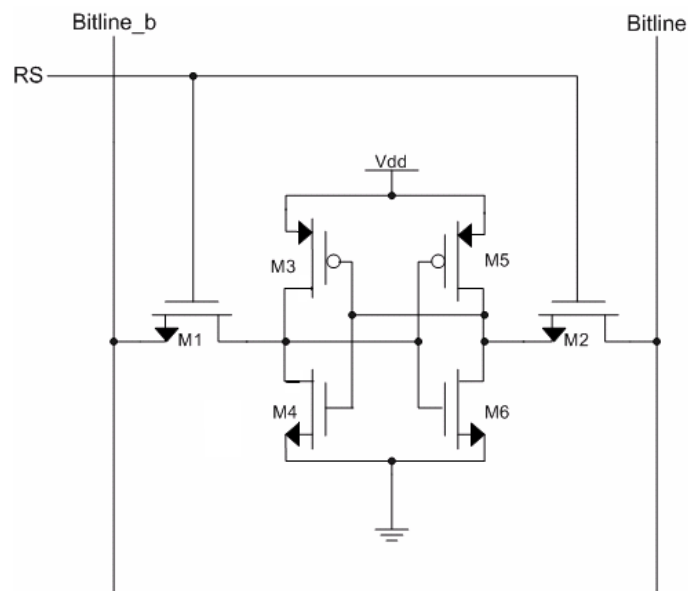


Figure 2.1: 6T SRAM Memory Cell

Additional sense amplifier circuit can be connected to the bit-lines to increase the speed of read operation. There is another similar memory cell which consists of only 4 transistors and is presented in Figure 2.2. The purpose of 4T memory cell is to save area by replacing pull-up PMOSs with resistor loads. However, this topology is not feasible in the deep-submicron era as the supply voltage is kept reducing. 6T design is more robust than 4T design as it has sufficient storage charge to protect against Soft-Error Rate (SER). The occurrence of soft error is due to trajectory of the alpha particles that

strike storage nodes of memory cell causing lost of charges. In addition, Static Noise Margin (SNM) of 4T memory cell is much lower than 6T. SNM is a measurement of how sensitive a memory cell is to process variations and operating conditions. Also, static power consumption of 4T memory cell is much higher unless very large resistance is used to limit the current flowing through resistors.

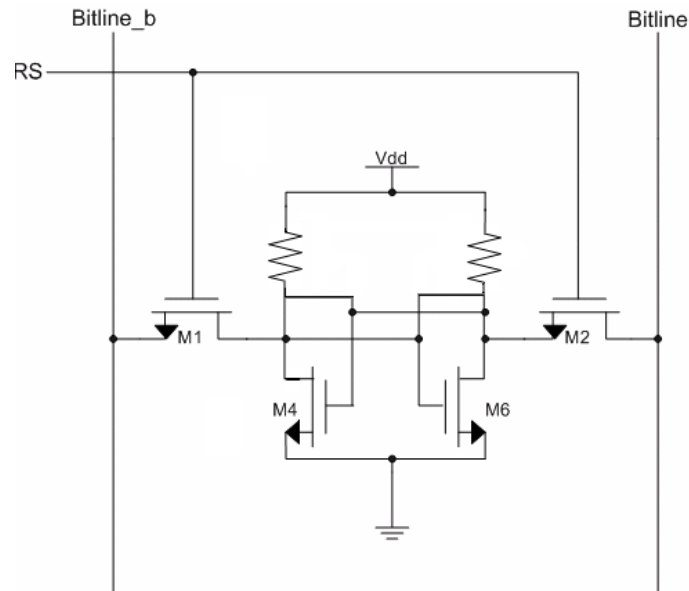


Figure 2.2: 4T SRAM Memory Cell

Besides single-port memory cell, there is also dual-port or multi-port SRAM memory cell that supports multiple synchronous read and write operations in the same cycle. Most of these designs need more than 6T such as 7T, 8T or 12T cell. Although the required size is larger, performance is improved. Figure 2.3 is the 8T SRAM memory cell that supports simultaneous one read and one write operations.

