National Library of Canada

Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch

Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Performance Analysis and Design of Optimized Static Random Access Memory (SRAM)

Dewan Jahangir

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

June 1992

© Dewan J., 1992

National Library of Canada

Acquisitions and Bibliograpnic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-80997-3

Canada

Abstract

Performance Analysis and Design
of Optimized Static Random Access Memory (SRAM)

Dewan Jahangir

High speed, low power & area, and reliability are important design goals in high-performance Static RAM. The SRAM design varies in multi-dimensions (block, row, column) and type of implementation of other peripheral circuits. Therefore, in general, the performance analysis of the whole structure is particularly difficult. A more comprehensive model rather than the general RC delay model is required in order to achieve realistic designs.

In this thesis, a new Precharge circuit called Power Down Y-controlled PMOS (PDYCP) load precharge circuit is designed. Our PDYCP precharge circuit reduces power consumption by a factor of three over that of a conventional Y-controlled load (YCL). A design methodology of SRAM with an analytical model such as area, delay and power consumption suitable for module generator is developed. The performance of the SRAM is analyzed using Elmore's delay model which is confirmed with SPICE results. An optimized design of SRAM using Khun-Tucker optimization criteria is presented. Furthermore, a design aid tool for optimized design of SRAM is developed which is written in 'C'. Some optimization results are tabulated under various array sizes and load conditions.

Acknowledgments

I would like to express my sincerest gratitude to my thesis supervisors, Dr. A.J. Al-Khalili and Dr. B. Haroun, for their constant encouragement and knowledgeabie guidance during the course of this work. They have always been available to direct and advise whenever the occasion arose. It was a really enjoyable experience working with people who are so enthusiastic and insightful. Their constructive criticisms and keen ability to pinpoint important issues and weaknesses have been a major part in the shaping of this thesis.

I would like to acknowledge the United Nations Educational Scientific & Cultural Organization (UNESCO), Paris and the United Nations Development Project (UNDP), Dhaka, Bangladesh for their Fellowship Award to me in the period from May 1989 to December 1991. Also, I would like to thank the Bangladesh Institute of Technology (BIT), Dhaka for granting me a study leave and thereby enabling me an opportunity to pursue higher studies.

I would also like to acknowledge the VLSI lab personnels Mr. G. Patel and Mr. D. Hargreaves for their great contribution & commitment to the lab which made a really pleasant environment for work.

Finally, I wish to express my deepest appreciation to my family for their patience and moral support. Without their generous support and encouragement this work would not have been possible.

Dedicated to my Family

and Friends

Table of Contents

1 Introduction to SRAM Design: An overview 1

1.0 Introduction 1

1.1 Motivation 1

1.2 Organization of the Thesis 3

1.3 Basic SRAM Architecture 4

1.4 Static Random Access Memory (SRAM) cells 6

1.4.1 Organization and Operation of a SRAM 8

1.4.2 A 6T SRAM Cell Design 14

1.5 Decoding Scheme 19

1.5.1 Dynamic Double-word line (DDWL) structure 19

1.5.2 Divided Word-line (DWL) structure 22

1.6 Circuit techniques for Decoder 24

1.6.1 PMOS-Load Decoder 24

1.6.2 Transfer Word Drive (TWD) Decoder 28

1.7 Precharge Scheme 27

1.7.1 Different Precharge Techniques 30

1.7.2 Power Down Y-Controlled PMOS (PDYCP) Load bit line Precharge Technique 35

1.8 Sensing Scheme 40

1.8.1 Operation of a Current- Mirror Sense Amplifier (CMSA) 40

1.8.2 Various SA Circuit Techniques 42

1.8.3 Hierarchical Sense Amplifier Architecture 45

2 Modeling and Analysis of SRAM 49

2.0 Introduction 49

2.1 Capacitance Model of Word and Bit Line 49

2.1.1 Word Line Capacitance Model 49

2.1.2 Word line driver I/O Capacitance 51

2.1.3 Bit line Capacitance 53

2.2 Word line Delay Analysis 54

2.2.1 Word line delay modeling using Elmore's Delay Model 54

2.2.2 Approximate form of Word line 57

2.3 Precharge Delay Analysis 62

2.4 Sensing Delay 68

2.5 Write Access Delay 76

3 Area Model of SRAM 82

3.0 SRAM Area 82

3.1 SRAM cell Area 83

3.2 Decoder Area 86

3.3 Sense Amplifier Area 86

3.4 Precharge circuit Area 87

3.5 Column select T-gate Area 88

3.6 I/O peripheral Area 89

4 Power consumption in SRAM 92

4.0 Introduction 92

4.1 Standby power 92

4.2 Read power 92

4.3 Write power 94

4.4 Standby Mode power dissipation 95

5 SRAM Design Optimization 98

5.1 Introduction 98

5.2 Kuhn-Tucker Equations 98

5.3 Sizing and Optimization of Area, Delay and Power in Logic Blocks 99

5.3.1 Optimum Sizing Algorithm of Transistors in a SRAM 101

5.4.1 Optimum Word Driver Design 103

5.4.2 Optimum Precharge Circuit Design 107

5.4.3 Optimum Sense Amplifier Design 110

6 Design and Layout Implementation of a SRAM 114

6.0 Example: Optimized SRAM Design 114

6.1 Layout Implementation 118

7 Conclusion 133

Appendix A SRAM Modeling 136

A.1 Capacitance and Resistance calculation of a MOS Transistor 136

A.2 SRAM Node Capacitances 140

Appendix B Optimized SRAM Design 141

B 1.1 Optimized Word Driver Design 141

B1.2 Optimized Precharge Circuit Design 142

B.1.3 Optimized Sense Amplifier Circuit Design 143

B.2 Regression Results for SRAM Modeling 148

List of Figures

1.1	A typical SRAM Architecture	5
1.2	(a) A 4-transistor SRAM cell	7
1.2	(b) A 6-transistor SRAM cell	7
1.3	(a) Organization of a SRAM	9
1.3	(b) SRAM with peripheral circuits	10
1.3	(c) Read-0 operation of a SRAM	13
1.3	(d)Write-1 operation of a SRAM	13
1.4	(a) Graphical representation of Static Noise Margin	16
1.4	(b) Stability margin as a function of r	16
1.5	(a) Dynamic Double-word line (DDWL) structure	20
1.5	Trade-off (b) Word line delay, (c) area and power as a function of number of sections or blocks in a DDWL structure	21
1.6	(a) Conventional Divided-word line structure	23
1.6	(b) Hierarchical word decoding Architecture	23
1.6	(c) Comparison of DWL and HWD	23
1.7	(a) PMOS load decoder and (b) Conventional CMOS decoder	25
1.7	(c) 1-Mbit SRAM word decoding scheme	25
1.7	(d) Circuit schematic of word decoder using PMOS load	25
1.7	(e) Delay time of two types of decoders	27
1.7	(f) Current of word decoder	27
1.7	(g) Word-decoder circuit (for four memory subarray)	27
1.8	(a) Transfer word drive decoder, (b) Conventional decoder	29
1.9	(a) and (b) Continuous precharge circuits	31
1.10	(a) and (b) Controlled precharge circuits	31
1.11	Bit line load and $\overline{ATD}$ controlled precharge circuit	32

1.12 $\overline{ATD}$ controlled conventional precharge scheme 32

1.13 Y-controlled bit-line load (YCL) precharge circuit 34

1.14 Power Down Y-controlled PMOS (PDYCP) load precharge circuit 36

1.15 SPICE simulation plots using PDYCP precharge technique
for consecutive READ/WRITE operation 37

1.16 (a) A CMOS Current Mirror Sense Amplifier (CMSA) 41

1.16 (b) Operation of a CMSA for a read-0 41

1.17 (a) Conventional CMSA 43

1.17 (b) Input controlled PMOS-load (ICPL) sense amplifier 43

1.18 Dynamic Gain Control Double end Amplifier 44

1.19 Modified two stage CMSA 44

1.20 Double line sensing structure 46

1.21 (a) Sense Ampiifier Circuit for fig. 1.21 (b) 47

1.21 (b) Two-stage local amplification and triple-sense-line structure 47

1.22 Four-stage Sensing Scheme 48

2.1.1 Word line Capacitance Modeling 50

2.1.2 Word driver I/O Capacitance Modeling 51

2.2.1 (a) Word driver driving a row of a SRAM, (b) Physical representation
of an word line, (c) R, C delay model of a Word line 55

2.2.2 (a) π- approximate model of a word line in a cell segment,
(b) Approximate π- R C model of a word line 58

2 2 2 (c) Comparison of analytical and SPICE simulation results
for different word sizes 61

2.3 (a) Precharge delay modeling 63

2.3 (c) Precharge Delay Analytical results 67

2.3 (d) Comparison of analytical and SPICE simulation results
for precharge delay 67

2.4 (a) Read-1 delay modeling 69
2.4 (b) Read-0 delay modeling 70
2.4 (c) Read-0: Comparison of analytical and SPICE simulation results 74
2.4 (d) Read-1: Comparison of analytical and SPICE simulation results 75
2.5 (a) Write-1 delay modeling 77
2.5 (b) Write-0 delay modeling 78
2.5 (c) Write-1 access delay & AT^2 as a function of write buffer size 81
3.1 (a) Top view of a SRAM cell access transistors 84
3.1 (b) Top view of a cross-coupled inverter in a SRAM cell 84
3.6 I/O Peripheral circuits of a SRAM 91
4. (a) SPICE Comparison of Read Power dissipation using PDYCP and YCL Precharge circuit 96
4. (b) SPICE Comparison of Write Power dissipation using PDYCP and YCL Precharge circuit 96
4. (c) SPICE Comparison of average cycle Power 97
6.1 Layout of a SRAM cell 121
6.2 (a) Test circuit of a SRAM cell (includes cell, decoder & peripherals) 122
6.2 (b) HSPLOT for test circuit- alternate READ/WRITE operations 123
6.3 A SRAM cell with precharge and column select circuit layout 124
6.4 A Current Mirror Sense Amplifier (CMSA) layout 125
6.5 I/O peripheral circuit layout of a SRAM layout 126
6.6 A READ-0 timing analysis of SRAM layout 127
6.7 WRITE-0 timing analysis of SRAM layout 128
6.8 An worst case precharge delay 129
6.9 A moderate case precharge delay 130
6.10 A WRITE-1 timing analysis of the layout 131
6.11 Current consumption in different read/write cycles in SRAM layout 132

A. 1.1 Parasitic Capacitance of a MOS Transistor 136

A. 1.2 Area and peripheral components of diffusion capacitance 137

List of Tables

1 Comparison of PDYCP & YCL precharge performance 39

2 Example of Optimized design of a word driver
under variable load conditions 106

3 Example of Optimized design of a precharge circuit
under variable load conditions 109

4 Example of Optimized design of a sense amplifier
under variable load conditions 112

B.1 Word delay fit constants 148

B.2 Precharge delay fit constants 149

B.3 Sensing delay fit constants 149

Chapter 1
INTRODUCTION TO SRAM DESIGN: An overview

1.0 Introduction

Memories are devices which can store information. There are two kinds of memory in general, such as volatile (data is erased when power is turned off) and non-volatile (retains data even if power is turned off). Semiconductor RAMs are volatile. Semiconductor RAMs are of two types: static (SRAM) and dynamic (DRAM). The SRAM capacity has quadrupled every two or three years in circuit and process technology[7]. The fastest access speed reported is 15ns with 4-Mb CMOS SRAM [1]. In SRAM, information is stored in the static structure of a cross-coupled inverter (latch-storage), while in DRAM information is stored in a capacitor. SRAM does not need to be refreshed but DRAM must be refreshed in a regular interval otherwise data will be corrupted. The advantages of a SRAM over a DRAM are its high speed, low power dissipation and high reliability. SRAMs are used in high speed applications such as in main and cache memory for computers, or test pattern memory in VLSI testing applications. They are also used for low power applications for holding data with battery backup in portable computers, memory cards and data terminals.

1.1 Motivation

Optimization in SRAM design is essential in many applications such as Embedded RAM, Cache Memory, Module Generators where the designer has to match the specific design goal. The optimization problem is related to the design requirements, circuit topology and modeling technique. Area, delay and power consumption are three

imperative factors for performance trade-off in VLSI. Accurate modeling of the signal path delays in the circuit is important for high performance integrated circuit design. Although a device level simulator like SPICE can provide detailed and accurate delay information and optimization for small circuits, the analysis becomes more complicated and the computation time increases rapidly for large circuits specially for SRAM. Therefore, analytical delay model is required in general to solve the optimization problem in SRAM.

There have been two methods of general delay models, macromodeling [27] and transistor level models using RC tree modeling [18]. In macromodeling the whole circuit is partitioned into different sub-circuits instead of individual devices. The use of macromodeling is limited to circuits with regular logic gates (i.e. NAND's, NOR's, XOR's, etc.). The RC tree delay model is most successfully used for circuits with arbitrary resistances and capacitances. In the RC approach, a nonlinear MOSFET is modeled in terms of linear RC element. In general, the RC delay model suffers from accuracy and deviates far from SPICE simulation results. Inaccuracies in RC delay model results from neglecting the nonlinearities of the MOS transistors and from difficulties in including input waveform effects [26]. The solution to these problems can be offered by incorporating fit constants and including the effect of ramp input into the model.

Developing an efficient analytical model and goal directed optimization algorithm for automatic generation of transistor sizes in SRAM with low computational complexity is one of the aims of this thesis. As regards to computation time (for both man and machine) we decided to use RC tree delay model to develop a tool for optimized design of SRAM. As far as the accuracy of the model is concern we regression fitted our RC delay model with numerous SPICE simulation results for various SRAM array sizes and finally obtained a comprehensive RC delay model with fair agreement with SPICE. Moreover, we include in the model the analytical expression for the effect of ramp input to the signal delay. The maximum deviation of our model from SPICE simulation is 10 percent. The

important characteristics of our modeling and optimization tool are its accuracy, robustness and modularity.

1.2 Organization of the Thesis

As stated earlier, the primary objective of this research is to develop a design aid tool for optimized design of SRAM to fulfill design criteria specially for embedded class of RAM. The circuits of interest for optimization are word driver, precharge and sense amplifier which has significant contribution on the access speed, overall chip area and power dissipation. However, the optimized design of other peripheral circuits are obtained from numerous SPICE simulation experience.

In chapter1, we investigated the design of SRAM cell and other peripheral circuits. Organization and operation of SRAM is briefly described. The design criteria for SRAM cell is also studied. Different types of decoding, precharge and sensing techniques are also depicted. We discussed our designed Power Down Y-controlled PMOS (PDYCP) load precharge circuit and also compared the performance using PDYCP and conventional Y-controlled load (YCL) precharge. The comparison reveals the excellent amount of power saving using PDYCP precharge.

Chapter 2 is devoted to modeling and analysis of SRAM using Elmore's RC delay modeling approach. The delay models specially for word, precharge, read and write are obtained. The RC delay models are regression fitted with numerous SPICE simulations and thus comprehensive delay models are obtained which confirms to SPICE simulation results with very little error (less than 10%) Comparison of our analytical models with SPICE for different SRAM array structure are depicted.

In chapter 3 we describe the area model of SRAM which is used by our optimization algorithm.

In chapter 4, power consumption in SRAM using PDYCP and YCL precharge are studied in terms of analysis and SPICE simulation. We compare power consumption using

both the precharge circuit techniques for any SRAM array size for read, write and standby time. Our results show that using PDYCP precharge technique power consumption in SRAM is reduced by a factor of three in comparison to YCL precharge.

Chapter 5 deals with optimization in SRAM design. We assume area is linearly related to power. Using the delay and area models in chapter 2 and 3 respectively we developed optimized design parameters for different circuits in SRAM. Since the unconstrained based optimization is insufficient to solve the optimization problem in general, we followed constrained based optimization. We used the well known Kuhn-Tucker optimization criteria to solve the problem. An algorithm for optimized design of SRAM is also discussed. Various results are tabulated using our approach which shows that with a little compromise in delay a significant amount of area can be saved.

Using our modeling and optimization formulas we developed a design aid tool for optimized design of SRAM. Chapter 6 presents a design example for a 4 k x1 bit (64x64) SRAM using our tool. We also present a layout for the 4 k x 1 bit SRAM using our results.

Finally, in chapter 7 we summarize our works in this thesis and identify future research problems that can be derived from this thesis.

1.3 Basic SRAM Architecture

A typical SRAM architecture is illustrated in Fig. 1.1. The chip is organized with densely packed cell arrays in the form of a matrix. Each memory cell can store one bit of data. Around the cell array are the row decoder, column decoder, precharge circuit, data multiplexors, read/write circuits, control circuit and I/O buffers. The memory array consists of $2^M x 2^N$ bits of storage, where M and N are the number of address bits. The purpose of row and column decoders is to determine which cell to address for read/write. The row decoder addresses one row (word) of 2^N bits out of 2^M words. The column decoder addresses 2^K of 2^N bits of the accessed row, where K is the number of output bits. The purpose of a precharge circuit is to precharge bit lines and the data bus and thereby improve

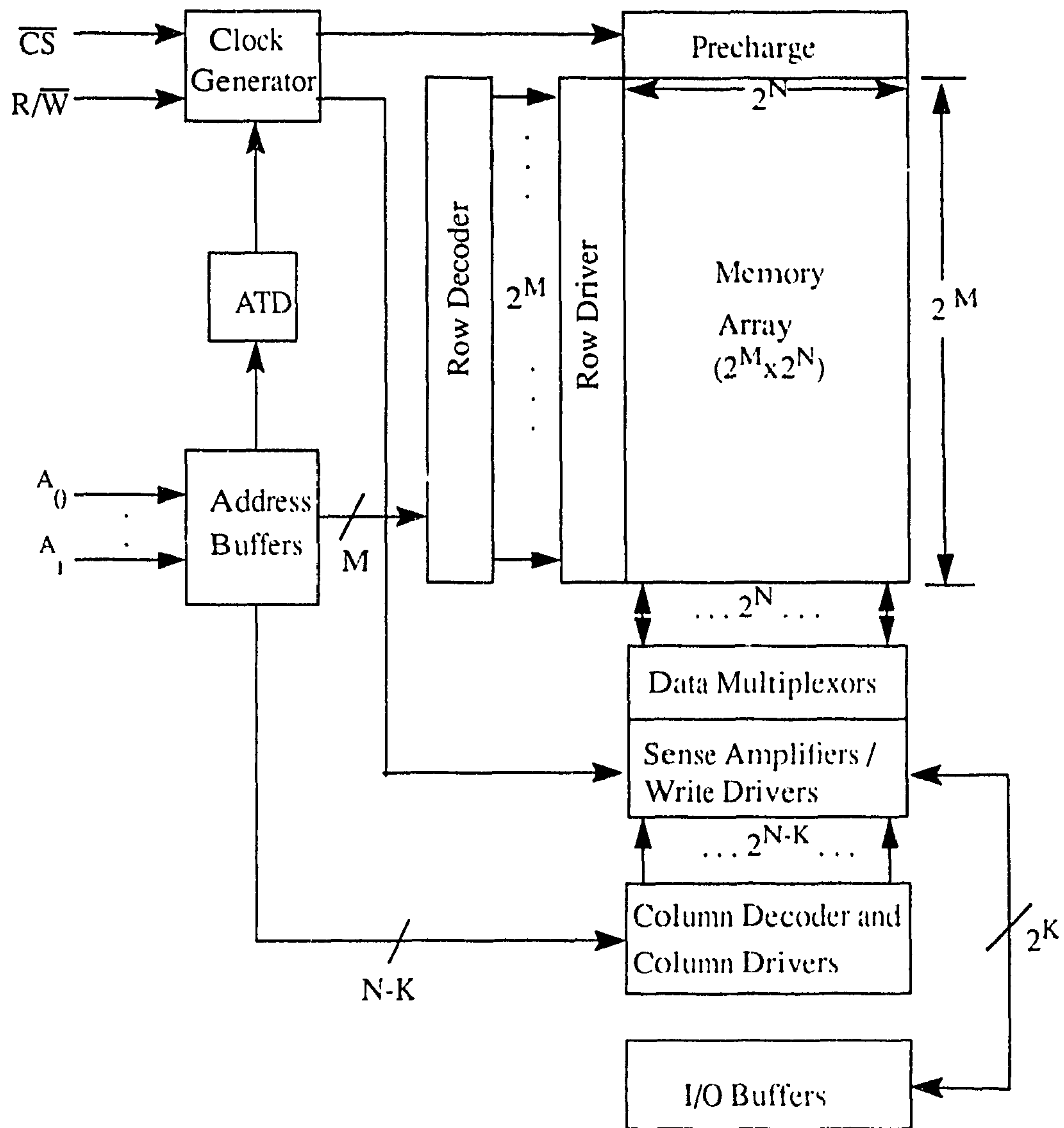


Fig. 1.1 A typical SRAM Architecture

the cell stability and expedite the read operation. The data multiplexors or column selectors select the column of the cell that is being addressed for read/write. The sense amplifiers are used to speed up the access (read) time of the memory. The clock generator generates the internal timing control signals for the precharge circuit and the sense amplifier. The address transition detection (ATD) technique is used to enhance the control of the chip. The ATD pulse generator triggers the clock generator to provide internal synchronization. The function of the I/O drivers is to buffer the data to or from the external world.

In Fig 1.1 we assumed a single block SRAM. However, in a chip there may be single or multiple blocks of cell arrays with other peripherals. The total number of blocks in a chip depends on the word size, total memory capacity, performance measure (access speed, power consumption) and layout issues. A number of different architectures and floor plans can be found in the literature (see for instance, [1-12, 28-30]).

1.4 Static Random Access Memory (SRAM) cells

A SRAM cell in general may be of a four-transistor (4T) or a six-transistor (6T or full CMOS) type as shown in figure 1.2.(a) & 1.2.(b) respectively. The selection of any of the 4T or 6T configurations mainly depends upon two aspects such as, cell area and stability of the cell. Those two aspects are inter-related, to ensure the stability of the cell it is required to increase area. The total cell area in a SRAM occupies at least 50% of the overall chip area. The cell stability determines the Soft-error Rate[15] for the case of R-load cell and Static-Noise Margin.

The 4T (R-load) cells are used in some high density SRAM design. As in Fig. 1.2(a) the pull-up resistance is made with a special process requiring low area. But as reported in [9, 13] the Static-Noise Margin (SNM) of the R-load cell becomes much lower than the full CMOS one at low supply voltage. So, to have sufficient noise margin the R-load cell has to be made larger. Also, for a high density SRAM the resistance of the cell R-load must increase in order to maintain a constant SRAM standby current. As a result, it becomes

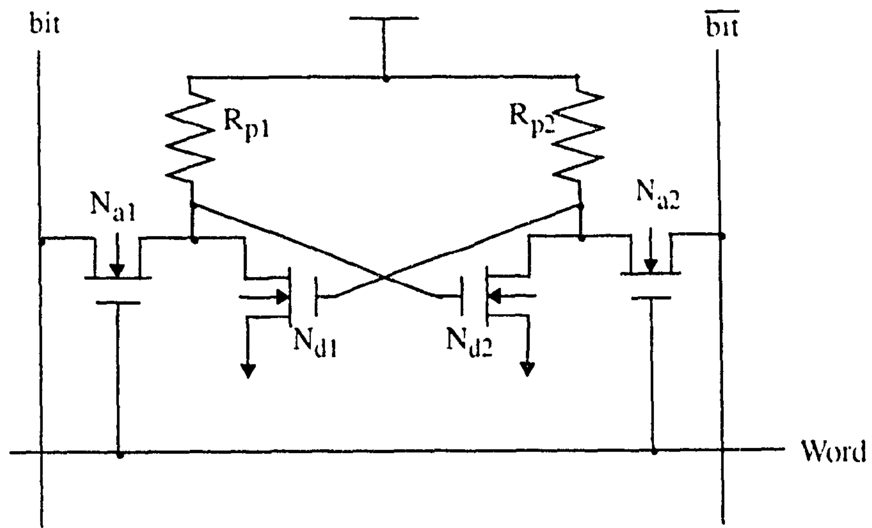


Fig. 1.2(a) A 4-transistor SRAM cell

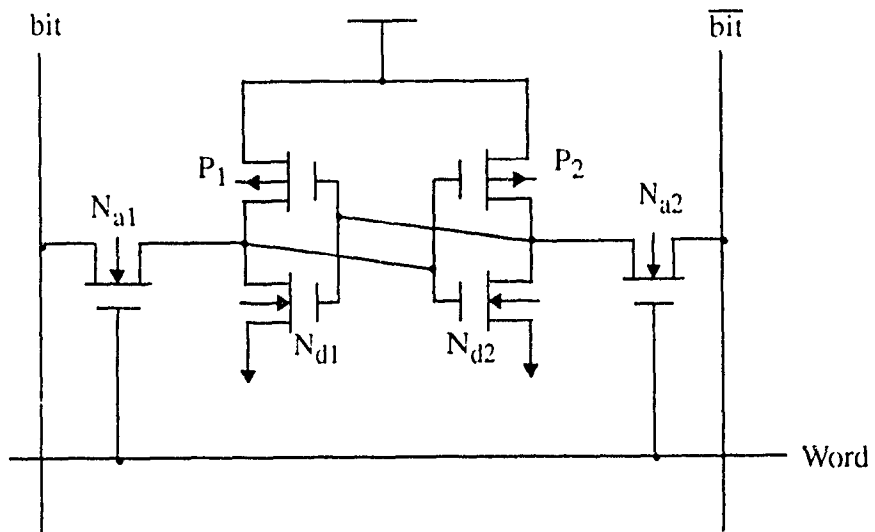


Fig. 1.2(b) A 6-transistor SRAM cell

more and more difficult for a R-load cell to retain cell data. This is because the resistance of the R-load becomes so high that the R-load can not supply enough current to hold the high-level voltage of the memory cell node.

Comparatively, the 6T cell is more robust and stable because of the presence of a PMOS load transistor which supplies sufficient leakage current to hold a stable cell state. The following are the advantages of a 6T SRAM over a 4T SRAM:

1. Higher noise margin.
2. Excellent data retention capability because of the active pull-up PMOS transistors.
3. No D.C. path exists in the cell, thus very low standby current can be obtained.
4. Since the cell stability can be achieved easily, the speed optimization can be achieved by increasing access transistor width.

1.4.1 Organization and Operation of a SRAM

A functional description of the memory operation is depicted in Fig. 1.3(a), showing the general organization of a SRAM. Fig. 1.3(b) illustrates a simplified circuit diagram for read and write operation of a SRAM. The bit and bus lines are precharged and equalized by the precharge circuit. The read operation is initiated by the precharge cycle. During precharge the row and column decoders and sense amplifiers are disabled. As shown in Fig. 1.3(b) we used our newly designed Power Down Y-controlled PMOS (PDYCP) load precharge circuit which is explained in Section 1.7.2. Here, the operation of the PDYCP circuit is controlled by address transition detection (ATD) and a column decoder. When ATD=1 and Y_i=0, the bit and bus lines are precharged to V_{dd}=5 volts. The address is decoded by two address decoders: row (X) and column (Y) decoder to select a cell for read/write. The word line is locally connected to all the cells which belong to the same row. The bit lines are laid into columns and connect all the cells which belong to the same column. The timing cycle starts with the chip select signal ($\overline{CS}$) as active low. The read/write control

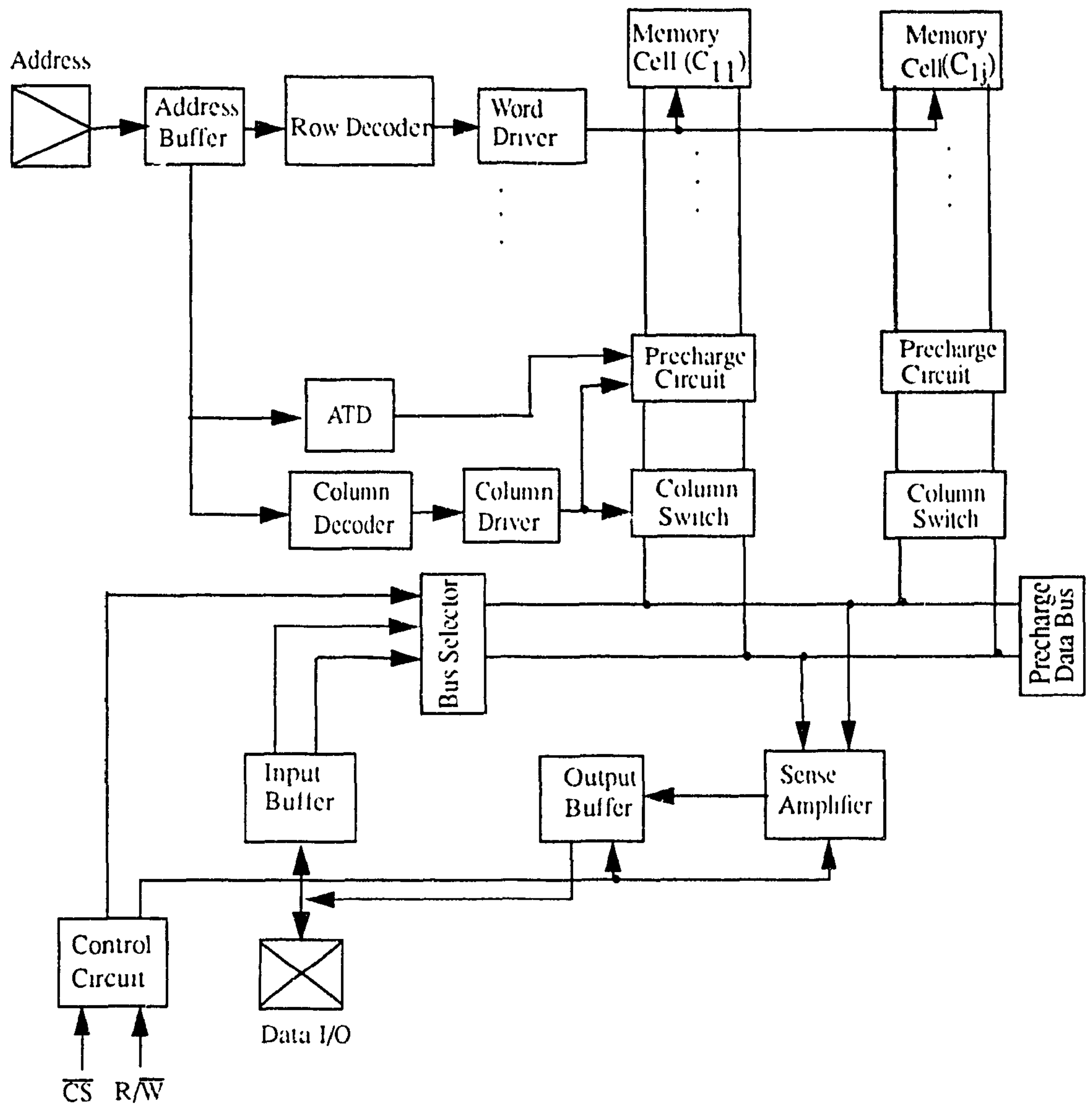


Fig. 1.3(a) Organization of a SRAM

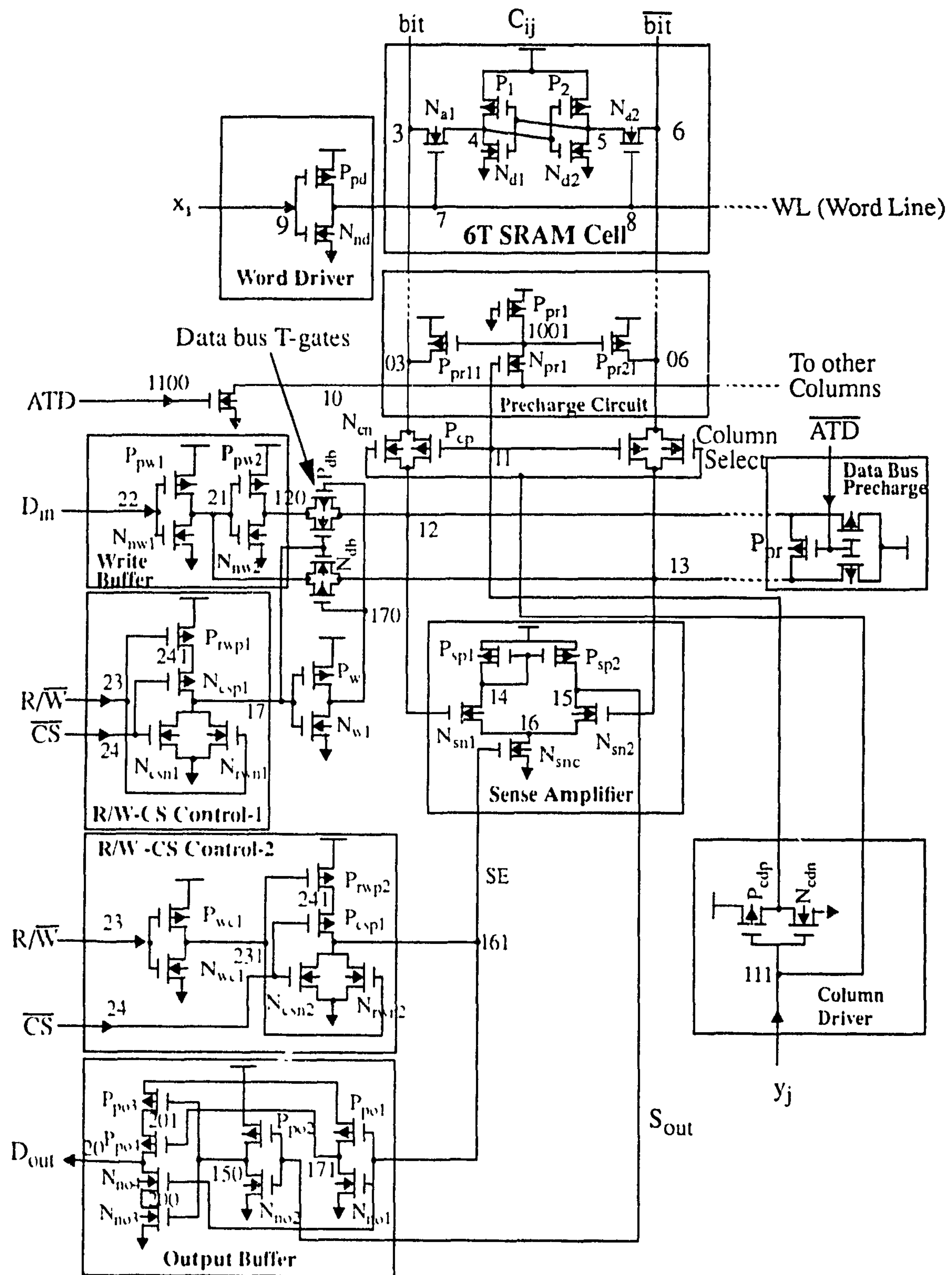


Fig. 1.3(b) SRAM with peripheral circuits

signal ($R/\overline{W}$) determines a read/write operation. For the sensing scheme we used a differential current mirror sense amplifier (CMSA) because of its high differential mode gain, low common mode gain and good common mode noise-rejection ratio (CMNRR). As soon as a small difference in potential between a pair of bit lines happens, it is amplified by the sense amplifier in a full swing and fed to the output buffer. An important issue here is the position of the column selectors. If the column selectors were placed after the sense amplifiers, more amplifiers would be required which increases power dissipation and area. As shown in Fig. 1.3(b), multiple bit lines are connected to the common data bus and feed to a single sense amplifier which is connected to the data bus. In high speed SRAM there is usually one or a few columns dedicated to a sense amplifier. Finally, data output is fed to the output pin by an output buffer. Input data comes through the same pin and goes through the input buffer. The sense amplifier is bypassed during a write operation, and the write buffer directly feeds the memory cell selected by the column and row decoders.

SRAM cell operation:

The key element in SRAM is the SRAM cell which can store information in its static structure as long as power is available in the chip. As in Fig. 1.2(b), the cell, being a cross-coupled inverter, with node 4 (S_1) containing a '0' or '1' the complementary value will be stored at node 5 ($\overline{S}_1$). Here, the transistors N_{a1} and N_{a2} are called access transistors, N_{d1} and N_{d2} are called driving transistors. The operation of the cell is as follows: when the word line is selected then the access transistors N_{a1} and N_{a2} conduct. The cell can be in one of the two stable states, either '0' or '1'. For state '0', the transistors N_{d1} and P_2 are in conduction but N_{d2} and P_1 are in non-conduction. In this case the voltage at nodes 4 and 5 will be '0' and '1' respectively. On the other hand, for state '1', transistors P_1 and N_{d2} will be conducting and P_2 and N_{d1} will be non-conducting which will ensure an '1' and a '0' at nodes 4 and 5 respectively.

Reading Data:

The read operation consists of two steps. First, the bit lines are precharged high keeping the word line disabled. As soon as the precharge time is over the word line and the column select line is enabled to address the desired cell to be read. Consequently, the $\overline{CS}$=0 and R/$\overline{W}$=1 signals are also set. The charge on one of the bit lines of the selected bit line pairs will be discharged through the enabled memory cell, representing the state of the cell. For instance, assume the cell C_{ij} as shown in Fig. 1.3(b) is selected to read. The word line signal at node 7 and the column select signal at node 111 goes high which makes transistors N_{a1}, N_{a2} and column select transmission gates conduct. If a logic '0' is stored in the cell (meaning node 4 ='0' and node 5 ='1') then the current flows through N_{a1} which will pull down the 'bit' line. If a logic '1' is stored in the cell (meaning node 4='1' and node 5='0') then the current would flow through N_{a2} and thereby pulling down the '$\overline{bit}$' line. Since, at the beginning of the cycle, $\overline{CS}$= 0 and R/$\overline{W}$= 1 which makes 'SE' line high and selects the sense amplifier. The sense amplifier is connected to the data bus, for a '0' read as the 'bit' line will be pulled down towards the ground then the differential voltage at the bit line pair will pull down sense amplifier's node 15 towards ground and thereby sensing a '0' which will be buffered at the output driver. For a read '1', the 'bit' line will stay high and the '$\overline{bit}$' line will be pulled down towards the ground which will result in pulling up node 15 towards V_{dd} (V_{dd}=5 volts: logic '1'). Fig. 1.3(c) illustrates a read-0 operation. Detailed operation of a CMSA is referred to Section 1.8.1.

Writing Data:

The write cycle starts by setting ATD=0, $\overline{CS}$=0, R/$\overline{W}$=0, WL=1, and Y_j=1. Now, to write a '0' in the cell, D_{in} should be set to '0'. Therefore, data '0' and '1' will be placed to the 'bit' and '$\overline{bit}$' lines respectively. Since the cell is already selected to write, the data placed in the bit lines will be written to the cell with nodes 4 and 5 at states '0' and '1' respectively. If the cell had a '1' stored in it then the new data will overwrite on it to '0'. In

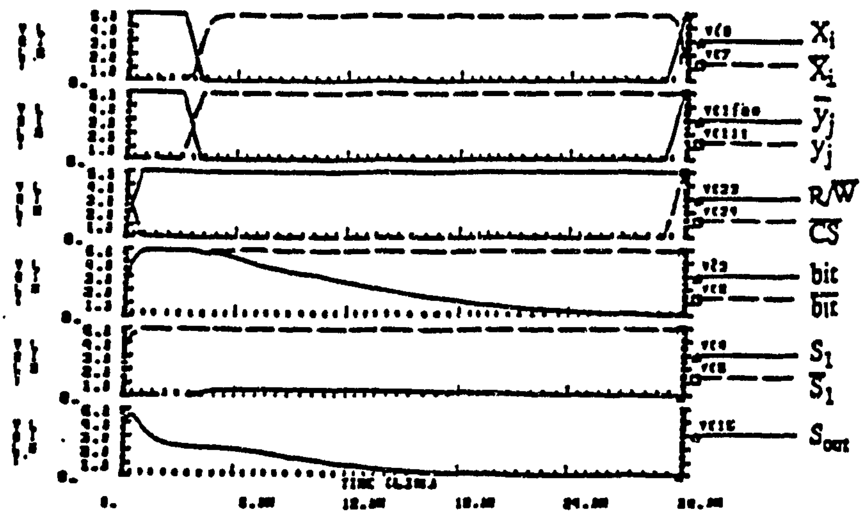


Fig. 1.3(c) Read- 0 operation of a SRAM

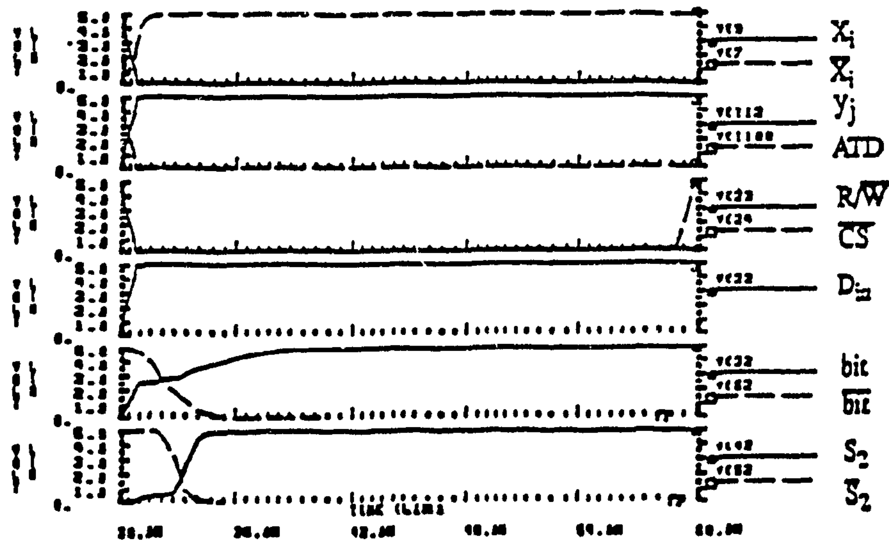


Fig. 1.3(d) Write-1 operation cf a SRAM.

this case transistors N_{d2} and P_1 will be switched off and N_{d1} and P_2 will be switched on. If the cell had a'0' stored in it then with 'bit'=0, it will keep the storage node 4 at '0' and thus the state of the memory will remain unaltered. If an '1' is to be written in the cell, regardless of what is stored in the cell, the D_{in} (data input) is required to provide '1' which will set 'bit'=1 and '$\overline{\text{bit}}$'=0. Since the cell is selected to write, transistors N_{a1} and N_{a2} will be switched on through word selection and logic '1' and '0' will be stored at nodes 4 and 5 respectively which confirms a write '1' into the cell. In this case, cell transistors N_{d2} and P_1 will be switched on and N_{d1} and P_2 will be switched off. If the cell originally had an '1' stored in it then the state of the cell will remain unchanged. Fig. 1.3(d) illustrates a write-1 operation. Where S_2 and $\overline{S}_2$ are the cell storage and complementary storage nodes respectively.

1.4.2 A 6T SRAM Cell Design

Optimum design of the SRAM cell has a significant effect on the performance of a SRAM. As discussed in Section 1.4 the cell area and stability of the cell are considered to be two important aspects in designing a SRAM cell. The cell area of a SRAM is determined by the minimum feature and the noise margin [16]. The noise margin of the cell depends on the conductance ratio of the cell driver and the access transistor. The conductance ratio also partly determines the speed performance and power dissipation. Therefore, optimization of the memory cell in terms of device dimensions is important to obtain high-density SRAMs.

Cell stability is influenced by the channel width mismatches or threshold mismatches in all paired devices [16]. In the typical SRAM cell as in Fig. 1.2(b) the PMOS pull-up transistors can be minimum sized since its only function is to offset the effects of leakage. Therefore, the design of the cell depends on the sizes of access and driver transistors. Their size ratio will affect the read/write speed. The size of the cell access transistor directly affects the load on the word and bit line. In order to design a high speed, low power and highly reliable

SRAM the following design criterion as proposed in [16] should be taken into account:

1. Nondestructive Read condition
2. Write condition
3. Data retention condition
4. Power dissipation condition.

Nondestructive Read Condition:

During a READ operation bit lines are precharged high to a voltage $\leq V_{dd}$=5 volts. Fig. 1.4(a) shows the graphical representation of Static Noise Margin (SNM) which is obtained by drawing and mirroring inverter the characteristic [13]. As in Fig. 1.3(b) assume nodes 3 and 6 are precharged high. Node 4 contains data with complementary data at node 5. Now, after precharge, when the word line is selected to READ the cell, a transient disturbance at soft nodes 4 and 5 may occur which can corrupt stored data. The term nondestructive read refers to the SNM of the cell. The SNM of the cell should be high enough to cope with the transient READ disturbance. The SNM of a cross-coupled inverter is defined as the maximum noise voltage V_n that can be tolerated by it before changing states [13]. A SRAM should be designed such that under all conditions some SNM is reserved to cope with dynamic disturbance caused by α particle, cross-talk, voltage supply ripple and thermal noise [13]. The static noise margin of the cell is determined by the β (transistor gain factor) ratio, r= β_d / β_a of the driver and access transistor respectively. Where, $\beta=\mu C_{ox}W/L$, μ is the mobility, C_{ox} is the gate oxide capacitance and W and L are the transistor channel width and length respectively. SNM is also related to the stability of the cell. Higher noise margin ensures better stability of the cell. The stability margin Δ is defined as the difference between the minimum voltage applied on the bit line to WRITE into the cell and the maximum voltage delivered on the bit lines by the cell during READ operation [9]. Δ must be positive to guarantee the functionality of the RAM [9]. Fig. 1.4(b) shows the stability of a cell as a function of r. It is seen that the stability margin of a 6T cell

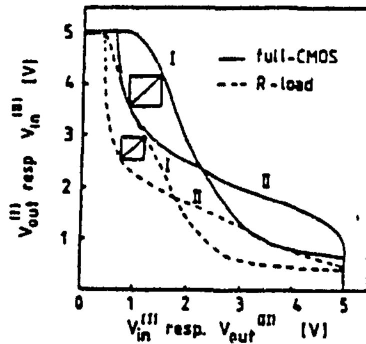


Fig. 1.4(a) Graphical representation of Static Noise Margin [13].