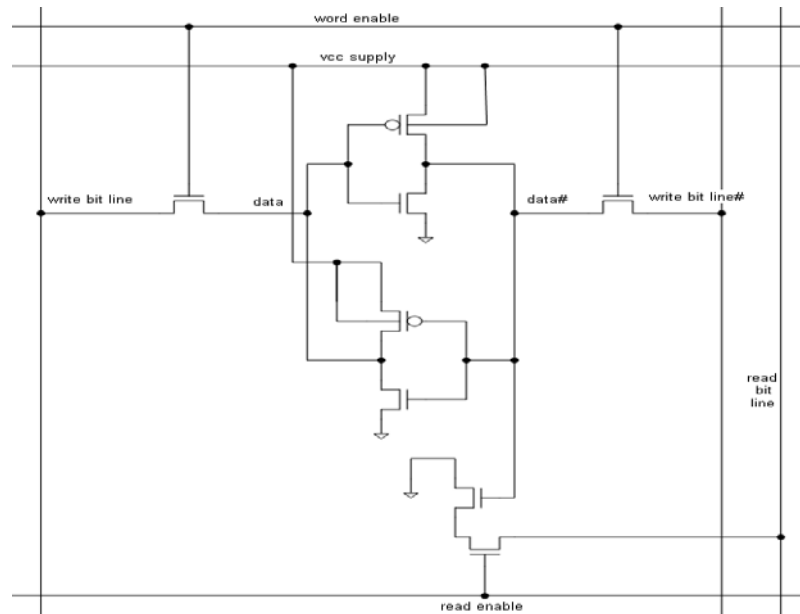


Figure 2.3: 8T SRAM Memory Cell

Refer to Figure 2.3, two additional NMOSs have been added. The NMOSs connect the extra read bit-lines to ground. Same like 6T cell, the storage of data is constructed using a latch. The latch is made out of 2 back-to-back inverters and it can keep logic '1' and '0' depending on the logic status of the write bit line and the write bit line bar.

For example, if the write bit line carrying logic '0', then the write bit line bar will be carrying a '1'. Thus, the data stored in the latch is '0'. The gate of the nmos having the source terminal connected to the ground is connected to the net of the data#, thus, the n transistor is on and it is drawing current to the ground if the n pass gate connected to the read enable is on. Once the n pass gate is on, the read bit line that is pre-charged high before the read operation will be yanked to ground.

Therefore, we are able to read what we write in the latch. This is the operation of read and write of the dual-ported 8T SRAM. With this memory cell, read and write operations can be performed simultaneously but not to the same memory cell. Although sense amplifier can be used for this memory cell design by modifying some circuits, it ends up becoming more complex.

Another similar memory cell design that uses 12T cell is illustrated in Figure 2.4. It has additional access transistors and bit-lines to perform two read and two write operations simultaneously.

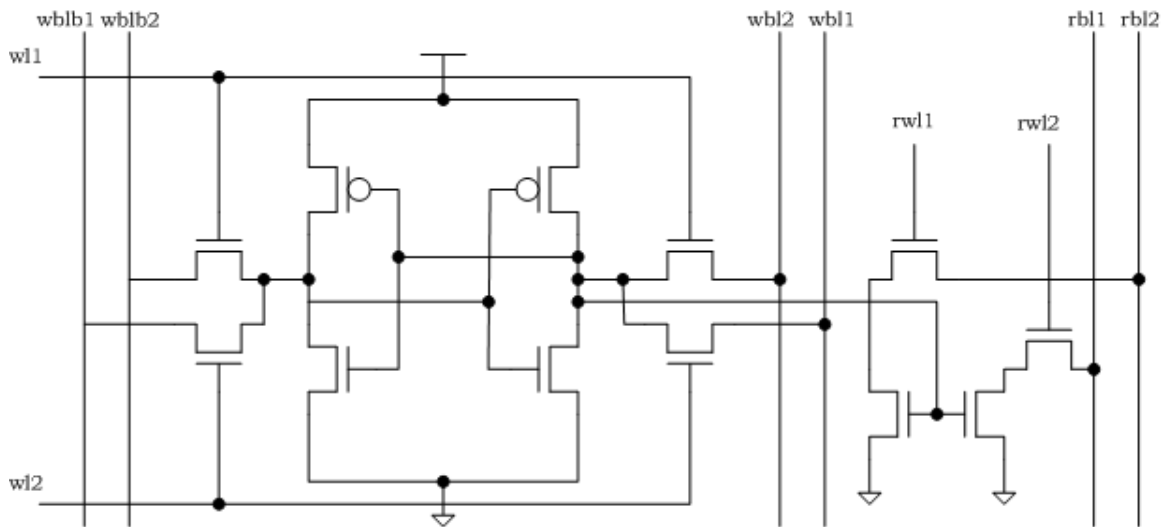


Figure 2.4: 12T SRAM Memory Cell

## 2.2 Summary

In this chapter, different SRAM memory cell design is shown. They consist of 4T, 6T, 8T and 12T. Basically, the 4 types of memory cell is compared in terms of its advantages, disadvantages, performance, functionality and its trade-offs. The purpose of explaining the functions of different types of memory cell is to give a clearer view of the types of memory cell available and its flexibility in memory design.