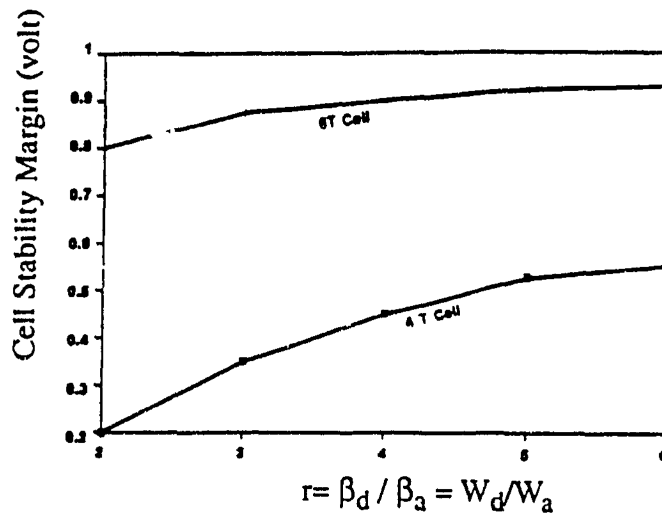


Fig. 1.4(b) Stability margin as a function of r [9].

is almost uniform and higher than a 4T cell because of the presence of the PMOS pull-up structure. According to our SPICE simulation and Fig. 1.4(b) an optimized design of a 6T cell can be done for a value of r=2 to 3.

Write condition:

During write operation the bit line pair and the word selecting the cell are enabled. Write data is placed on the 'bit' line and a complementary value is placed in the '$\overline{\text{bit}}$' line. The critical situation happens for the cells which belong to the same row of the cell under 'WRITE' but in other columns. To avoid miswrite to those unselected cells the ratio r becomes crucial. Nevertheless, it is observed that write condition does not affect stability of the unselected cells because of PMOS pull-up in 6T cell. In a conventional Y-controlled bit line precharge SRAM the bit lines of the unselected column are precharged high. Due to precharging of all unselected columns the write power dissipation increases. In our design, we propose a precharge technique which will avoid precharging of the unselected columns and thereby a significant reduction of power can be achieved and the cell will become more reliable.

Data Retention Condition:

The data retention in a memory cell means that the cell remains undisturbed by the bit line voltage set in the preceding read cycle. At the preceding read cycle one of the memory cells in a column is selected by a word line. As a result, the bit line is driven by the memory cell and one of a pair of the bit lines is set to a low voltage. In the unselected columns the bit line load precharge transistors act as a pull-up device. Since the column select transistors in the unselected bit lines are off, then the high-impedance pull-up results in a large voltage swing in those bit lines. The low level of the bit line in this period is lower than that of the column selected for read operation. The low level varies according to the channel width of MOS transistors of the memory cells in column. The lowest level of the bit line has to be set higher than the largest V_{FL} of memory cells in the column to retain the data. Where,

V_{FL} is defined as the bit line voltage at which the data changes from a high state to a low state when the voltage of a bit line is set under static condition.

Power dissipation condition:

In each column DC current flows into the memory cell connected to the selected word line. The power dissipation of the memory cell mainly depends on the size of the access transistors. As the bit line and input/output bus line are discharged by the current of these transistors, they determine the access time. Therefore, power dissipation is also a limiting factor in the design of the cell.

The optimum sizes of the access transistor and the inverter transistors are determined by the four conditions mentioned above and by design rules. As we have investigated, the above conditions mostly depend upon the cell ratio r and the precharge technique. The optimum value of r and our newly designed precharge technique will provide excellent results. In our design, the cell PMOS and access transistors are considered to be minimum sized according to the technology and design rule with an optimum value of r=2.5 because of the following reasons:

1. The function of the PMOS transistor is to supply little leakage current.
2. Since r is fixed, an increase in W_a also increases W_d as r times of W_a. So, the overall SRAM array area increases as an O(mn). Where m and n are the number of rows and columns respectively in the SRAM array.
3. Increased W_a may reduce cell intrinsic access delay but word delay will be increased as an O(n).
4. Increased W_a will increase bit line capacitance which will increase bit line delay as an O(m).

1.5 Decoding Scheme

The word decoding delay time can be defined as the delay time between the address setup time and word line selection time. The implementation of a decoding scheme is relative to the density of the RAM. For low density SRAM the simple conventional decoder [5] is sufficient. But as the memory density increases the word delay affects the total access time. For medium and high density SRAM's to have faster word selection time, dynamic double-word line (DDWL), and divided word line (DWL) structures are proposed in [10] and [11] respectively.

1.5.1 Dynamic Double-word line (DDWL) structure

The basic architecture of DDWL is shown in Fig.1.5(a). Word lines are doubly placed, namely Main Word Lines (MWL) and Section Word Lines (SWL) [10]. Due to the global arrangements of MWL, the capacitance of the MWL is relatively small, which reduces overall word line RC delay. Moreover, the power dissipation is reduced because only one SWL is selected at a time and consequently only a small number of memory cells are activated. From the view point of delay time and power dissipation better performance can be achieved with an increased number of sections/ blocks. But with the increase of sections the number of local decoders required is increased which increases the overall chip area linearly. Fig.1.5(b) and (c) shows the trade-off of delay, power and area as a function of the number of sections or blocks. A 16 or 32 blocks can be a reasonable choice of better trade-off. The clock-controlled word lines are one of the main features of the DDWL scheme [10]. In this scheme, the activation signal, ϕ_{wa} is created by address transition detectors (ATD). The pulse width of ϕ_{wa} is made greater than the access time and is used to cut all the D.C. path in the SRAM after a read operation is over. Consequently, a low D.C. active power dissipation has been achieved with no power consumption after access time. This is known as automatic power down (APD) technique.

Fig. 1.5(a) Dynamic Double-word line (DDWL) structure [10]

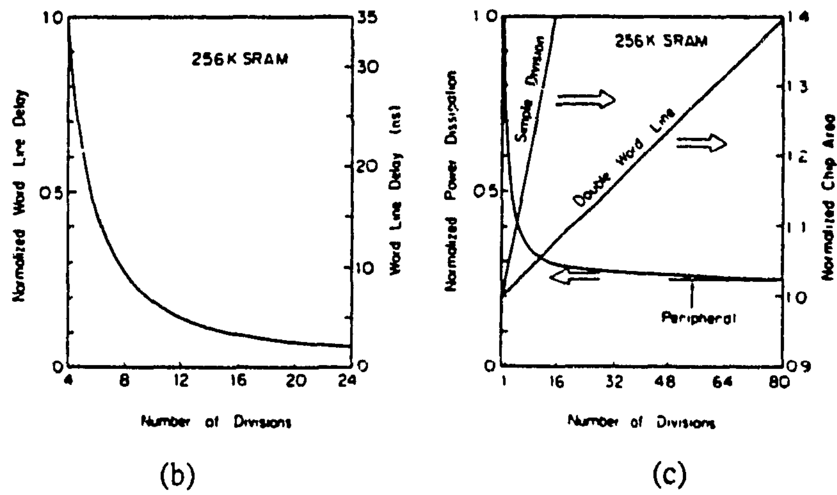


Fig. 1.5 Trade-off (b) Word line delay, (c) area and power as function of number of sections or blocks in a DDWL structure [10].

1.5.2 Divided Word-line (DWL) structure

The concept of conventional divided word line architecture as proposed in [11] is illustrated in Fig. 1.6(a). The architecture is similar to that of the DDWL except that it is not clock controlled. In this scheme each block contains N (N= n_c / n_b, where n_c= total number of cells in the memory, and n_b= number of blocks) number of cells. A local word line is placed in each block, is activated by a global word line and block select line. Since only one block is activated to READ/WRITE the DWL structure reduces both word-line delay, and power consumption.

For a high density SRAM with a capacity of 4MBit or more, the number of multi-divided blocks will have to be increased. In this case, the DWL structure suffers from increased load capacitance of the global word line and causes a significant increase of both delay time and power consumption. Newly, hierarchical word decoding (HWD) architecture has been proposed in [3]. In this architecture, the word select line is divided into more than 3 levels as shown in Fig.1.6(b). The number of hierarchies, in other words, the level of word-line division, is determined by the total load capacitance of the word decoding path.

With HWD, the load capacitance of the word decoding path is efficiently distributed [3]. Therefore, the HWD architecture realizes a significant reduction in both delay and power consumption. Fig.1.6(c) shows the simulation results as in [3] of word decoding delay and total load capacitance of the word decoding path as a function of storage density. The conventional DWL and the proposed HWD are compared with each other at the optimum point of 256- Kb, 1- Mb, and 4- Mb SRAMs.

In a smaller density, like a 256-Kb SRAM, there is no significant difference. As the density becomes larger, although the HWD needs an extra decoding stage compared to the DWL, it shows better performance than that of the DWL. In a 4-Mb SRAM, for example, the HWD architecture can reduce the delay time 20% and the total load capacitance 30%

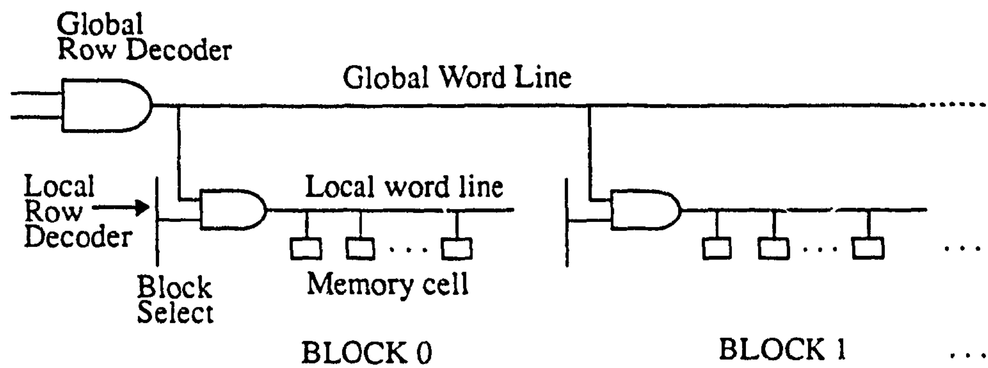


Fig. 1.6(a) Conventional Divided-word line structure [3]

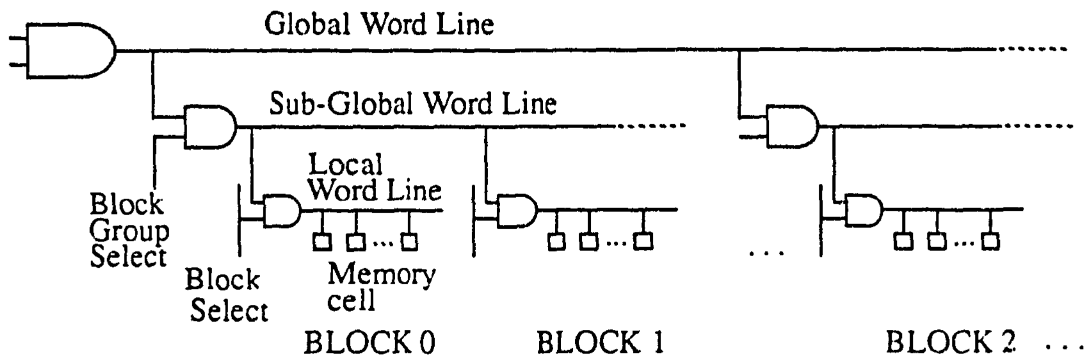


Fig. 1.6(b) Hierarchical word decoding architecture [3].

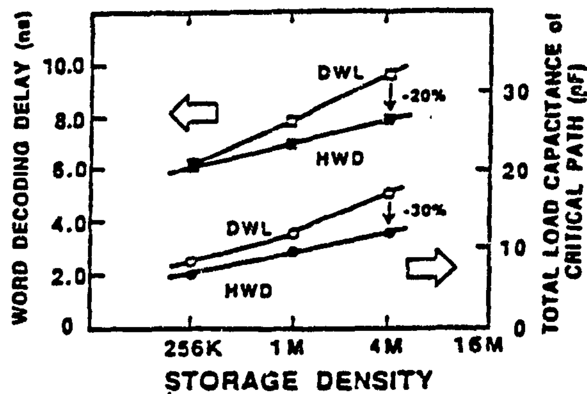


Fig. 1.6(c) Comparison of DWL and HWD [3].

compared to the conventional DWL [3]. The estimation predicts that the proposed HWD architecture will be very effective for future high-density SRAM.

The advantages of the above two schemes are as follows:

1. Elimination of wasted column current. The column currents flows only in a selected block which reduces total active power.
2. Reduced delay time. Due to 2 or 3 levels of hierarchy of the decoding technique the capacitance of the row select line is much smaller than that of a conventional word line, because it includes gate capacitance of every access transistor in the memory cells.

The drawback of DWL/HWL is that the column current in the selected block flows during the whole read and write cycle which increases active power. This can be overcome by using a block isolating circuit. The purpose of the block isolating circuit will be to disable the block when the sense amplifier senses the differential voltage. The signal produced by ATD circuit can also be used for this purpose.

1.6 Circuit techniques for Decoder

As we see in the above DDWL & DWL structures, an increase in the number of blocks and / or hierarchy increases the requirement of the number of decoder circuit linearly. Therefore, an optimized decoder circuit technique is very effective to reduce overall chip area. Two possible optimized decoder circuits namely, PMOS-load decoder [2],[5] and Transfer Word Decoder (TWD) [1] can be noted for high density SRAM applications in comparison to a conventional CMOS AND / OR decoder.

1.6.1 PMOS-Load Decoder

Figs. 1.7(a) & (b) show a PMOS-Load decoder [5] and a conventional CMOS

(a) (b)

Fig. 1.7(a) PMOS load decoder and ,

(b) Conventional CMOS decoder [5]

Word Decoder	$\frac{C_{gate}}{C_{wire}}$
PMOS Load	1
CMOS	2

Fig. 1.7(c) 1-Mbit SRAM word decoding scheme [5]

Fig. 1.7(d) Circuit schematic of word decoder using PMOS load [5].

decoder respectively. The address signal line is connected only with the NMOS in a PMOS load decoder, while it is connected with both the PMOS and the NMOS in a CMOS decoder. Therefore, the address load capacitance caused by the gates in PMOS-load is almost half of that in a CMOS decoder. This implies that a smaller delay time can be obtained by using a PMOS-Load decoder. Fig. 1.7(c) shows a simplified 1- Mbit SRAM word decoder scheme which has typical PMOS load circuitry. A word decoder is the final decoder to select a word line, which is connected with 64 cells. An address signal is connected with 128 word decoders. As those 128 word decoders are densely packed in one cell array block, gate capacitance makes up a large portion of the total capacitance of the address signal line. In the 1- Mbit SRAM [5] the ratio of C_{gate} to C_{wire} is about 1 for a PMOS-load word decoder and about 2 for a CMOS woıd decoder. A circuit schematic of a word decoder used in a 1- Mbit SRAM [5] is shown in Fig. 1.7(d). Eight NAND gates are designed by using PMOS-load circuits. The bottom NMOS is used common to the eight NAND gates to reduce the word decoder area.

Fig. 1.7(e) shows the dependence of the delay times on the ratio of C_{gate} to C_{wire} for the two types of decoders. Apparently, the PMOS-load decoder produces more delay time than the CMOS decoder for the same capacitance ratio. However, in actual circuitry, because of the difference of the capacitance ratios between the two types of decoders, the PMOS-load decoder yields a smaller delay time in most cases. As shown in Fig. 1.7(e), a 15% decrease in the delay time is reported in [5].

The PMOS-load decoder has a D.C. current due to the normally ON PMOS transistors, while the CMOS decoder has no D.C. current. This fact might discourage us from employing the PMOS-load decoder. However, we have to remember that the CMOS decoder circuitry has more ac current than the PMOS-load decoder because of its large capacitance [5]. Fig. 1.7(f) shows the dependence of the average current upon cycle times for the two types of word decoders. With a cycle time of more than 25ns, a PMOS-load word decoder has more current because the D.C. current of the PMOS-load decoder is

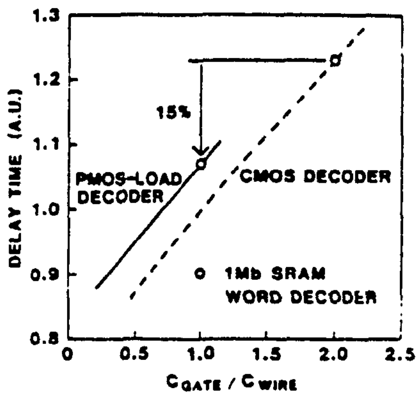


Fig. 1.7(e) Delay time of two types of decoders [5]

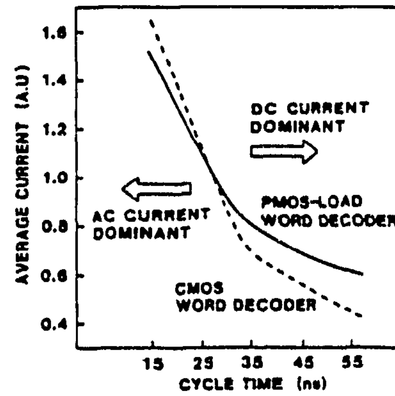


Fig. 1.7(f) Current of word decoder [5]

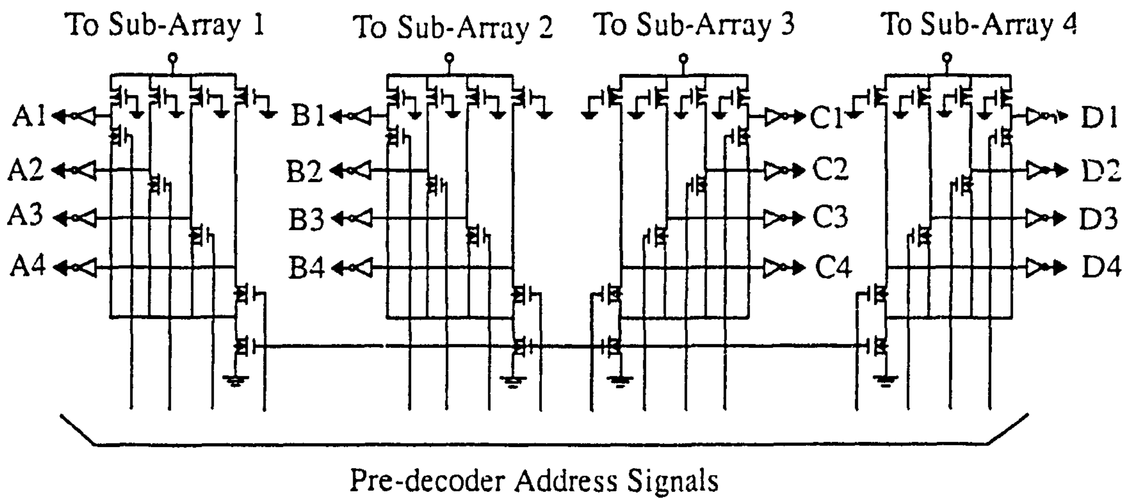


Fig. 1.7(g) Word-decoder circuit (for four memory subarrays) [2].

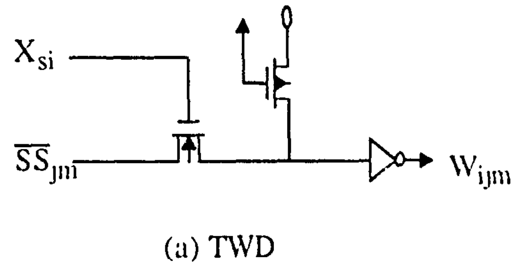
dominant. However, with a cycle time of less than 25ns, a CMOS word decoder has more current because the ac current charging and discharging the capacitance becomes dominant.

A similar PMOS-load decoder circuit is also implemented in [2], where the word decoder for a four memory sub-array are laid in one region as shown in Fig. 1.7(g). This architecture reduces 40% word decoder layout area by sharing a common diffusion region [2].

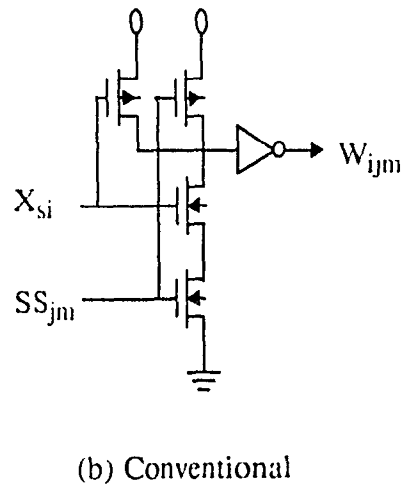
1.6.2 Transfer Word Drive (TWD) Decoder

As described before a conventional NAND decoder is not suited for high-speed high-density SRAM because of its high gate capacitance. A sophisticated decoder circuit called Transfer Word Drive (TWD) is proposed in [1]. Here, the conventional two input NAND gate is modified as in Fig.1.8. The principle of operation of the TWD decoder is such that when the main word line signal X_{si} is high and the selection signal $\overline{SS}_{jm}$ is low, the word line W_{ijm} is selected. This circuit is very simple and has a similar function to the AND gate.