## CHAPTER 3 – HIGH PERFORMANCE SRAM DESIGN

This chapter discusses briefly the SRAM design-related work in this project. Design requirements, top level view of the SRAM and design specifications.

### 3.1 Design Specifications

The following are design requirements of the SRAM:

- ❖ Power supply voltage is fixed at 1.05V to support the whole SRAM design. This condition is considered low power design.
- ❖ SRAM operating frequency is at 125MHz. In other words, the read and write operation will be completed much less than 8ns. That is the total cycle time referring to the external clock signal to SRAM.
- ❖ SRAM power consumption:  
Maximum Active Mode Current (Dynamic): 10mA  
Maximum Idle Mode Current (Static): 2mA  
Maximum Dynamic Power Consumption: 10.5mW
- ❖ Dynamic power is during the write, read and read + write operation. Thus, active mode current is dynamic. Contrary to that, static power is at idle and leakage operation. Therefore, the idle mode current is set at static operation.
- ❖ Memory capacity is 4Kb with entry size 128 and 33 number of entries. The accurate number of bit-cells available is  $128 \times 33 = 4224$  bit-cells.
- ❖ Design should function properly for the typical class operational temperature. Besides validating according to its typical corners, design should be passing for different Process, Voltage and Temperature (PVT) corners to ensure its reliability and robustness. Different process corners are due to the variation in the transistor's channel length during fabrication process. Different voltage levels are due to the fluctuation of power supply causing by noise or other sources and different temperatures are caused to the heat generation during long hours of operation.
- ❖ This project is using 90nm Intel Process Technology.

Table 3.1: SRAM Design Specifications

Parameter	Descriptions/Specifications
Operating Voltage	1.05V
Operating Frequency	125MHz
Memory Capacity	4224 bit-cells $\approx$ 4Kb
Power Consumption	Max = 10.5mW
Process Corners	typical, fast, slow
Process Technology	Intel 90nm process

### 3.2 Top Level System View

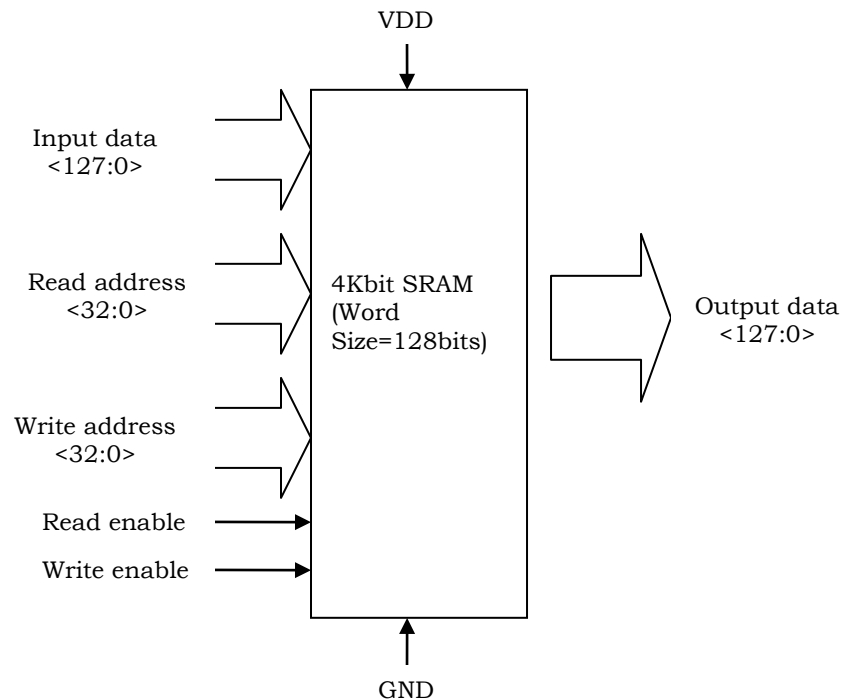


Figure 3.1: SRAM Top Level System View

Figure 3.1 illustrates the top level system view of the SRAM in this project. Two external control signals – read enable (ren) and write enable (wen) are used to control the read and write operation of SRAM. Other control signals are generated internally. As stated before, there are 128 columns in this SRAM circuitry, meaning there are 128 input data available to write into the memory cells. There are 33 rows of read and write decoders, addressing from '0' location till '32', meaning there should be 6 bits address input to accommodate the 33 read and write decoders. Finally, for the output data the condition is the same with the input data. Basically, output data is reading out whatever data stored in the memory cell.

Memory capacity = 33 rows x 128 columns = 4224 bits

### 3.2.1 Pin List

Table 3.2: Pin List

Pin	Direction	Type	Description
idin<127:0>	in	data	Data input
iraddr<33:0>	in	address	Read address
iwaddr<33:0>	in	address	Write address
iren	in	control	Read enable
iwen	in	control	Write enable
odout<127:0>	out	data	Data output

### 3.2.2 Modes of Operation

The SRAM has 5 different operation modes:

1. *Active Write Mode* ( $iwen = '1'$ ,  $iren = '0'$ )

During write operation, input data will be written to the memory cell of SRAM accordingly when write enable and write address is asserted.

2. *Active Read Mode* ( $iren = '1'$ ,  $iwen = '0'$ )

Depending on the read address location and read enable pin, when these 2 signals are asserted to high, read operation can be carried out with reference to the rising edge of the read clock.

3. *Active Dual-Port Operation Mode* ( $iwen = '1'$ ,  $iren = '1'$ )

During active dual-port operation mode, SRAM performs both read and write operations simultaneously depending on the specified read and write address location. User should avoid performing read and write operations simultaneously to the same memory location as the data output will become undefined. In other words, we cannot guarantee that the read out data is the previous or latest data stored in the memory cell. SRAM does not have any mechanism to prevent this fault.

4. *Idle Mode* ( $clk = 'toggling'$ ,  $iwen = '0'$ ,  $iren = '0'$ )

Idle mode is the condition when write enable and read enable is inactive while the clock is still toggling. This means that SRAM is not performing any operation and retains the stored data in each memory cell.

5. *Leakage Mode* ( $clk = 'inactive'$ ,  $iwen = '0'$ ,  $iren = '0'$ )

When the clock is inactive, this means that the whole circuitry is not functioning, without any read or write operation running. All operations stop at a halt and this



doesn't change even external signal write and read enabled. The reason why enable pins does not effect the change in the operations is that it is gated to the clock, once the clock signal is inactive, then the read or write operation will not occur.

### **3.3 Summary**

This chapter discusses the high-performance SRAM design work. The design specifications of the operating voltage and frequency, memory capacity, power consumption, process corners and lastly the process technology is listed down and further elaborated. Top level system view is shown by listing out the pin list required for this SRAM. Lastly, 5 modes of operations of active, idle and leakage in the SRAM is explained.

## CHAPTER 4 – SYSTEM ARCHITECTURE DESIGN

This chapter discusses one of the stages in VLSI/Analog design flow – architectural design. In this stage, design specifications are translated into a collection of functional blocks and how these blocks are connected. Proper architecture design will reduce designer’s burden in the circuit design phase.