By using TWD, the load capacitance of the selected signal can be reduced by around 30% [1]. The advantages of the TWD decoder are, its simplicity, reduced layout area and faster delay time. For an optimized decoder design, a TWD decoder or a similar type will be highly preferable.



(a) TWD



(b) Conventional

Fig. 1.8 (a) Transfer word drive decoder
(b) Conventional decoder [1].

1.7 Precharge Scheme

In this Section, we describe various precharge techniques for a SRAM. Some typical precharge circuits have been presented with their merits and demerits. We also present a Power Down Y-Controlled PMOS (PDYCP) load precharge circuit. The performance of PDYCP precharge technique is compared with a Y-Controlled bit line load (YCL) precharge. Our results show that the PDYCP precharge technique performs superior to any other technique.

1.7.1 Different Precharge Techniques

Faster precharge of the bit lines for faster read operation is an important issue. Various circuit techniques have been reported in the past ten years for the precharge technique of a SRAM. Accordingly, precharge circuits can be classified into two categories:

1. Continuous precharge of bit / bus line,

2. Controlled precharge of bit / bus line.

Fig. 1.9 & 1.10 shows some common form of precharge schemes. The continuous precharge scheme has the advantage of high current driveability which enables quick precharge. But, the serious disadvantage of it is that it increases total standby power due to continuous precharging of every column. For high density SRAM application, continuous precharge bit / bus line technique is impractical because of high power consumption. Instead, controlled bit / bus line precharge technique is preferred. A combination of both approaches can be used for better performance as in [3]. A bit line load with an Address Transition Detection (ATD) controlled precharge circuit [3] is shown in Fig. 1.11. Here, the bit line load consists of PMOS and NMOS transistors in parallel. The advantages of this type of structure are quick precharging and its voltage bump free characteristic. The Block Select and ATD signals generate a bit line equalize (ϕ_{EQ}) signal to logically

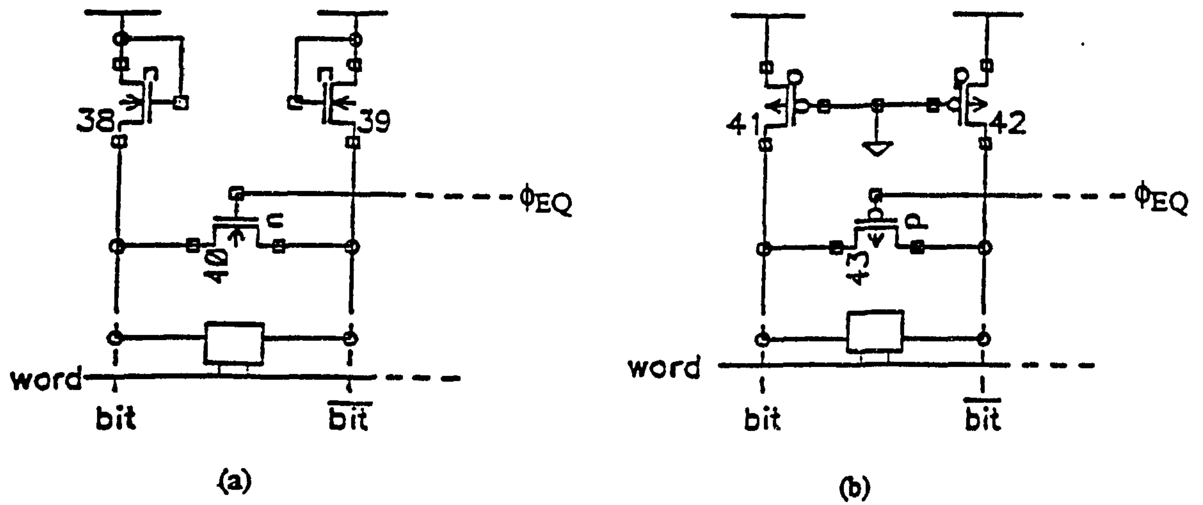


Fig. 1.9(a) and (b) Continuous precharge circuits.

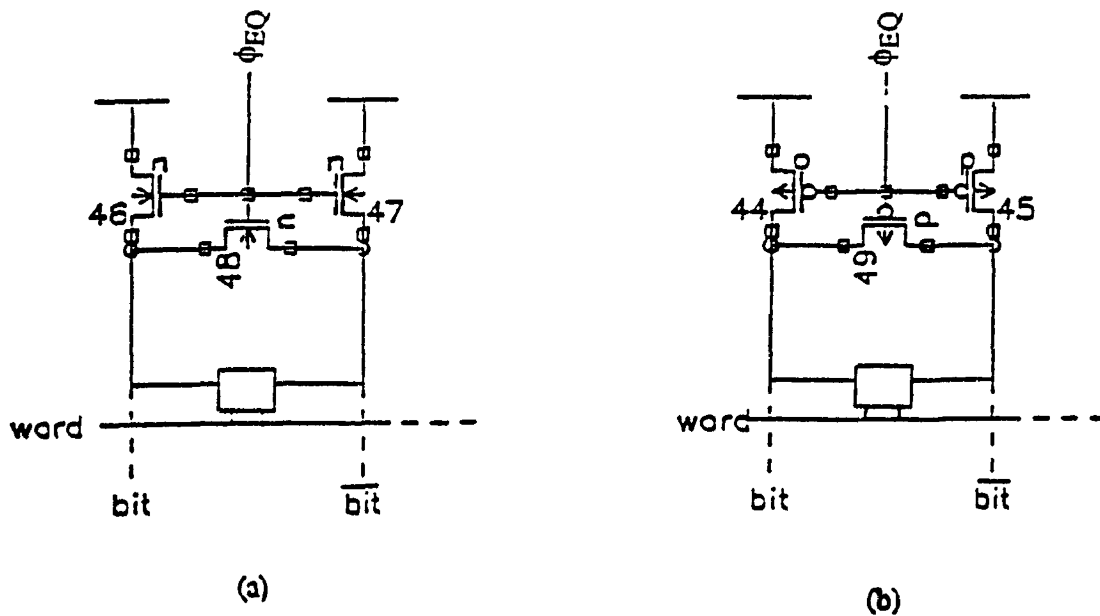


Fig. 1.10(a) and (b) Controlled precharge circuits.

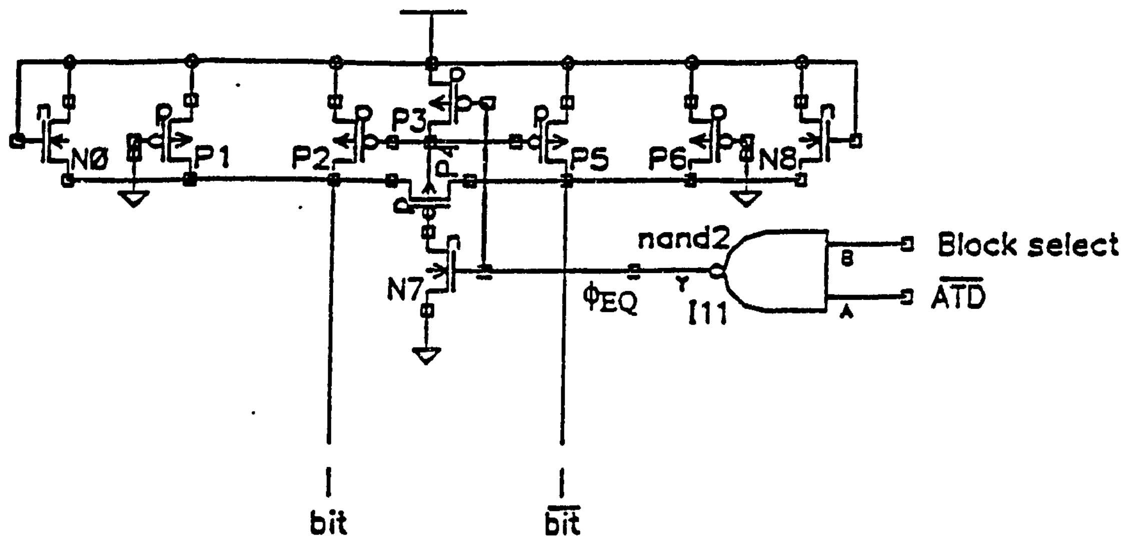


Fig. 1.11 Bit line load and $\overline{\text{ATD}}$ controlled precharge circuit [3].

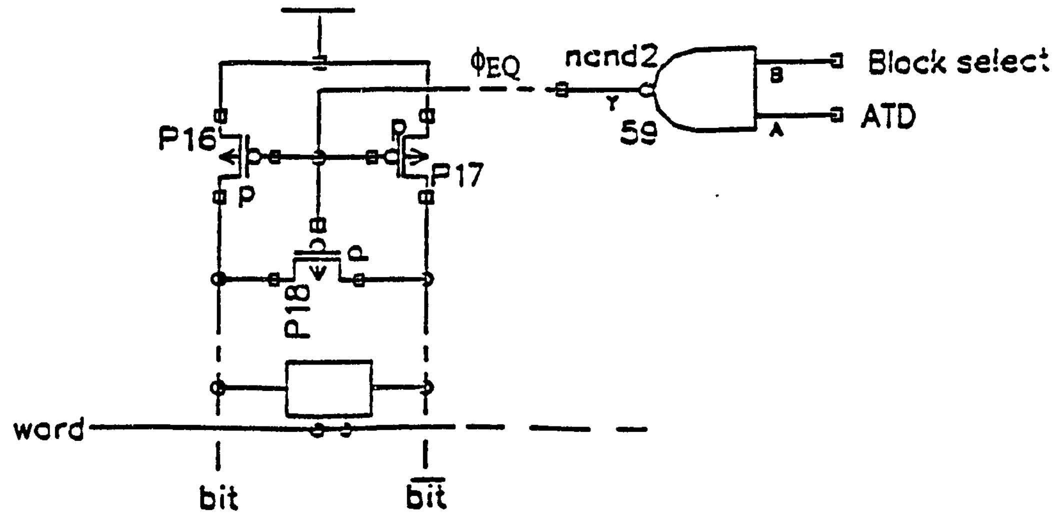


Fig. 1.12 $\overline{\text{ATD}}$ controlled conventional precharge scheme.

precharge and equalize the bit lines of the addressed block. The main difficulty with this method is that the unaddressed blocks are continuously precharged since transistors P1, P6, N0 and N8 as in Fig. 1.11 are always ON which results in high power dissipation. Fig. 1.12 shows a conventional fast precharge circuit scheme used in high density SRAM design. In this scheme, the precharge and equalize NAND gate experiences a very large precharge circuit gate capacitance because of precharging every column in the selected block which increases precharge delay.

A technique known as YCL (Y- Controlled bit line load) as shown in Fig. 1.13 is used in [1]. In this approach, precharge is controlled by the Y-select (Column driver) signal. When the bit line is unselected, PMOS transistors M_{np1} & M_{np2} turn on, precharging the bit lines. The precharge gate capacitance in this case is just only for a single precharge circuit. Now, when the column address is valid, PMOS transistors M_{np1} & M_{np2} turn off. However, the bit lines are still precharged and equalized by the data bus line precharge circuit.

Therefore, the precharge and equalize operation of the YCL circuit is the same as that of the conventional circuit without sacrificing any precharge capability. Moreover, the load capacitance of the precharge clock line ϕ_{EQ} generated by the row and column address transient detector has been reduced largely [1] because in this case the load consists of only the gate capacitances of the data bus precharge circuit.

The main advantage of YCL precharge is that faster precharge can be achieved. But, the problem with the unselected columns is still crucial. The unnecessary precharge of those columns increases power dissipation and in some cases may affect cell stability.

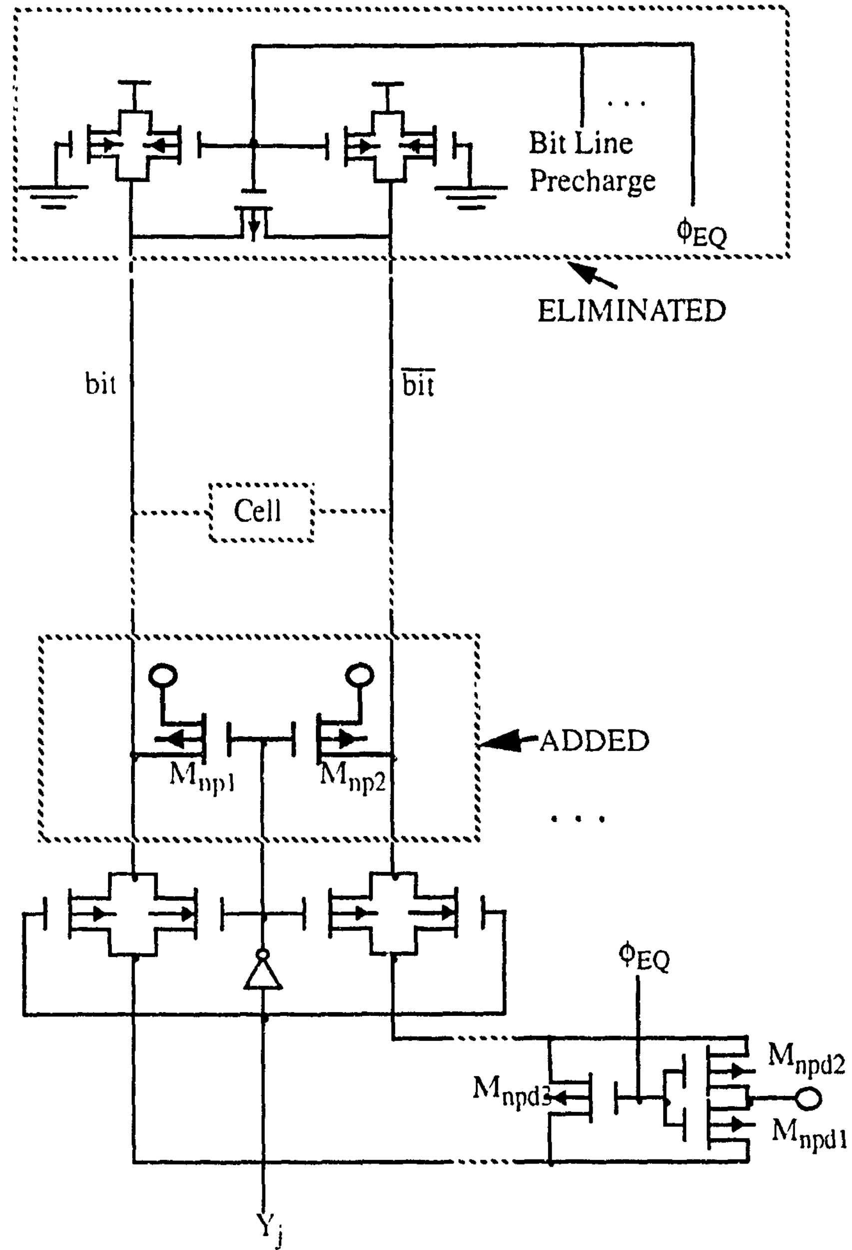


Fig. 1.13 Y -controlled bit-line load (YCL) precharge circuit [S.AIZAKI '90 [1]]

1.7.2 Power Down Y-Controlled PMOS (PDYCP) Load bit line Precharge Technique

The primary purpose of the precharge circuit in a SRAM is to accelerate the reading process. It is observed that most of the previous precharge techniques suffer from drawbacks in terms of speed and power consumption. The most recently reported YCL[1] (Y-Controlled bit line load) induces faster precharge but experiences huge amount of power dissipation due to the D.C. current in every unselected column under read/write or standby mode because of unwanted precharge. Considering the above situations we developed a modified form of YCL circuit called PDYCP (Power Down Y-Controlled PMOS) load precharge which is shown in Fig. 1.14. Here, we added a power down circuit (inside heavy line area) which is able to prevent unnecessary precharge of the column. The precharge is controlled by an Address Transition Detection (ATD) signal. The ATD signal will go high at the beginning of the precharge cycle. As soon as the precharge is over the ATD signal will go down. During precharge all the column addressing signals Y_1, Y_2, Y_3, ..., Y_j will be low and as a result transistors T_n, T_{n1}, T_{n2}, ... , T_{nj}, will be ON. Therefore, with ATD high, nodes $X_1$to X_j will be pulled down to the ground. The formal READ operation starts when the column is addressed with Y high and at the same time ATD will go down meaning that address is in transition. When ATD=0, the 'Deselect' line (Z) will be pulled-up through T_{pj}, T_{nj}. Assume that $Y_1=1$, $Y_2=Y_3=$ ... $=Y_j=0$, so as a result transistor T_{n1} will be OFF and T_{n2}, T_{n3}, ... T_{nj} will be ON. Now, the high voltage at Z and the grounded gate PMOS transistors will turn OFF the precharge PMOS transistors in every unselected column and the selected column. During standby mode all $Y_1=Y_2=Y_3=$ ... $=Y_j=0$, with ATD=0 which will disable the precharge PMOS transistors and thereby avoid unnecessary precharge.

The PDYCP circuit with other peripheral circuits in SRAM is given by Fig. 1.3(b). Fig. 1.15 depicts the SPICE simulation plot for the PDYCP precharge, READ and WRITE

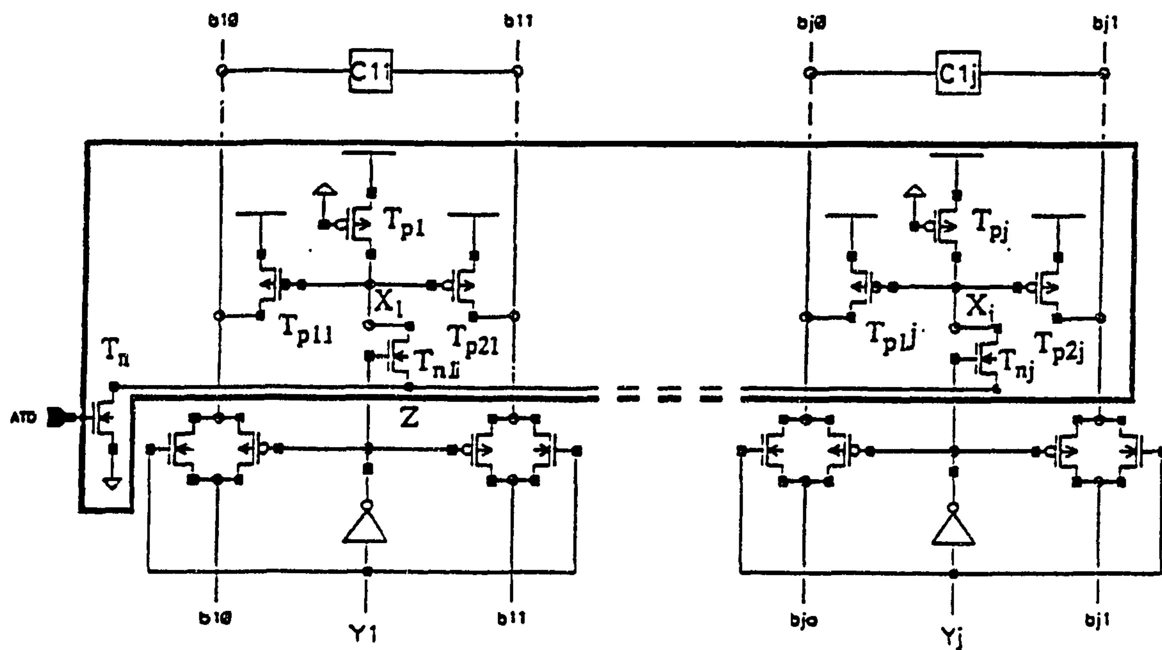


Fig. 1.14 Power Down Y-controlled PMOS (PDYCP) load precharge circuit.

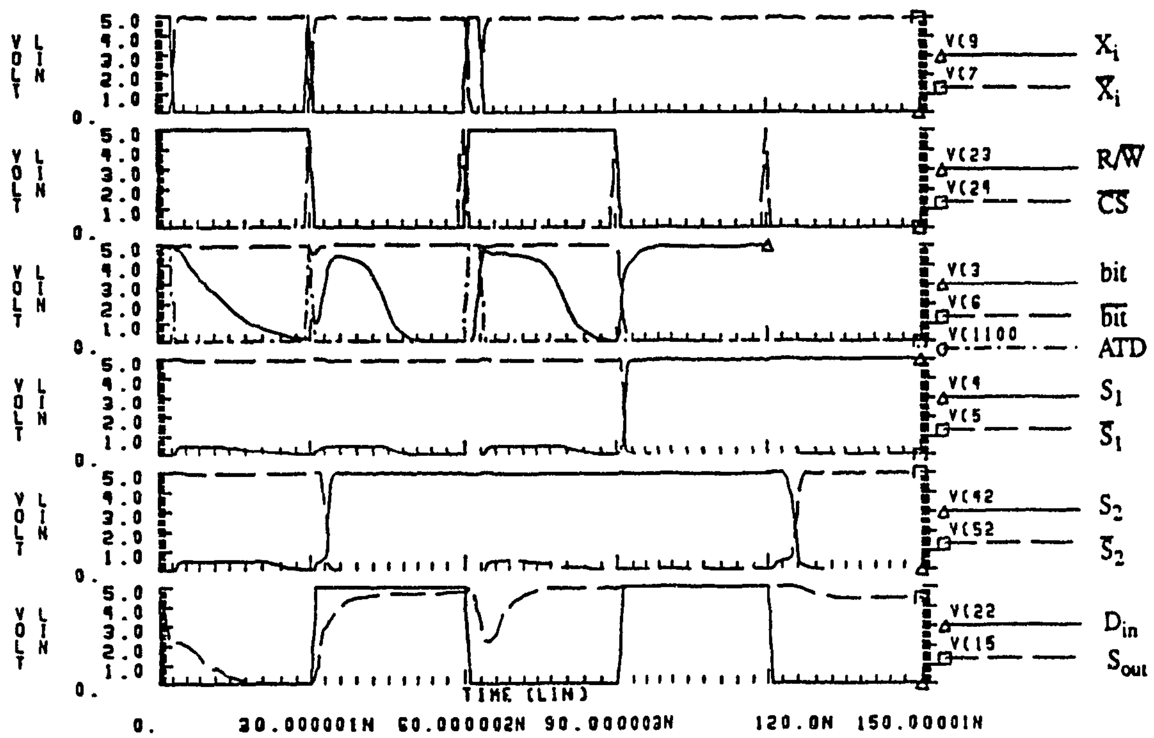


Fig. 1.15 SPICE simulation plots using PDYCP precharge technique for consecutive READ/WRITE operation.

operation. Here, we considered two cells lying in two different columns for READ/ WRITE operation. The functionality of the circuit is tested for any READ followed by READ/WRITE or vice versa. It is assumed that the READ cycle consists of precharge time plus the sensing time. As shown in Fig. 1.15 the READ/WRITE timing cycles are as follows:

0- 30 ns: READ-0, cell 11 30- 60 ns: WRITE-1, cell 12

60- 90 ns: READ-1, cell 12 90- 120 ns: WRITE-1, cell 11

120- 150 ns: WRITE-0, cell 12.

Where cell 11 is the cell at row 1 of column 1.

cell 12 is the cell at row 1 of column 2.

The symbols used for the SPICE plots in Fig. 1.15 are referred in Fig. 1.3(b). Moreover, we define,

S_1= cell 11 storage node which corresponds to node 4 of Fig. 1.3(b).

$\overline{S}_1$= cell 11 complementary storage node which corresponds to node 5 of Fig. 1.3(b).

S_2= cell 12 storage node which corresponds to node 42 of Fig. 1.3(b) (not shown).

$\overline{S}_2$= cell 12 complementary storage node which corresponds to node 52 of Fig. 1.3(b) (not shown).

In our PDYCP precharge scheme each precharge circuit contains two additional transistors over that of the conventional YCL precharge circuit. The PDYCP layout may look as it will take more area than that of the YCL, but in reality the wasted column area of YCL layout can efficiently be used in PDYCP implementation since the column width is fixed for a 6T SRAM. Our results show that with little or no layout area overhead the

PDYCP precharge scheme will be a best choice.

Table-I compares the performance of a PDYCP precharge with a conventional YCL precharge. It is clear that our PDYCP precharge technique will give very low power dissipation with faster precharge speed and will be very suitable for HDHSLP (High Density, High Speed, Lower Power dissipation) SRAM design. Further analytical and experimental comparisons are also available in Chapter-4.

mxn	cct type	T_{pr} (ns)	T_{sa0} (ns)	T_{sa1} (ns)	T_{w0} (ns)	T_{w1} (ns)	P_r (mw)	P_w (mw)
64x64	YCL	1.53	13.09	14.93	6.21	4.69	69.28	29.25
64x64	PDYCP	1.06	14.13	15.43	5.86	4.49	15.01	8.27

Table 1: Comparison of PDYCP & YCL precharge performance

As in Table 1 where,

m= number of rows, n= number of columns.

T_{pr} = Precharge delay.

T_{sa0} = Sensing '0' delay.

T_{sa1} = Sensing '1' delay.

T_{w0} = Write '0' delay.

T_{w1} = Write '1' delay.

P_r = Read cycle power.

P_w = Write cycle power.

1.8 Sensing Scheme

The sense Amplifier (SA) is the key circuit in the RAM to have a fast read access time. Plenty of research has been done by the RAM designers in the past decade ['80-'92] on sensing schemes. The implementation of a SA varies with the SRAM density. For a small scale SRAM a single stage SA is used. For medium and large scale SRAMs, a multi-stage SA with hierarchical architecture is used, which will be described later.

1.8.1 Operation of a Current - Mirror Sense Amplifier (CMSA)

The most commonly used of the SA circuits is the Current-mirror (CM) amplifier as shown in Fig.1.16(a). The advantage of this circuit is its fast sensing speed, large voltage gain and good output voltage stability. The SA circuit as shown is basically a differential amplifier. The transistor, N_{snc}, is a long channel device and acts as a current source. The sensing delay mainly depends upon the sizes of the NMOS transistors (W_{sn1}, W_{sn2} and W_{snc}). During the precharge condition both the bits lines are pulled up while the Sense Enable (SE) signal is low which ensures the current transistor N_{snc} is OFF. In this situation, transistors N_{sn1} & N_{sn2} are ON. Consequently, node 16 is pulled up because of capacitive coupling and as a result the output node 15 goes high as well. Now, depending on the cell content ('0' or '1') one of the bit lines will be pushed down with a high SE signal. The differential mode gain of the differential CMSA is such that a small differential input voltage (Say, 0.1 v) will be sensed at the output. If the cell contains an "1", the voltage at node 13 will start to drop through the RAM cell pass transistor and thereby turn N_{sn2} OFF. Since, node 12 stays high which keeps N_{sn1} ON and node 14 follows ground, then node 15 stays high and will imply a read '1'. If a '0' is stored in the memory cell then node 12 would be pulled down. In this case, transistors N_{sn1}, P_{sp1} & P_{sp2} would be OFF and N_{sn2} & N_{snc} would be ON which would pull down node 15 towards the ground representing a READ '0'. The operation of a CMSA is depicted by

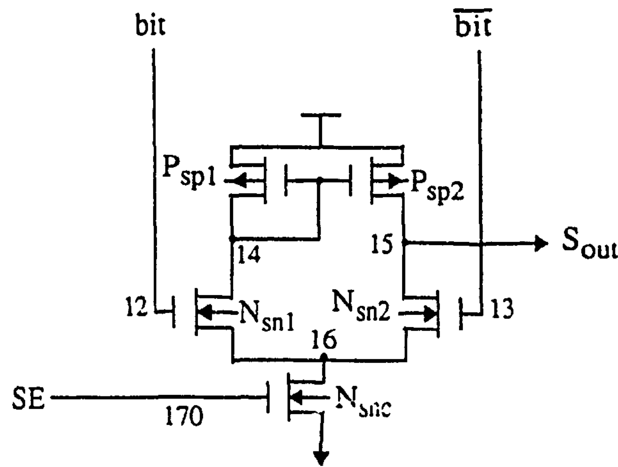


Fig. 1.16(a) A CMOS Current Mirror Sense Amplifier (CMSA)

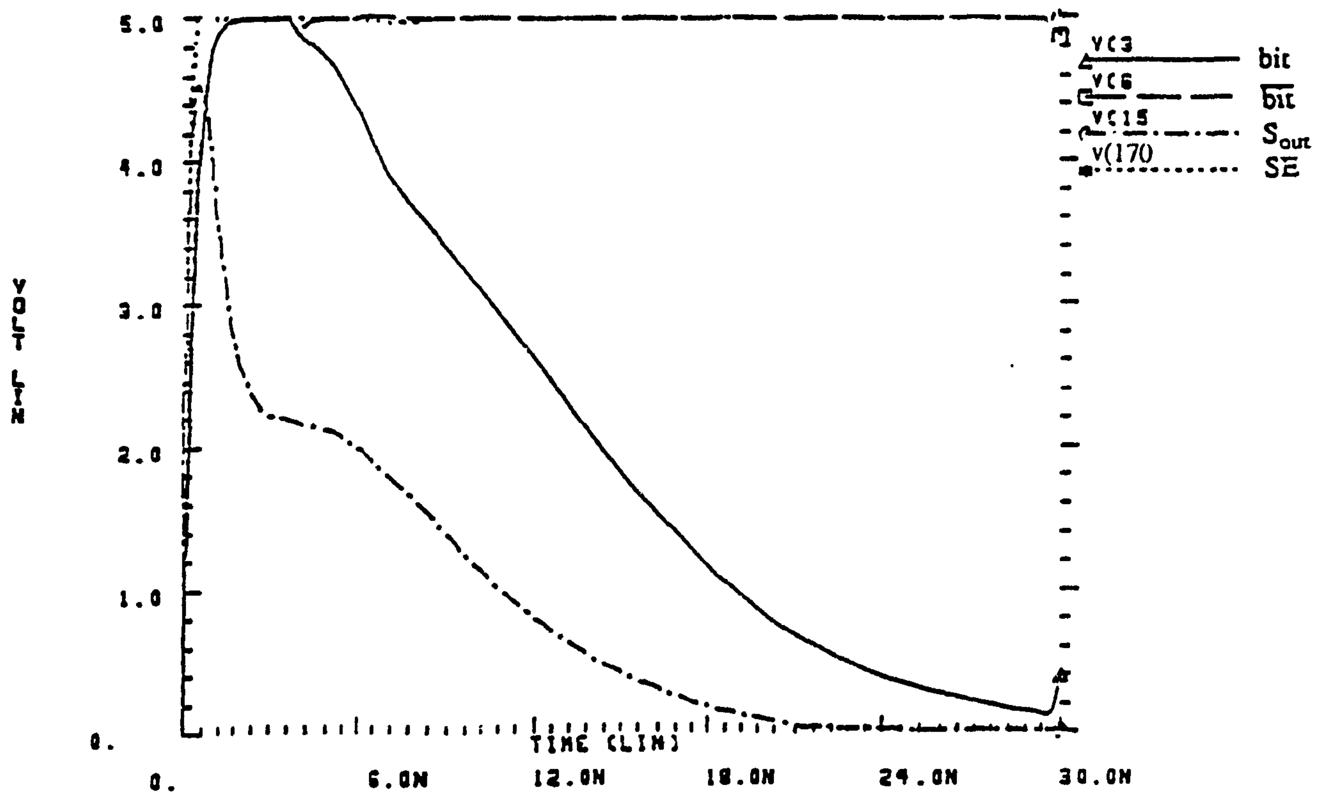


Fig. 1.16(b) Operation of a CMSA for a read-0.

Fig. 1.16(b) for a read-0 case. Here, V(3) and V(6) are the voltages at bit and $\overline{\text{bit}}$ nodes respectively. V(15) and V(170) are the sense amplifier output node and sense select (SE) node voltages respectively. The sense amplifier is selected with V(170) high. Since the cell contains a zero so, V(3) goes down and V(6) stays high. As a result, V(15) goes down towards the ground voltage (V_{ss}) meaning that the cell contains a zero.

1.8.2 Various SA Circuit Techniques

Various high speed SA circuits have been observed in the recent past. Some of them with their salient features will be presented in this Section. Fig.1.17(a) and (b) show the conventional CMSA and Input Controlled PMOS load (ICPL) [1] sense amplifier, respectively. As realized in [1],[2] the conventional SA is not suited for high- speed high-density SRAM applications because its gain is not enough for the high sensitivity needed for very fast access time. The ICPL sense amplifier as proposed in [1] includes two PMOS transistors as load in the conventional circuit which is shown in Fig.1.17(a). The four PMOS transistors act as active load and increase sensing gain [1]. It is reported in [1] that the ICPL sense amplifier has around 20% higher gain than the conventional sense amplifier. Another high speed and low power sense amplifier called Dynamic Gain Control Double end Amplifier [5] for a 1Mbit SRAM with 15ns access time which is shown in Fig.1.18. In this case a sensing delay (cell to data bus delay) of 4.5ns has been reported. The circuits within the dotted area are added to the conventional double-end currents-mirror (CM) amplifier. These additional circuits are used to vary the NMOS loading of the sense amplifier. A control signal, SAQ with a short pulse is used during the sensing operation, a fast sensing speed and low power dissipation is achieved simultaneously. A faster sense amplifier circuit so far has been reported in [29]. The circuit configuration is shown in Fig.1.19. This is a modified form of a two stage CMSA, first stage is connected to the input of the second stage. A voltage difference of 0.1v can be amplified and the sensing delay of approximately 1ns is reported.

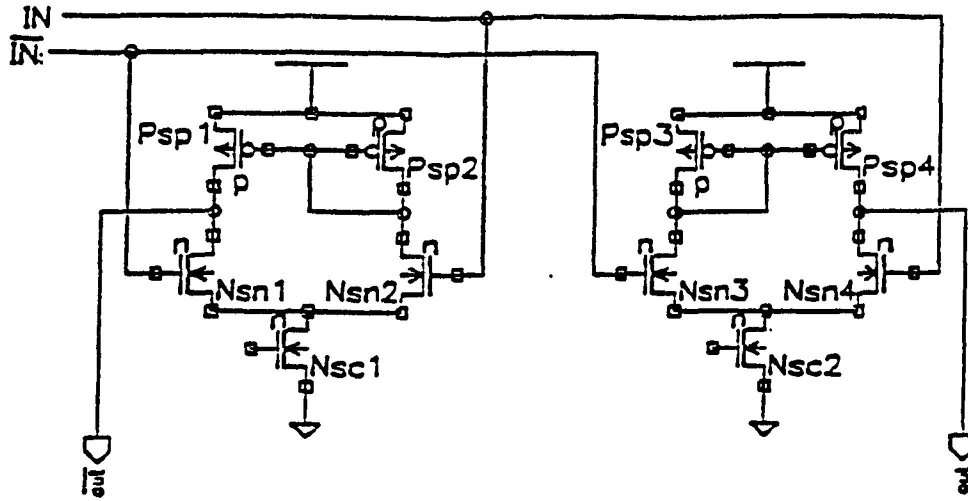


Fig. 1.17(a) Conventional CMSA [1]

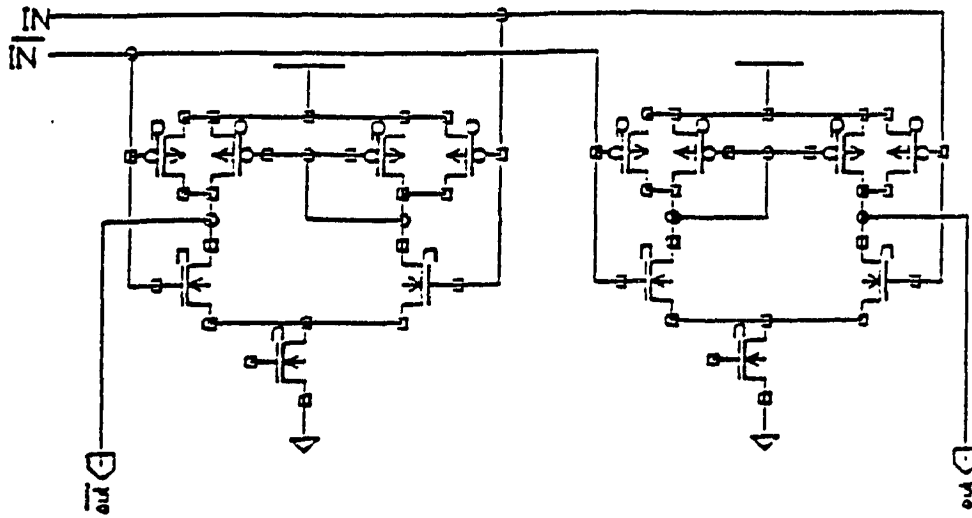


Fig. 1.17(b) Input-controlled PMOS load (ICPL) sense amplifier [1].

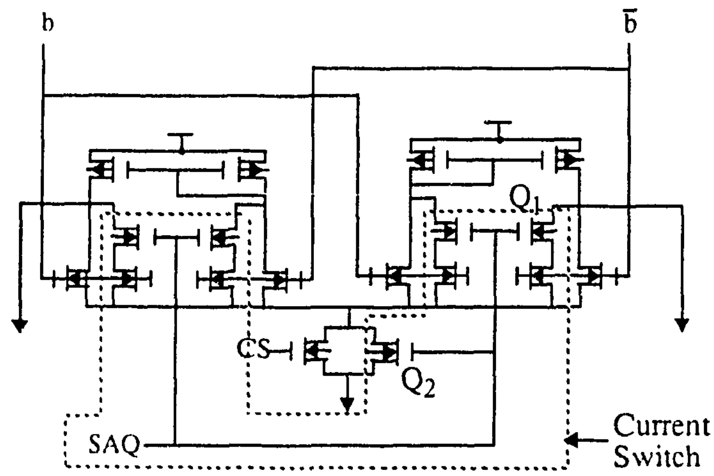


Fig. 1.18 Dynamic Gain Control Double end Amplifier [5]

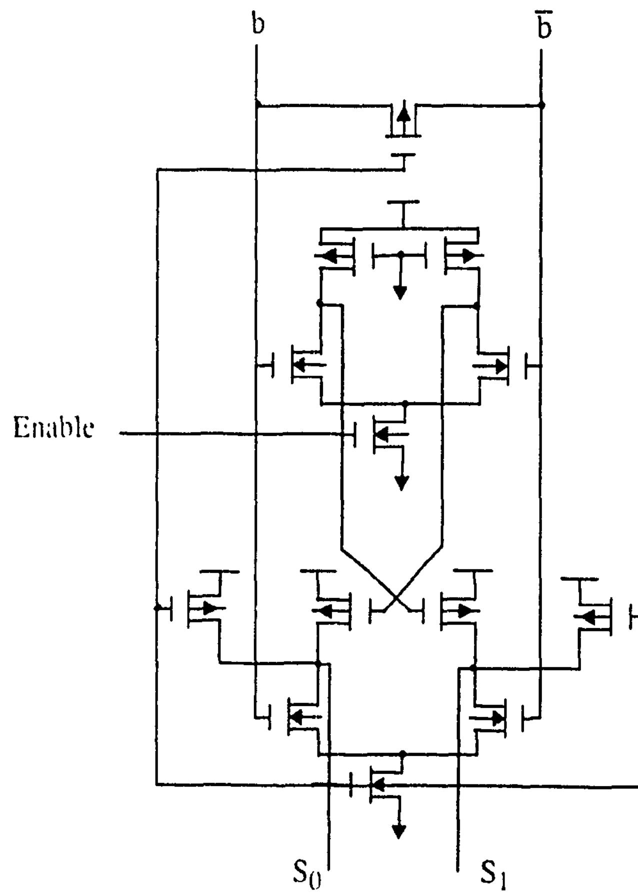


Fig. 1.19 Modified two stage CMSA [29]

1.8.3 Hierarchical Sense Amplifier Architecture

In a high density SRAM as the number of columns and blocks or divisions increase the sense amplifier input data bus line capacitance increases tremendously. So, it is necessary to partition the data-line in different hierarchical structures which will distribute the total capacitance in different hierarchical levels so optimum sensing delay can be achieved. The double line SA structure as reported in [10] is shown in Fig.1.20. This structure implements a local sense amplifier in each column section and a main sense amplifier as a master or global sense amplifier. In this configuration all the bit lines in a particular Section are connected to the local bus line which is input to the Section sense amplifier. Each Section sense amplifier output is connected to the global (main) sense amplifier input bus. Final sense output is carried out from the main SA. Sense clocks ϕ_{SA} and ϕ_{MA} handle the sense operation. This sense amplifier architecture can be suited for a medium density SRAM. In a high density SRAM as the number of column per block increases or the total number of blocks increases, for example in a 1Mbit SRAM, there may be a total of 32 blocks with 64 columns per block. In this case, the double sense line structure is not adequate because of the increased local as well as global data-line capacitance. To overcome this problem a two - stage local sense amplification with a triple - sense - line structure [6] and a four stage SA (four sense lines) [4] is proposed which is depicted in Fig.1.21(a) and 1.22 respectively. In these schemes a Section sense amplifier and block sense amplifier is selected by a hierarchical column decoding system. It can be realized that the circuit in Fig. 1.22 might need double the layout area of that in Fig.1.21(a).

For our 'Sensing Scheme' design we would prefer an optimized design of a CMOS CMSA. As far as high-density high-speed (HDHS) SRAMs are concerned we use a multi-stage SA structure with a hierarchical sense enabled architecture.

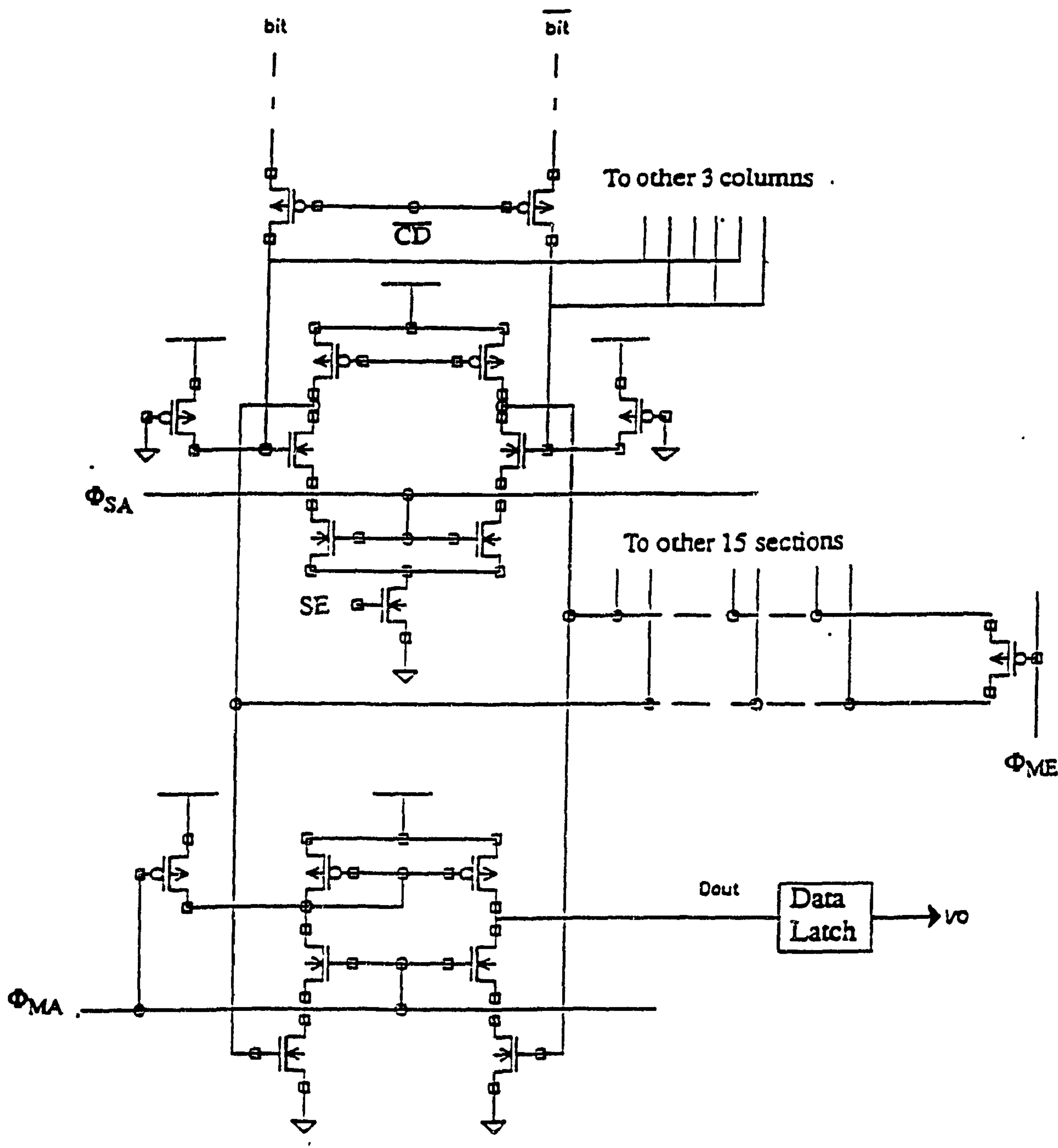


Fig. 1.20 Double line sensing structure [10].