### 4.1 SRAM Architecture Design

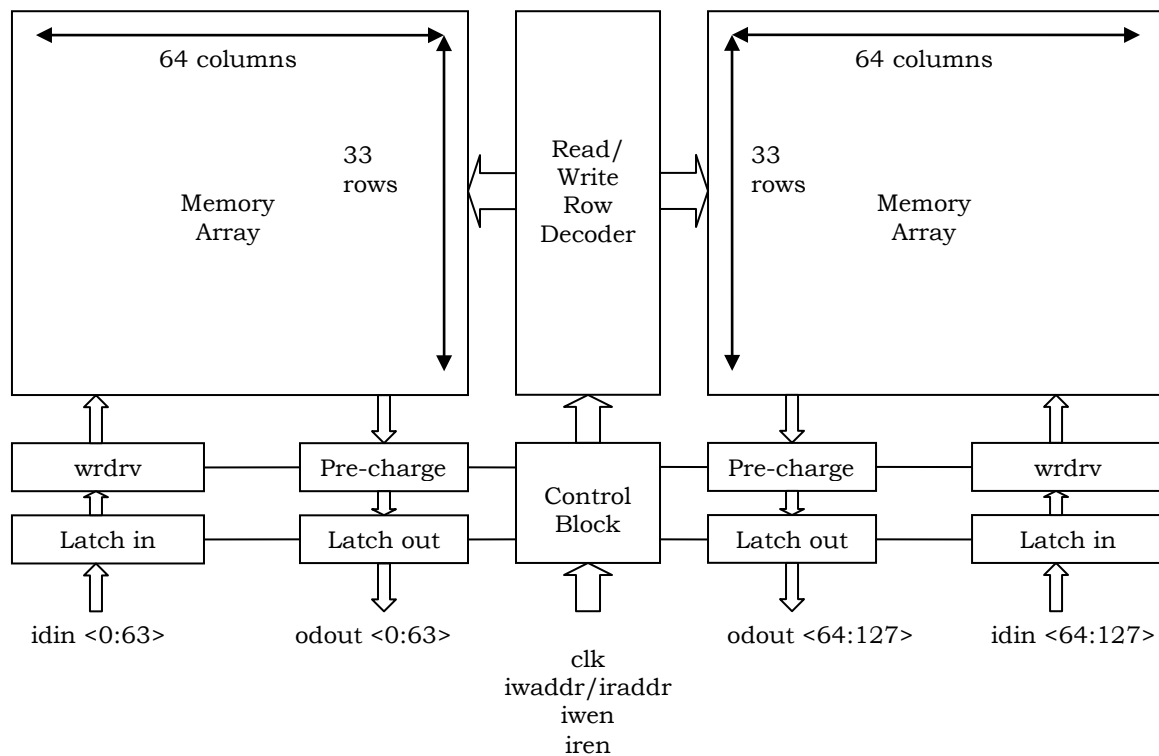


Figure 4.1: SRAM Overall Block Diagram

The overall block diagram of designed SRAM is illustrated in Figure 4.1. There are 2 memory arrays of 2112 bit-cells situated on the left and right of the RAM. Every single bit cells are connected to the IO blocks which contain data, signals of address input buffers, data input and data output latches. The tapered buffers with large current driving capability supply signals to the internal and external circuits. The reason of designing the RAM in such a way by dividing the memory array to the left and right is to optimize the driving capability of the row decoders. In other words, when the number of columns increases, the capacitive effect of the decoded signal connecting to all the columns will be higher, thus, this will impact the speed of the circuit which we’re

aiming to run at high frequency. In addition, when the capacitive effect takes place, sizing of the decoded driver will be much larger, and this will consume more area than expected.

This is a dual-ported RAM that can perform a read and a write operation simultaneously in the same cycle in the condition that the read and the write operation are not located at the same address. The meaning of dual port is referring to the bit cell in the memory array as it is capable of a read operation and a write operation. It is constructed in such a way that it is formed from 8 transistors (8T).

Figure above shows the alignment of the SRAM blocks. The dual-ported SRAM have 128x33 bit cells in the memory array. 128 is the number of bit cells column wise while the 33 is the word lines for the vertical rows. Thus, we can say that there are 128x33 or 4224 bit cells in the memory array of the dual-ported 8T SRAM.

This is a generic type of designing a RAM. The placement of the memory array is situated on the left and right. As stated before, the column x row is 128x33 bit cells, thus, due to the division of the memory array to the left and the right, the memory array is partitioned as 64x33 each. With equal bit cells on each side, the decoder block in between is functioning as an address decoder, to locate the specific bit cell or so-called address to store data in the bit cell.

As shown above, the IO block is located on both sides below the memory array. These blocks are the interface to the external world as it feeds the input data to the RAM and also get the required data from the RAM to the external world during the read operation. Thus, the data writing in to the RAM and the data reading out from the bit cells are controlled by latches. These latches are located inside the IO block and they operate to control the latching in and latching out of the signals.

And lastly, the controller block. This block as its name stated, is mainly to control the operation of the whole RAM. The clocking signals for read and write operations are connected into this block. All the operation for reading, writing, latching the data in and out, etc, are controlled by the controller through the reference signaling of the clock.

## 4.2 Summary

This chapter discusses about SRAM architecture design. Each sub-block is connected and integrated into an overall block diagram. In short, this architecture has basically 4 main blocks that needs to be put into consideration. They are the memory array, IO interfaces, row decoders and lastly the controller block. These 4 main blocks are further elaborated in terms of its functionality and operating conditions.

## CHAPTER 5 – CUSTOM CIRCUIT DESIGN OF SRAM

Full custom design style requires human's intervention in both transistors level circuit with the intention of fully optimizing the design. This chapter presents the circuit design of SRAM building blocks. Every building block will be further elaborated of its functionality in terms of transistors level.

### 5.1 Memory Cell

Stability and area of memory cell are needed to be considered in memory cell design. Stability of memory cell is related to the Static Noise Margin (SNM). SNM is the ability to cope with the DC noise due to process variation and operating conditions. SNM shows the maximum value of noise voltage that can be tolerated before storage node tripping occurs. Some SNM is reserved for dynamic noise such as voltage supply ripple, thermal noise, cross talk and soft error.

For this particular design, 8T architecture is used. The reason why 8T is used for this SRAM circuitry is that it is more stable compared to 6T. The tradeoff is in terms of speed and area in comparison with 6T architecture. Figure 5.1 shows how the bit cell is connected internally using eight transistors.

In the bit cell, the storage of data is constructed using a latch. The latch is made out of 2 back-to-back inverters and it can keep logic '1' and '0' depending on the logic status of the write bit line and the write bit line bar.

For example, if the write bit line carrying logic '0', then the write bit line bar will be carrying a '1'. Thus, the data stored in the latch is '0'. The gate of the nmos having the source terminal connected to the ground is connected to the net of the data#, thus, the n transistor is on and it is drawing current to the ground if the n pass gate connected to the read enable is on. Once the n pass gate is on, the read bit line that is pre-charged high before the read operation will be yanked to ground.

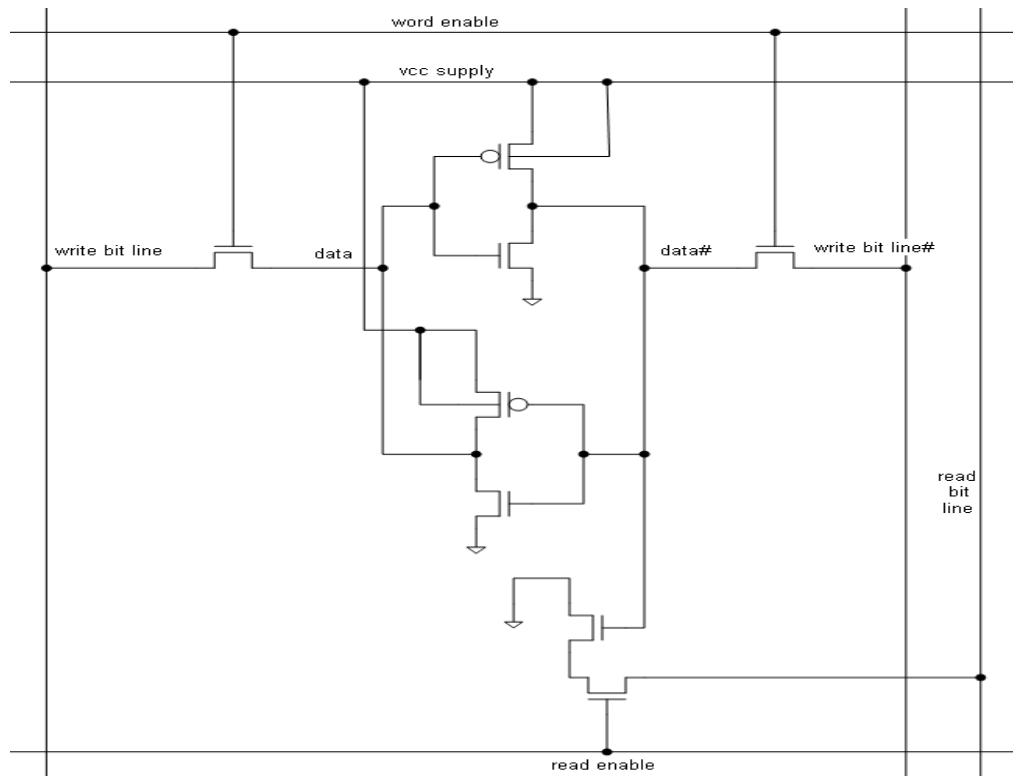


Figure 5.1: 8T Dual-Ported Bit-Cell

Therefore, we are able to read what we write in the latch. This is the operation of read and write of the dual-ported 8T SRAM.

## 5.2 Memory Array

There are 2 memory arrays located at the left and right of the whole RAM. Each memory array is having 64 columns and 33 rows. Overall the total bit-cells in each memory array are 64 x 33 that is equivalent to 2112 bit-cells. The schematic building of the bit-cells to bigger blocks till the top hierarchy will be narrated further in the following chapter.

### 5.3 Pre-charge Circuitry / Bit-line Keeper

Bit-line keeper is used to recharge back the bit-lines capacitance after every operation and to compensate for the charge leakage. For this project, bit-lines voltage has to be close to supply voltage, so that accurate value can be read out from the output latch. As a result, PMOSs type pre-charge circuitry was chosen as it causes very small bit-lines voltage drop. If NMOSs type circuit is chosen, bit-line voltage can only be charged up to  $V_{dd} - V_{thn}$ . An additional bit-line keeper is needed for implementing dual-port feature. There are 3 types of keeper that can be used accordingly depending on its suitability to its circuit requirement and specifications. Figure 5.1 shows the PMOSs type 1 pre-charge circuitry that is built from 3 PMOSs transistors.