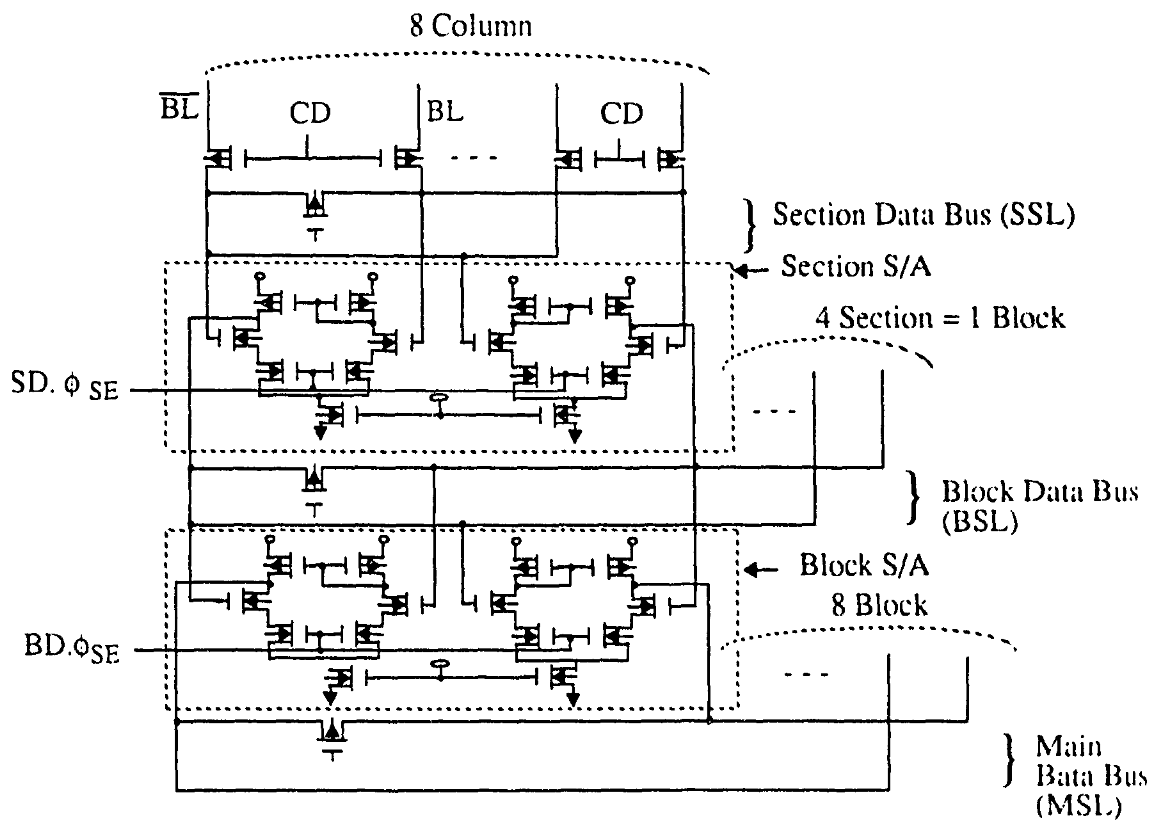


Fig. 1.21(a) Sense Amplifier circuit for Fig. 1.21(b) [6]

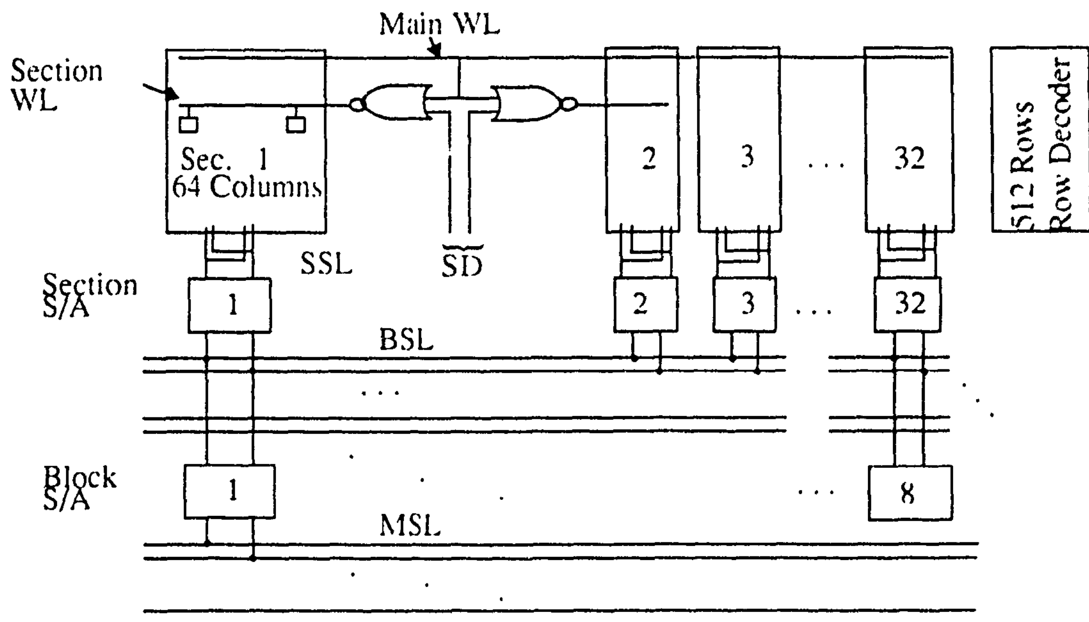


Fig. 1.21(b) Two-stage local amplification and triple-sense-line structure [6].

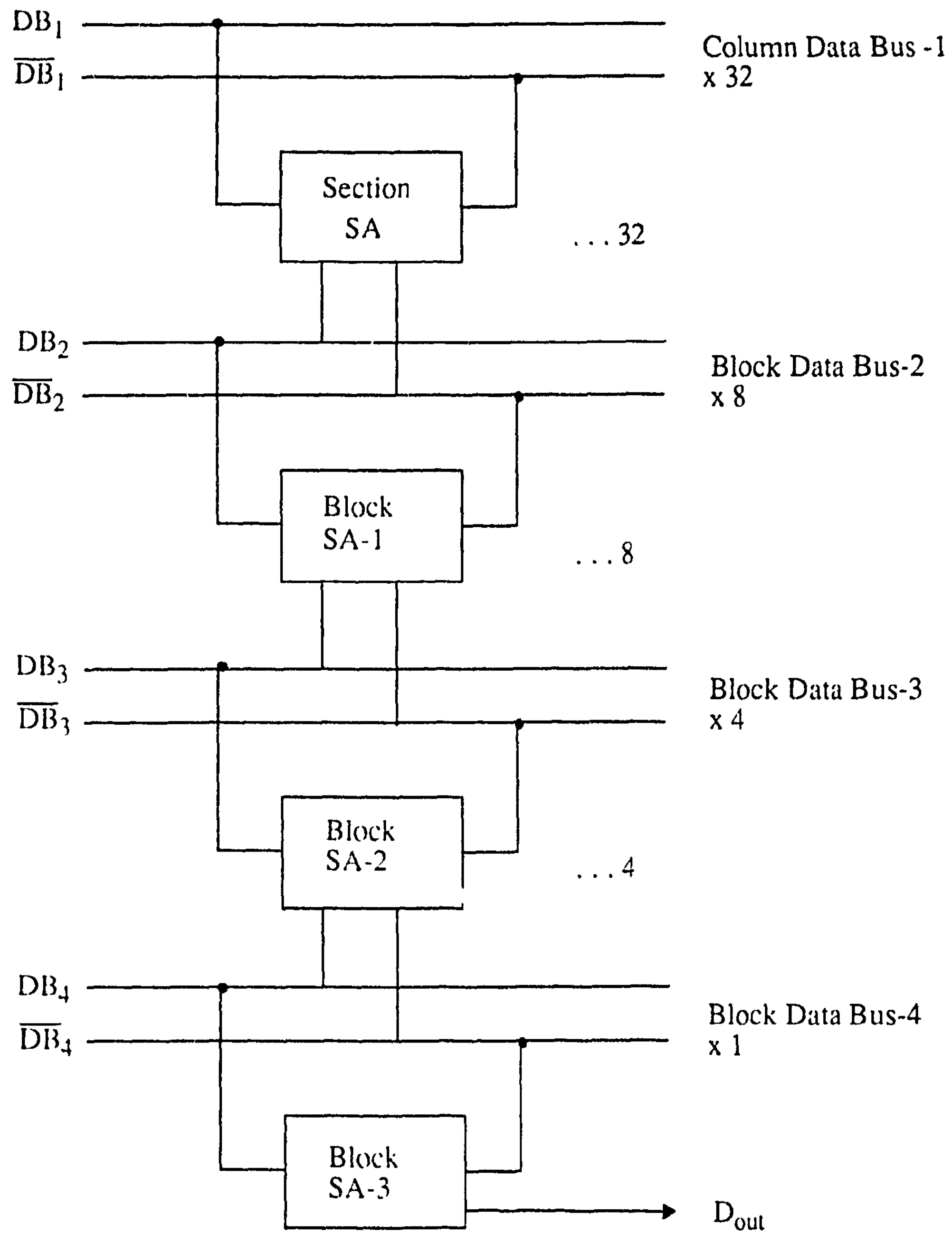


Fig. 1.22 Four-stage Sensing Scheme [4].

Chapter 2

Modeling and Analysis of SRAM

2.0 Introduction

In this Chapter, we first developed the approximate generalized capacitance model of a SRAM specially for bit line and word line. The SPICE device parameters are used for analysis. Furthermore, the capacitance model is used to develop the RC delay model for a SRAM. In our delay model we assume that the RC time constant is linearly related to the actual delay. Therefore, we will use time constant as an absolute measure of delay and hereafter regression fit it with the actual delay obtained by SPICE simulation. The regression fit consists of two steps. First, the initial fit constants are obtained from regression analysis of the mathematical modeling results and SPICE simulation. Then the second or final regression fit is done between the initial fit constants and m (number of rows), n (number of columns) in order to generalize the model.

2.1 Capacitance Model of Word and Bit Line

A very good capacitance model is imperative to have a better delay model. The gate capacitance of the CMOS transistor is assumed to be linearly dependent on the transistor width. Furthermore, in our present analysis we assume the MOS device capacitance as a lumped element and the interconnect capacitance as an uniformly distributed parameter. It can be noted that in our analysis the subscript 'n' & 'p' will be used for NMOS and PMOS transistors respectively.

2.1.1 Word Line Capacitance Model

The word line capacitance consists of the cell access transistor gate capacitance plus the distributed capacitance of the word line interconnect metal/poly which is depicted in

Fig. 2.1.1. The SRAM cell access transistor gate capacitance C_g can be determined as,

$$C_g = C_{gba} + C_{gsa} + C_{gda} \tag{2.1}$$

$$= C_{oxn}\, W_a\, L_a + C_{oxn}\, W_a\, LD_{sa} + C_{oxn}\, W_a\, LD_{da} \tag{2.2}$$

Where C_{gba}, C_{gsa} & C_{gda} are gate to bulk, gate to source, gate to drain capacitance respectively of the cell access transistor.

C_{oxn} is the NMOS transistor gate oxide capacitance per unit area.

L_a, W_a are the Length and width of the NMOS cell access transistor.

LD_{sa}, LD_{da} are the source and drain lateral diffusion length respectively of the cell access transistor.

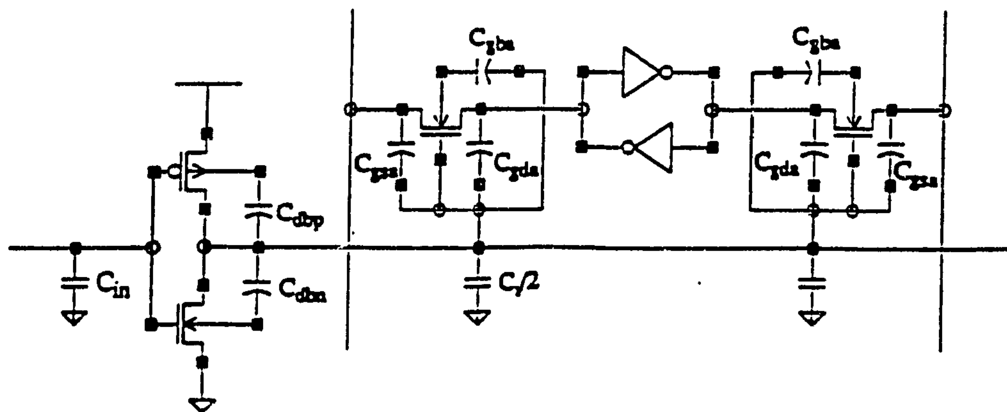


Fig. 2.1.1 Word line Capacitance Modeling

Assuming the word line is connected to n cells, there might be n word segments in a row. The word line interconnect capacitance C_i per segment can be determined as,

$$C_i = c_o\, w_w\, l_w \tag{2.3}$$

where c_o is the Word line inter-connect capacitance per unit area.

w_w & l_w are the width and length respectively of each word segment.

Since there are two access transistors per cell in SRAM therefore, the capacitance per segment of word line can be written as,

$$C_w = 2\, C_g + C_i. \tag{2.4}$$

2.1.2 Word line driver I/O capacitance

The input capacitance of a driver, C_{din} consist of input gate capacitance C_{gi} of driver PMOS & NMOS transistors and the output capacitance $C_{d(i-1)}$ of the driver driving it which is depicted in Fig. 2.1.2.

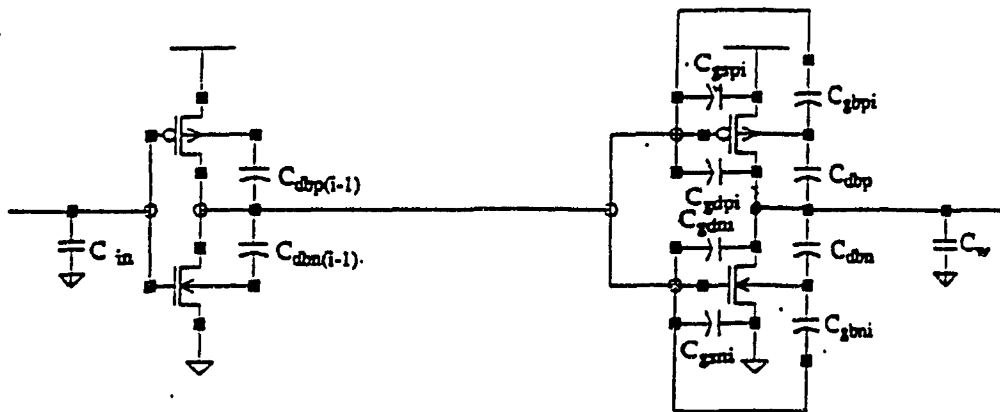


Fig. 2.1.2 Word driver I/O Capacitance Modeling

Now, C_{din} can be written as,

$$C_{din} = C_{d(i-1)} + C_{gi} = (C_{dbn} + C_{dbp})_{i-1} + (C_{gn} + C_{gp})_i$$

$$= (C_{dbn} + C_{dbp})_{i-1} + (C_{gbn} + C_{gsn} + C_{gdn} + C_{gbp} + C_{gsp} + C_{gdp})_i$$

$$= K_{eq}(CJ_n Ad_n + CJSW_n Pd_n + CJ_p Ad_p + CJSW_p Pd_p)_{i-1}$$

$$+ (C_{oxn} W_n L_n + C_{oxn} W_n LD_{sn} + C_{oxn} W_n LD_{dn}$$

$$+ C_{oxp} W_p L_p + C_{oxp} W_p LD_{sp} + C_{oxp} W_p LD_{dp})_i \quad (2.5)$$

Where C_{dbn} and C_{dbp} are the drain-bulk capacitance of both P & N channel transistors respectively. C_{gbn} and C_{gbp}, C_{gsn} and C_{gsp}, C_{gdn} and C_{gdp} are the gate-bulk, gate-source, gate-drain capacitance of both channel transistors. K_{eq} [30] is the dimensionless constant applied to all equivalent junction capacitances. CJ_n and CJ_p are the zero-bias junction bottom capacitances per unit area of junction of both channel transistors. $CJSW_n$ and $CJSW_p$ are the zero-bias bulk junction sidewall capacitances per unit length of junction of both channel transistors. Ad_n, Pd_n & Ad_p, Pd_p are the area and perimeter of the drain diffusion region of both channel transistors. W_n, L_n and W_p, L_p are the width and length of both channel transistors. LD_{sn} and LD_{sp} are the source lateral diffusion length of both channel transistors. LD_{dn} and LD_{dp} are the drain lateral diffusion length of both channel transistors.

Assuming drain diffusion area as rectangular in shape, the area of the drain diffusion region can be given by,

$$Ad_n = W_{dn} L_{dn}$$

$$Ad_p = W_{dp} L_{dp} \quad (2.6)$$

The perimeter of the drain diffusion region can be given by,

$$Pd_n = 2(W_{dn} + L_{dn})$$

$$Pd_p = 2(W_{dp} + L_{dp}) \quad (2.7)$$

Where W_{dn} and L_{dn} are the drain width and length of the NMOS transistor.

W_{dp} and L_{dp} are the drain width and length of the PMOS transistor.

The output capacitance of the driver consists of the driver self output capacitance C_{do} and the word line capacitance which is sum of all the cell access transistor's gate capacitances and the interconnect capacitance. The driver output capacitance C_{do}can be represented by the drain to bulk capacitances C_{dbn} and C_{dbp} of the NMOS and PMOS transistors and the π–model word line interconnect capacitance $C_i/2$ respectively. So, we can write,

$$C_{do} = C_{dbn} + C_{dbp} + C_i/2$$

$$=K_{eq}\,(CJ_n\,Ad_n + CJSW_n\,Pd_n + CJ_p\,Ad_p + CJSW_p\,Pd_p) + C_i/2 \qquad (2.8)$$

Where K_{eq}, CJ_n, CJ_p, $CJSW_n$, $CJSW_p$, Ad_n, Ad_p, Pd_n, Pd_p, and C_i are as defined previously.

2.1.3 Bit line Capacitance

The bit line capacitance has a significant effect on the access time. An integral increase of bit line delay happens as the number of cells connected per column increases. The major capacitance contribution in the bit line can be realized as the sum of all cell access transistor's capacitances and bit line self capacitance. Assume there are m cells connected per bit line. We consider there are m bit segments in our R, C model. The capacitance per segment C_{bs} may be represented by the sum of the source to bulk capacitance C_{sba} of the cell access transistor and the bit line interconnect capacitance C_{ib} as,

$$C_{bs} = C_{sba} + C_{ib}$$

$$=K_{eq}\,(CJ_n\,As_a + CJSW_n\,Ps_a) + c_b\,A_b \qquad (2.9)$$

Where K_{eq}, CJ_n and $CJSW_n$ are as defined in Section-2.1.2.

As_a and Ps_a are the source area and perimeter of the cell access transistor.

c_b and A_b are the bit line interconnect capacitance per unit area and area of each bit line segment respectively.

2.2 Word line Delay Analysis

As the memory density increases in the SRAM structure, the delay due to the RC time constant of the word line increases considerably and has a greater influence on the total address access time. The word line can be modeled as a general T or π-ladder network and the distributed parameters can be approximated to the total resistance and capacitance of each cell involved on the word line. The design of the word line driver is one of the important issues in having a faster access speed.

2.2.1 Word line delay modeling using Elmore's Delay Model

Fig. 2.2.1(a) & 2.2.1(b) show the circuit diagram and physical representation of a word line in a SRAM. Let the word line consist of n (number of columns/block) segments. Each segment is connected with two access transistor's gate contacts. So, there are $2n$ number of gate contacts plus one driver's output contact which is shown in Fig. 2.2.1(b). Let us define the following parameters:

$R_{s\,i,j}$ is the resistance of segment i, j;

$C_{s\,i,j}$ is the capacitance of segment i, j;

$R_{a\,l,i,j}$ is the resistance of the access transistor gate contact of l,i,j;

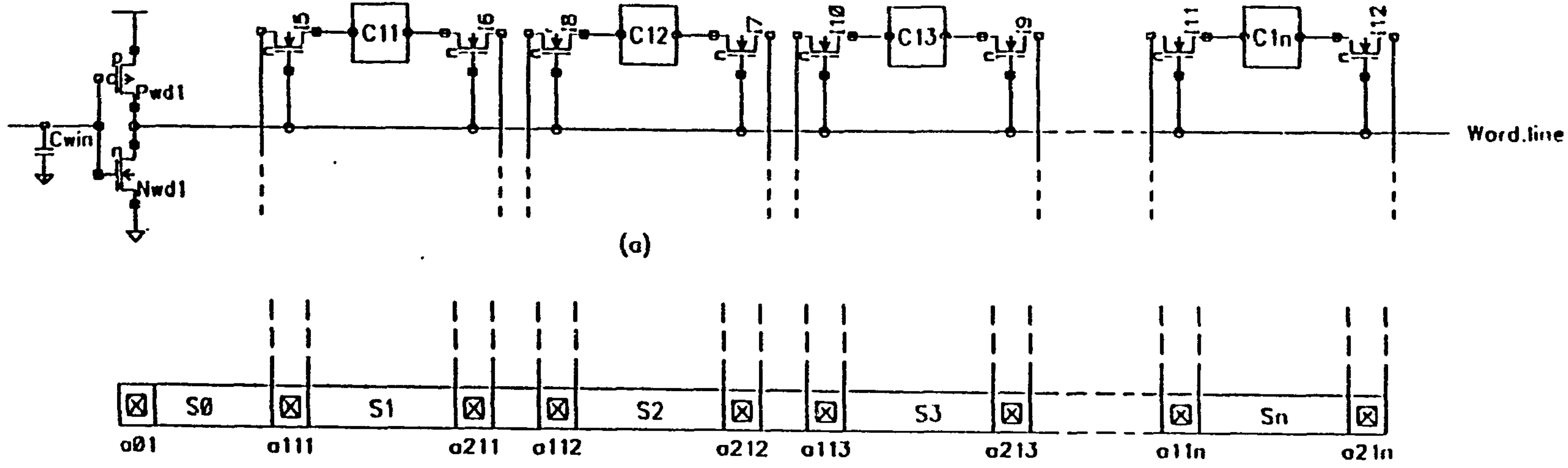
$C_{a\,l,i,j}$ is the gate capacitance of access transistor l,i,j;

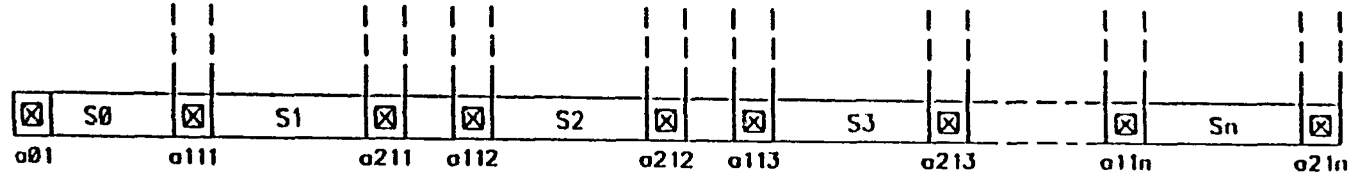
R_{ds} is the drain to source resistance of the driver PMOS transistor;

C_{ds} is the output capacitance of the driver;

l = 1 or 2 (cell access transistor); i = row, j = column.

(* for the simplicity of analysis the subscript 'i' will be omitted for any row).





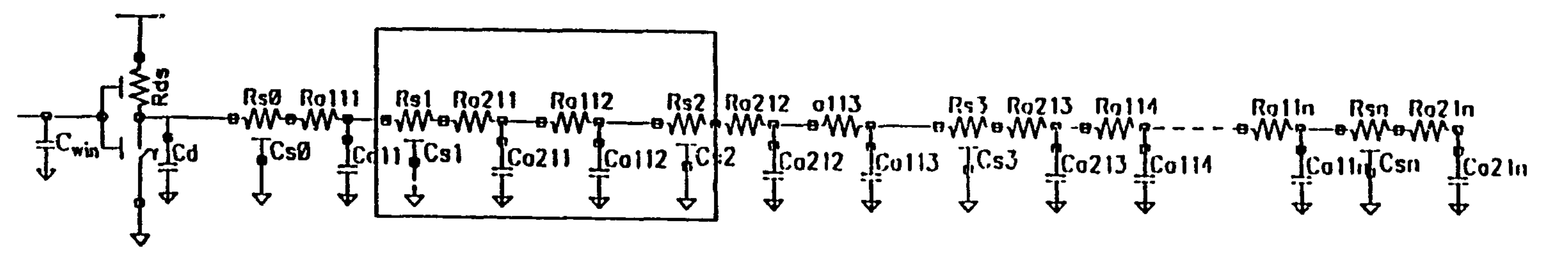


Fig. 2.2.1 (a) Word driver driving a row of a SRAM
(b) Physical representation of a word line
(c) R, C delay model of a Word line.

The word line can be modeled by a distributed R, C tree. The propagation delay, T_w of the word line can be represented by an Elmore's time constant [18]. The delay is to be calculated for the whole word line loading in which the resistance and capacitance of the gate is represented by the lumped element and the interconnect wire is represented by uniformly distributed R, C lines.

Elmore's time constant can be defined as follows:

$$T_{d_i} = \sum_k R_{ki} C_k \tag{2.10}$$

Where R_{ki} is the resistance between node k & i, C_k is the capacitance at node k.

We assume that the resistance and capacitance of the word line is uniformly distributed along the length L_l and the cumulative capacitance $c(x)$ is a function of the position along the wire. For each value of $c(x)$, there is a cumulative resistance $r(c)$ linearly increasing with $c(x)$, so that $c(0)=0$, $r(0)=0$ and $c(L_l)=C_l$, $r(C_l) = R_l$. For uniform lines $r(c) = (R_l/C_l)c$. Elmore's time constant for a single R, C wire is,

$$T_{d_i} = \int r(c)\,dc = \frac{1}{2} R_l C_l \tag{2.11}$$

The R C models of the word line of Fig. 2.2.1(a) is shown in Fig.2.2.1(c). Elmore's time constant for word line delay can be represented as follows (the subscript 'i' is left for any row):

$$T_w = R_{ds}C_{ds} + (R_{ds} + \frac{1}{2}R_{s0})C_{s0} + (R_{ds} + R_{s0} + R_{a11})C_{a11} +$$

$$(R_{ds} + R_{s0} + R_{a11} + \frac{1}{2}R_{s1})C_{s1} + (R_{ds} + R_{s0} + R_{a11} + R_{s1} + R_{a21})C_{a21} +$$

$$\ldots + (R_{ds} + R_{s0} + R_{a11} + \ldots + R_{sn-1} + R_{a1n} + R_{sn} + R_{a2n})C_{a2n}$$

$$= R_{ds}C_{ds} + \sum_{l=1,2,} \sum_{j=1,n} R_{ds}(C_{sj} + C_{alj}) + \sum_{j=0,n} R_{sj}(\frac{1}{2}C_{sj} + \sum_{l=1,2} C_{alj})$$

$$+ \sum_{l=1,2,l} \sum_{1,n} R_{alj}(C_{alj} + C_{sj}) \tag{2.12}$$

Where R_{ds} is the channel resistance of the word driver PMOS transistor.

C_{ds} is the drain output resistance of the word driver.

$R_{s0}, R_{s1}, \ldots, R_{sl}$ are the resistances of the word line segments.

$R_{a11}, R_{a21}, R_{a12}, R_{a22}, \ldots, R_{a1n}, R_{a2n}$ are resistances of the word line contacts.

$C_{so}, C_{s1}, \ldots, C_{sn}$ are the capacitances of the word line interconnect segments.

$C_{a11}, C_{a21}, C_{a12}, C_{a22}, \ldots, C_{a1n}, C_{a2n}$ are the gate capacitances of the cell access transistors.

j is defined as 1 to n and n is the number of cells connected to the word line.

l determines the number of cell access transistors in each segment which is equal to 2.

2.2.2 Approximate form of Word line

The approximate form of the word line is shown in Fig. 2.2.2(b). Here, the word line is modeled as π-ladder circuit with lumped R, C parameters. Considering that there are n cells per row, the signal delay according to Elmore's delay for this type of interconnection line can be approximated as follows:

$$T_{wd} = T_{di} + T_{do} \tag{2.13}$$

where T_{di} and T_{do} are the signal delay at the driver input and output respectively.

Assume the driver is driven by an another driver at the input. So, the signal delay at the input of the word driver can be determined as,

$$T_{di} = R_{nin}\, C_{din}. \tag{2.14}$$

Where R_{nin} and C_{din} are the input resistance and capacitance of the driver.

It is observed that due to the effect of I/O capacitance and intrinsic delay of the logic block the step input delay model suffers from accuracy. A better approximation of a delay

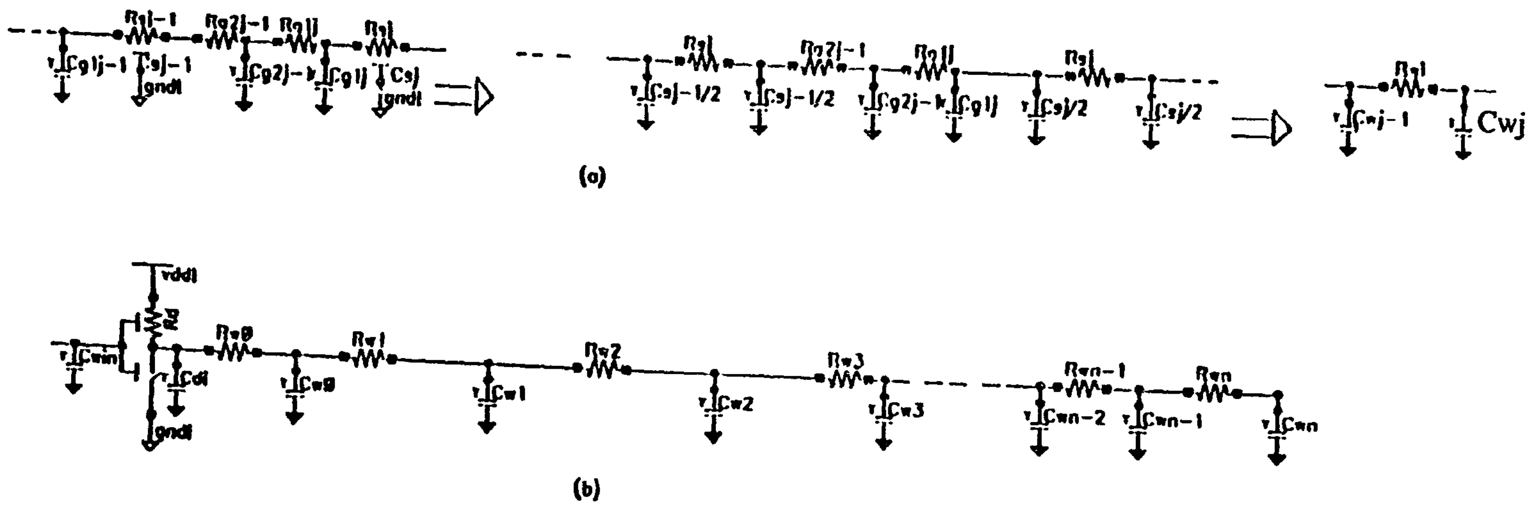


Fig. 2.2.2 (a) π- approximate model of a word line in a cell segment

(b) Approximate π- R, C model of a word line.

model can be realized using a ramp input model as in [17]. According to this model the output delay T_{do} of the word driver consists of step response delay T^s_{do} and the input ramp dependent delay T^r_{do}, as

$$T_{do} = T^s_{do} + T^r_{do}. \tag{2.15}$$

Assuming equal word segment resistance R_w, the step response delay T^s_{do} can be determined using equation (2.12) as,

$$T^s_{do} = R_{pd}\, C_{do} + (R_{pd} + R_w)\, C_{w0} + (R_{pd} + 2R_w)\, C_{w1} + \ldots$$

$$+ (R_{pd} + nR_w)\, C_{wn-1} + [R_{pd} + (n+1)\, R_w]\, C_{wn} \tag{2.16}$$

Where the contact resistances R_{alj} in equation (2.12) is neglected which is assumed to be very small.

R_{pd} is the channel resistance of word driver PMOS transistor which can be determined by,

$$R_{pd} = \frac{K_r}{\beta_p (V_{gs} - V_{tp})} \tag{2.17}$$

Where K_r is the resistivity factor which depends upon the region of operation of the transistor.

C_{do} is the driver drain output self capacitance which is given in equation (2.8).

R_w resistance of each word line segment which can be determined by,

$$R_w = r_o\, N_{sw} \tag{2.18}$$

Where r_o is the resistance per square of the word line metal/poly and N_{sw} is the number of squares of the word line for each cell segment which can be determined as,

$$N_{sw} = L_w / W_w \tag{2.19}$$

Where L_w, W_w are length and width of each word segment under a cell.

$C_{w1}, C_{w2}, \ldots, C_{wn}$ are the capacitances of word segments.

The capacitance of each segment of the word line except the first and last node

capacitance is equal, i.e.,

$$C_w=C_{w1}=C_{w2}= \dots = C_{wnn-1}=2\ C_g+C_i$$

$$C_{w0}=C_{wn}= C_g+C_i/2= C_w/2.. \tag{2.20}$$

Where C_g, C_i and C_W are determined in Section-2.1.1.

Now, putting equation (2.20) into equation (2.16) we have,

$$T^S_{do} = R_{pd}C_{do} + nR_{pd}C_w + (1+2+3+\dots+n)R_wC_w + \frac{n}{2}R_wC_w \tag{2.21}$$

Which follows to,

$$T^S_{do} = R_{pd}C_{do} + nR_{pd}C_w + \frac{n(n+2)}{2}R_wC_w \tag{2.22}$$

where n is the number of columns or number of cells connected locally per word line.

The ramp input dependent delay $T^r{}_{do}$ derived in [17] can be extended as,

$$T^r_{do} = (\frac{1}{6})\left(1+2\frac{V_t}{V_{dd}}\right)\tau \tag{2.23}$$

Where τ is the input rise/fall time.

V_t is the transistor threshold voltage.

V_{dd} is the power supply voltage.

Now, equating equation (2.14), (2.22) & (2.23) into equation (2.13) we have,

$$T_{wd} = R_{nun}C_{dun} + R_{pd}C_{do} + nR_{pd}C_w + \frac{n(n+2)}{2}R_wC_w + (\frac{1}{6})\left(1+2\frac{V_t}{V_{dd}}\right)\tau \tag{2.24}$$

We will use equation (2.24) as an absolute measure of word delay. The results obtained from equation (2.24) are fitted using regression analysis with actual SPICE simulation results in order to derive an appropriate and generalized simple model which is given by equation (5.15) in Chapter 5. Fig. 2.2.2(c) depicts a comparison of our analytical (fitted) and SPICE simulation results for different word line loading as a function of word

driver size. The model can be used to predict an approximate size of the driver according to design requirements or goals. It is verified that the average error rate between our analytical result and the SPICE simulation lies within 10%.

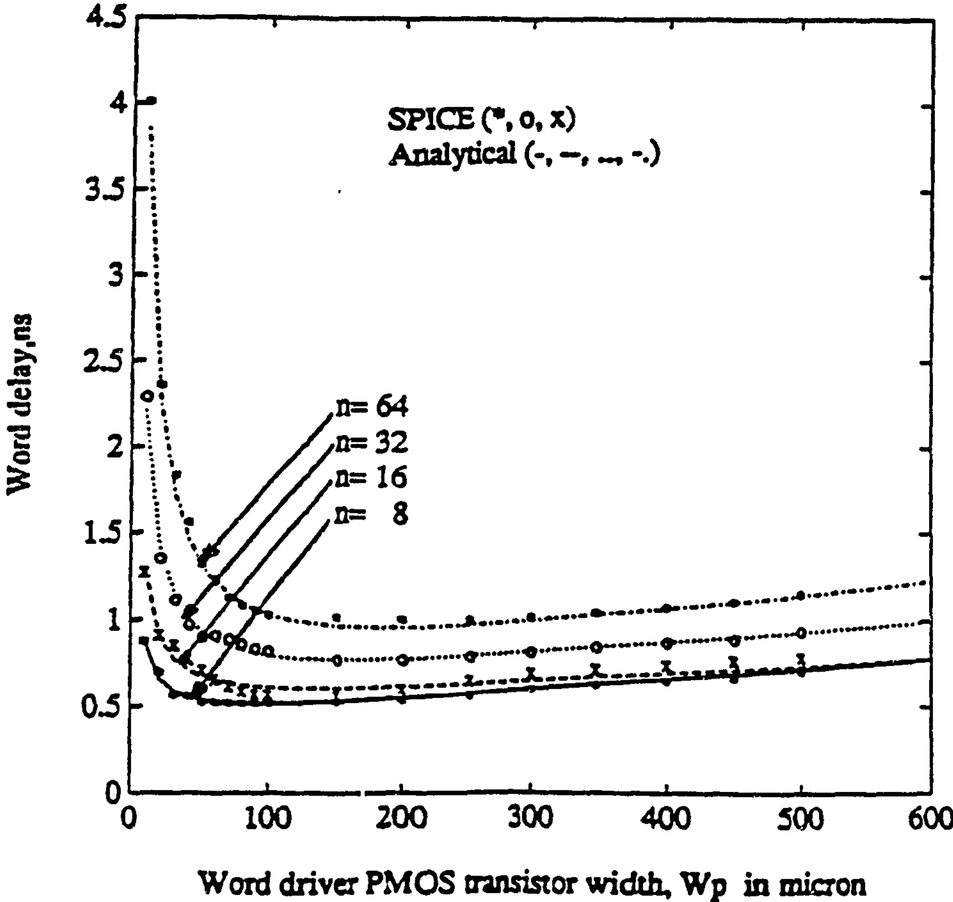


Fig. 2.2.2(c) Comparison of analytical and SPICE simulation results for different word sizes

2.3 Precharge Delay Analysis

Our Power Down Y-controlled PMOS(PDYCP) bit line precharge circuit is depicted in Fig. 1.14 (inside box). The precharge circuit is modeled according to Elmore's delay model as shown in Fig. 2.3(a).The total precharge delay in a bit line can be determined as,

$$T_{prech} = T_{cd} + T_{pd} + T_{pr} + T_{bit} + T_{rampi} \qquad (2.25)$$

Where T_{prech} is the total precharge delay.

T_{cd} is the column driver delay

T_{pd} is the delay due to power down circuit to push node x to V_{SS}.

T_{pr} is the delay due to precharge transistor P_{pr1} or P_{pr2}.

T_{bit} is the delay due to bit line loading.

T_{rampi} is the delay due to ramp input.

The delay due to the column driver T_{cd} can be determined as,

$$T_{cd} = R_{cd} C_{11} \qquad (2.26)$$

Where R_{cd} is the column driver P/N MOS transistor resistance.

C_{11} is the column driver and load output capacitance.

In our design the column driver plays a very important role. It is used for dual purpose: column and precharge selection. According to our implementation when the column is selected to READ or WRITE the precharge is deselected automatically. The output capacitance of the column driver C_{11} consists of driver self capacitance and the precharge gate capacitance which can be determined as,

$$C_{11} = C_{dbcdn} + C_{dbcdp} + 2\, C_{gcs} + C_{gtn1} \qquad (2.27)$$

Where C_{dbcdn} & C_{dbcdp} are the drain to bulk capacitance of the column driver N & P transistors respectively.

C_{gcs} is the column select transistor gate capacitance.

Fig. 2.3(a) Precharge delay modeling.

C_{gtn1} is the precharge select transistor T_{n1} gate capacitance.

The power down precharge control circuit delay can be determined as,

$$T_{pd} = R_n C_{10} + \frac{n(n+1)}{2} R_{wp} C_{wp} + (R_{tn} + R_{tn1}) C_x \qquad (2.28)$$

Where R_{tn} is the channel resistance of the bottom most NMOS transistor T_n which is controlled by the address transition detection (ATD) signal.

R_{tn1} is the channel resistance of the transistor T_{n1} as in Fig. 1.14.

C_{10} is the capacitance at node *10* in Fig. 1.14 where all middle NMOS transistors are interconnected.

n is the number of columns which is equal to the number of precharge circuits.

R_{wp} is the resistance per segment of precharge interconnect.

C_{wp} is the capacitance per segment of precharge interconnect.

C_x is the capacitance at node x (called precharge node) in Fig. 1.14.

C_{10} and C_x can be determined as,

$$C_{10} = n\,C_{sbtn1} + C_{dbtn} + n\,C_i$$
$$C_x = 2\,C_{gpr} + C_{dbtn1} + C_{dbtp1} \qquad (2.29)$$

Where n is the number of columns which is equal to the number of precharge circuits.

C_{sbtn1} is the source-bulk capacitance of transistor T_{n1} as in Fig. 1.14, assume T_{n1}, T_{n2}, ..., T_{nj} are equivalent transistors.

C_{dbtn} is the drain-bulk resistance of transistor T_n as in Fig. 1.14.

C_i is the power down circuit interconnect capacitance which is assumed to be equivalent to the word line interconnect capacitance as given by equation (2.3).

C_{dbtn1} is the drain-bulk capacitance of transistor T_{n1} as in Fig. 1.14.

C_{dbtp1} is the drain-bulk capacitance of transistor T_{p1} as in Fig. 1.14.

C_{gpr} is the gate capacitance of a precharge PMOS transistor (T_{p11} or T_{p21}) as in Fig. 1.14.

The delay due to each precharge transistor T_{pr} can be determined as,

$$T_{pr} = R_{pr} (C_{03} + m\, C_{bs}) \tag{2.30}$$

Where R_{pr} is the resistance of the precharge transistor.

m is the number of rows or cells connected per bit line.

C_{bs} is the bit line capacitance per segment which is given by equation (2.9).

C_{03} is the capacitance at node 03 (Fig. 2.3(a)) of the bit line where the precharge transistor drain is connected. C_{03} can be determined as,

$$\begin{aligned} C_{03} &= C_{dbpr} + C_{dbcsp} + C_{dbcsn} \\ &= K_{eq} [CJ_p\, W_{pr}\, L_d + 2\, CJSW_p (W_{pr} + L_d) + \\ &\quad CJ_p\, W_{csp}\, L_d + 2\, CJSW_p (W_{csp} + L_d) + \\ &\quad CJ_n\, W_{csn}\, L_d + 2 CJSW_n (W_{csn} + L_d)]. \end{aligned} \tag{2.31}$$

Where C_{dbpr} is the drain-bulk capacitance of the bit line precharge transistor T_{p11} as in Fig. 1.14.

C_{dbcsp} is the drain-bulk capacitance of the column select P transistor.

C_{dbcsn} is the drain-bulk capacitance of the column select N transistor.

K_{eq}, CJ_n, CJ_p, $CJSW_n$, $CJSW_p$ are as defined in Section 2.1.2.

The Elmore delay model of the bit line is depicted in Fig. 2.3(a). The capacitance contribution in the bit line is mainly due to bit line distributed capacitance and the access transistor source-bulk or drain-bulk capacitance.

Now, T_{bit} can be determined as,

$$T_{bit} = \frac{m(m+1)}{2} R_b C_{bs} \tag{2.32}$$

Where m is the number of cells connected to the bit line.

R_b is the resistance per square of bit line segment for each cell which can be determined as,

$$R_b = r_b N_{sb}. \tag{2.33}$$

where r_b is the resistance per square of the bit line.

N_{sb} is the number of squares in a bit segment (cell height).

C_{bs} is the capacitance per bit segment which is given by equation (2.9).

The ramp input dependent delay T_{rampi} is defined as in equation (2.23).

$C_{db\sim}$, $C_{sb\sim}$, $C_{g\sim}$, etc. are defined in appendix-A.

Our mathematical modeling results, which is fitted with SPICE simulation results for precharge delay, is depicted in Fig. 2.3(b). Also, a comparison between the analytical result & SPICE simulation is obtained which is shown in Fig. 2.3(c). It can be inferred that our simple Elmore delay model induces a very good approximation with less than 10% error.