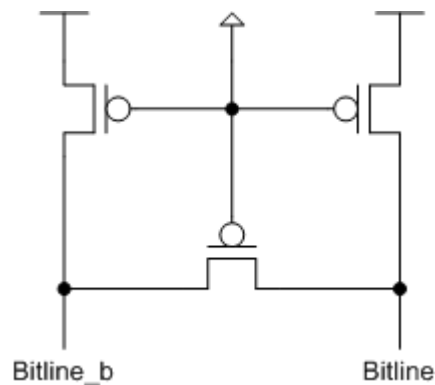


Figure 5.2: PMOSs Type 1 Pre-Charged Circuitry

According to the Figure 5.2 above, all the PMOSs operate in linear region and current will flow down to bit-lines whenever bit-lines voltage drops. The pull up PMOSs will determine the recharging delay of high capacitive bit-lines after operation. The transistor in between or located in the center is for equalization purpose. This means that it is used to ensure the voltage level of the two bit-lines are equal else we'll encounter read/write error during a run. Besides balancing up the voltage difference, the central transistor is also used to fasten the bit-line recovery process.

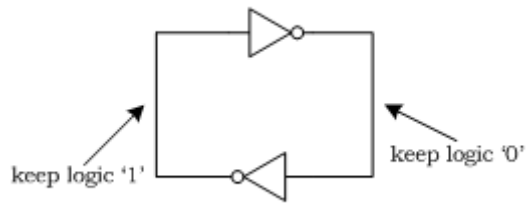


Figure 5.3: Full Keeper

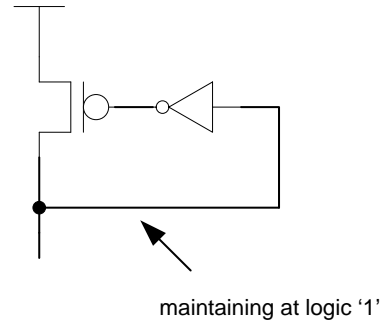


Figure 5.4: Half Keeper

Figure 5.3 and Figure 5.4 above are bit-line keepers that are keeping logic level constant for a certain period of time. Due to the fact that it has a keeper to keep it at a constant level, thus, it is considered as a stable circuit. Figure 5.3 is a full keeper (type 2) that is capable of keeping logic '0' and '1'. It is a back-to-back inverter that is feeding and maintaining the logic level to each other. As for Figure 5.4, it is a half keeper (type 3) that is capable of keeping one logic level and maintaining it. This half keeper is suitable to use in SRAM architecture as it is capable of keeping and maintaining the bit-line at a certain required logic level.

For Type 1 architecture, it is used in 6T architecture more often as it requires equalization for the purpose of having sensing amplifier during read operation. Equalization is critical in 6T as when there's a small difference between the bit-line and bit-line bar, it'll be amplified in sense amplifier to the output during read operation. If the voltage difference is not caused by data stored in the memory cell, then we'll get the wrong data.



### 5.4 Row Decoders for Read and Write

The read and write decoders are located at the center of the RAM. There are a total of 33 decoder blocks that is connected to the memory array in order to support 33 rows of address location. Once the read/write decoders are enabled, one of the specified rows will be activated and the whole row of bit-cells will be active to perform a read/write operation.

Row decoders are built from NAND logic gates and inverters. It is a simple implementation to define the exact location of the specific address. In this architecture we have 33 rows of address location. In order to drive 33 word lines, we need  $(2^6 = 64)$  6 bits of read/write address to accommodate 33 rows. The read/write address signals will be ranging from  $\langle 5:0 \rangle$ .

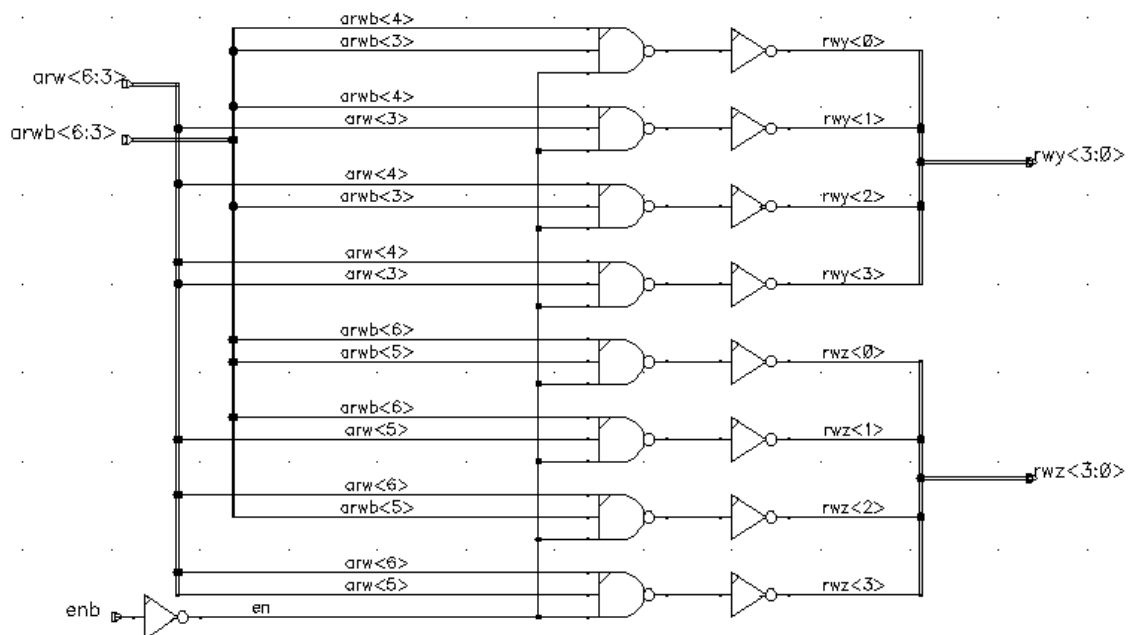


Figure 5.5: 2 - 4 Decoder

Read and write address  $\langle 5:0 \rangle$  is used to generate 33 rows with respect to the word line connecting to the memory cells. In this SRAM, there are 7 signals ranging from  $\langle 6:0 \rangle$ . Due to the fact that we're not using read/write address  $\langle 6 \rangle$ , thus, we deserted the signal by connecting it straight to ground. Referring to the Figure 5.5 above, this is a 2 to 4 decoder that is basically having input signal read/write address  $\langle 6:3 \rangle$  convert to y and z

output of 4 each. This will define which row that is needed to be enabled. The purpose of having the enable input pin is to control the activity of the decoder. If the enable pin is deserted, then the decoder will not be able to perform any row select to its respective location.

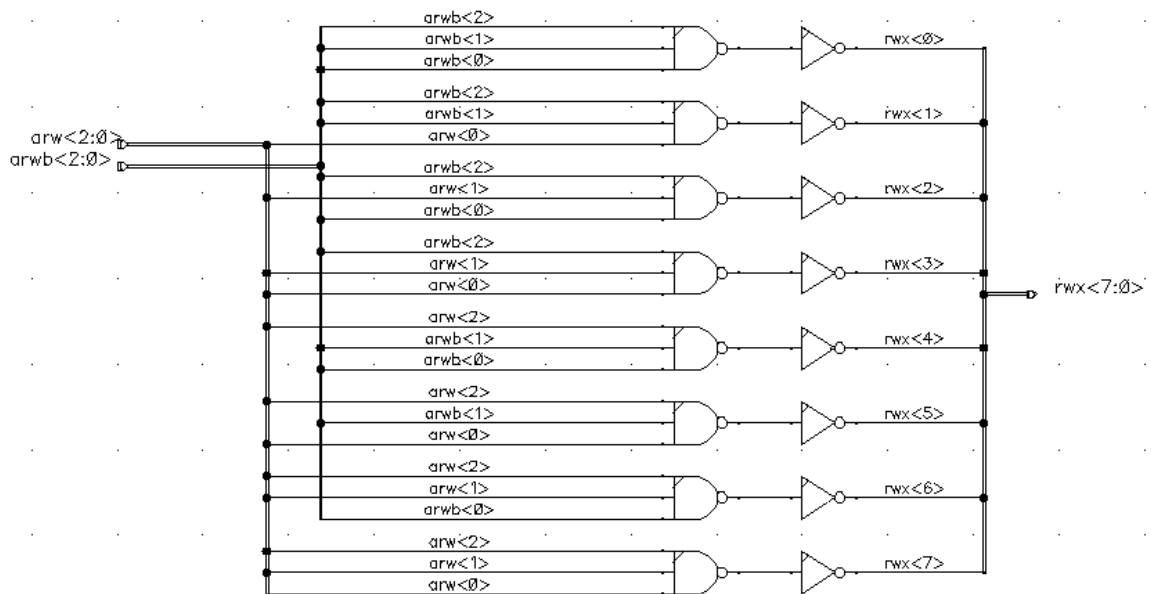


Figure 5.6: 3 - 8 Decoder

Observing Figure 5.6, there are 3 signals of read/write address ranging from  $\langle 2:0 \rangle$  connected as the input signal to logic NAND3s. This is a 3 to 8 decoder that is representing output of x that is going to determine which row of addresses that the signal going to initialize. Outputs of read/write x, y and z will be connected to 33 blocks of word-line drivers to assert the specified rows.