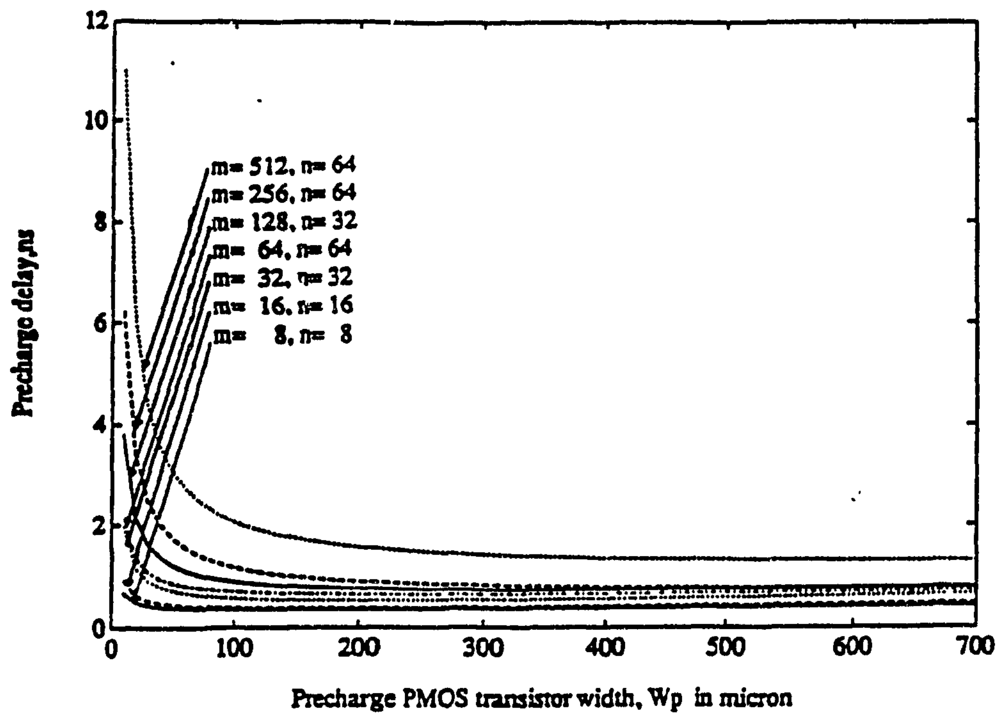


Fig. 2.3(c) Precharge Delay Analytical result.

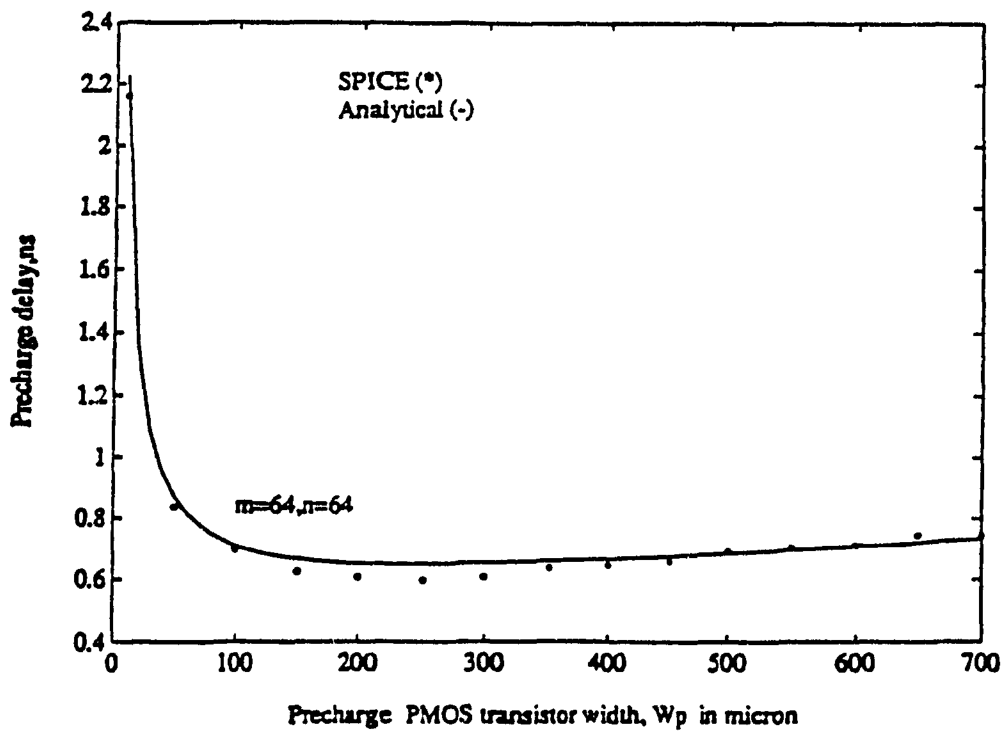


Fig. 2.3(d) Comparison of analytical and SPICE simulation results for precharge delay.

2.4 Sensing Delay

Assume a read-1 situation in which the '$\overline{\text{bit}}$' line in Fig.1.16(a) will be pushed down towards V_{ss} and thereby turn Sense amplifier N transistor N_{sn2} OFF. So, the sensing speed will depend upon the rate of change of the precharge discharge voltage, the switching characteristics of sense current transistor, N_{snc},and the gain of the sense amplifier. In the resulting situation transistors P_{sp1}, P_{sp2}, N_{sn1}, N_{snc} will be ON and N_{sn2} will be OFF. The Elmore delay model for read operation is depicted in Fig. 2.4(a) and 2.4(b) to read an '1' and '0' respectively.

The total sensing delay can be determined as,

$$T_{sense} = T_{dc} + T_{prech} + T_{bus} + T_{bit} + T_{acell} + T_{sa} + T_{ob} \qquad (2.34)$$

where T_{dc} is the decoding system delay.

T_{prech} is the Precharge delay.

T_{bus} is the data bus delay.

T_{bit} is the bit line delay.

T_{acell} is the cell access delay.

T_{sa} is the sense amplifier delay.

T_{ob} is the output buffer delay.

The decoding system delay consists of predecoder, decoder and word driver delay. Since the word driver is the one which drives the whole load in the word line then in our analysis, we will give special emphasis to the word driver delay and consider predecoder and decoder as a fixed time slice.

The word driver delay T_{wd} and the precharge delay T_{prech} are referred to Section-2.2.2. and 2.3 respectively. The data bus delay T_{bus} mainly depends upon the bus capacitances. The bus capacitance C_{bus} consists of column select T-gate drain-bulk

Fig. 2.4(a) Read-1 delay modeling

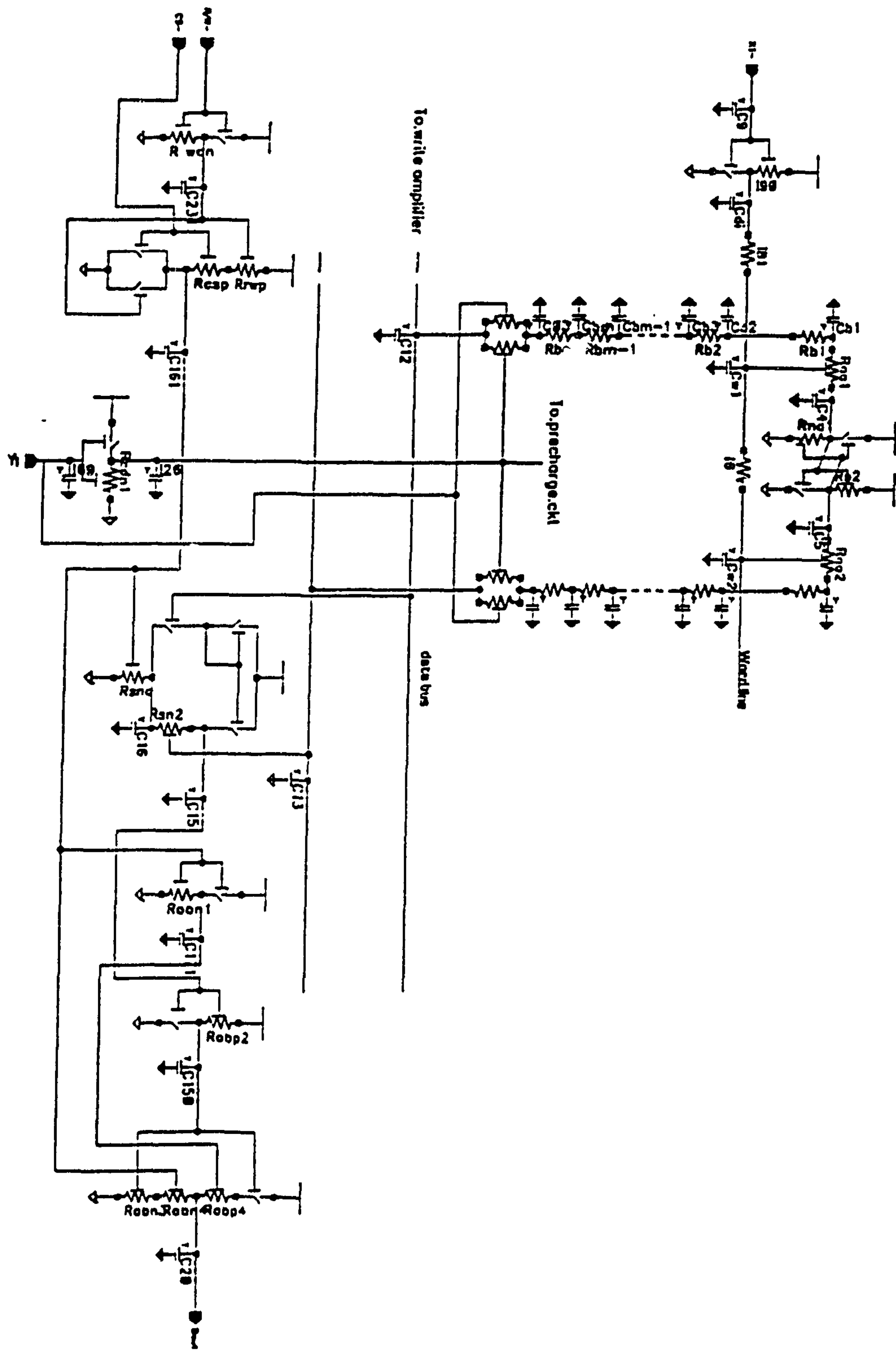


Fig. 2.4(b) Read-0 delay modeling

capacitance, sense amplifier gate capacitance and data bus interconnect capacitance.

The data bus delay can be determined as,

$$T_{bus} = \frac{n(n+1)}{2} R_{bus} C_{bus} + n R_{bus} (2C_{dbwpp} + C_{gns}) \qquad (2.35)$$

Where n is the number of columns connected to the word line.

R_b is the data bus interconnect resistance per segment, assuming there are n segments.

C_{bus} is the data bus capacitance per segment.

C_{dbwpp} is the drain-bulk capacitance of the write T-gate transistor connected to the data bus.

C_{gns} is the gate capacitance(s) of the sense amplifier(s) transistor(s) connected to the data bus.

The data bus interconnect resistance R_{bus} per bus segment can be determined as,

$$R_{bus} = r_{bus} N_{bus} \qquad (2.36)$$

where r_{bus} is the resistance per square of data bus interconnect.

N_{bus} is the number of squares per bus segment, assume there are n (number of columns) bus segments.

The data bus capacitance C_{bus} can be determined as,

$$C_{bus} = C_{dbcsn} + C_{dbcsp} + C_{ibus} \qquad (2.37)$$

Where C_{dbcsn}, C_{dbcsp} are column select N & P transistor drain-bulk capacitance respectively and C_{ibus} is the data bus interconnect capacitance per segment.

The bit line delay T_{bit} for sensing case can be determined as,

$$T_{bit} = R_{eqcs} (C_{o3} + mC_b) + \frac{m(m+1)}{2} R_b C_b \qquad (2.38)$$

Where m is the number of bit line segments or cells per bit line.

R_{eqcs} is the column select T-gate equivalent resistance.

C_{03} is the capacitance at node 03 of the bit line as in Fig. 2.4(a).

R_b and C_b are the resistance and capacitance of the bit line which are given in Section-2.3.

The cell access delay can be determined as,

$$T_{acell} = (m\, R_b + R_{eqcs} + R_{na})\, C_4 + R_{pc}\, C_5. \tag{2.39}$$

Where m, R_b,R_{eqcs} etc are defined as above.

R_{na} is the resistance of the cell access transistor.

R_{pc} is the resistance of the cell PMOS transistor.

C_4 is the capacitance of node 4 (cell stage node) as shown in Fig.2.4(a).

C_5 is the capacitance of node 5 (complimentary storage node) as shown in Fig 2.4(a).

The sense amplifier delay T_{sa} can be determined as, (refer to Fig.2.4(a) & 2.4(b)),

$$T_{sa} = T_{ssel} + T_{s1/0} \tag{2.40}$$

Where T_{ssel} is the signal delay to enable the sense amplifier.

$T_{s1/0}$ is the sense amplifier delay to sense an '1' or a '0'.

The sense select signal delay T_{ssel} can be determined as,

$$T_{ssel} = R_{wcn}\, C_{231} + (R_{rwp} + R_{csp})\, C_{161}. \tag{2.41}$$

Where R_{wcn} is the resistance of the read-write control driver NMOS transistor.

R_{rwp} is the resistance of the read-write PMOS transistor of the R/W-CS control-2 NOR gate.

R_{csp} is the resistance of the chip select PMOS transistor of the R/W-CS control-2

NOR gate.

C_{231} and C_{161} are the capacitances at node 231 and 161 as in Fig.2.4(a) or 2.4(b), these are given in appendix-A.

The Elmore delay model for sensing '1' is depicted in Fig.2.4(a). So, T_{s1} can be determined as,

$$T_{s1} = (R_{ns}+R_{nsc}+R_{ps})C_{14}+R_{ps}\,C_{15}+R_{nsc}\,C_{16} \tag{2.42}$$

Where R_{nsc} is the sense current transistor resistance.

R_{ns} is the sense NMOS transistor resistance.

R_{ps} is the sense PMOS transistor resistance.

C_{14}, C_{15}, C_{16} are capacitances at node 14, 15 and 16 respectively as in Fig.2.4(a), those capacitive parameters are defined in appendix-A

To sense a 0, transistors N_{sn2} & N_{snc} will be ON and P_{sp1}, P_{sp2} & N_{sn1} will be OFF. The Elmore delay model for this situation is depicted in Fig.2.4(b). In this situation, the sense amplifier delay,T_{s0} can be determined as,

$$T_{s0} = (R_{nsc}+R_{ns})C_{15} + R_{nsc}\,C_{16} \tag{2.43}$$

The output buffer delay, T_{ob} can be determined as,

Read-1 (According to Fig.2.4(a)):

$$T_{ob1} = R_{obn1}\,C_{171} + R_{obn2}\,C_{150} + R_{obp3}\,C_{201} + (R_{obp3}+R_{obp4})C_{200} \tag{2.44}$$

Read-0 (According to Fig.2.4(b)),

$$T_{ob0} = R_{obn1}\,C_{171} + R_{obp2}\,C_{150} + R_{obn3}\,C_{200} + (R_{obn3}+R_{obn4})C_{20} \tag{2.45}$$

Where resistances and capacitances are depicted in Fig.2.4(a) & 2.4(b), the expressions are given in appendix-A. Also $C_{db\sim}$, $C_{sb\sim}$, $C_{g\sim}$ etc. are defined in appendix-A.1.

Fig.2.4(c) and 2.4(d) shows a comparative analysis of our fitted mathematical modeling results with SPICE simulation. It can be inferred that our simple RC model of a SRAM

circuit induces an excellent source of estimating access time which confirm with SPICE simulation results with very little error (less than 10%).

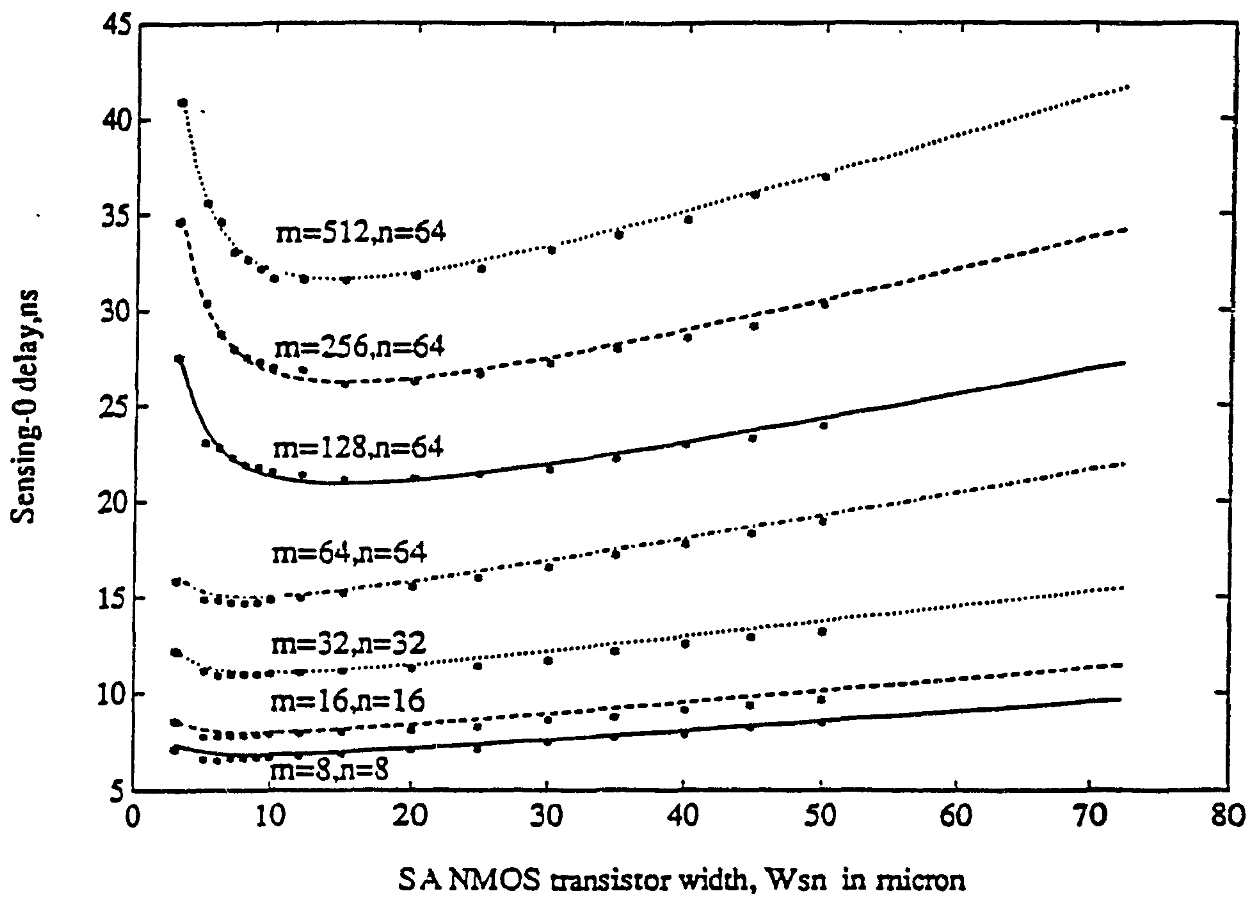


Fig. 2.4(c) Read-0: Comparison of analytical and SPICE simulation results.

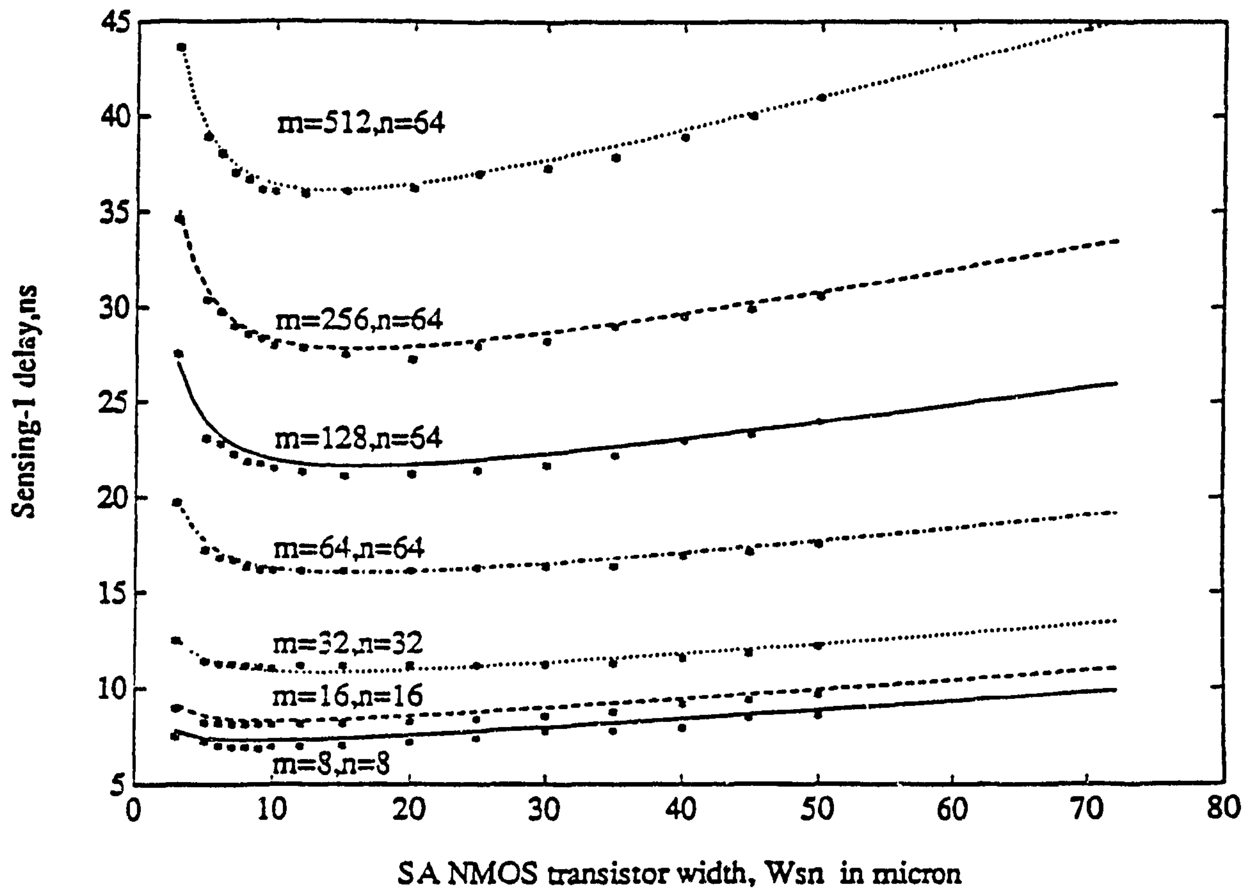


Fig. 2.4(d) Read- 1: Comparison of analytical and SPICE simulation results.

2.5 Write Access Delay

The signal delay path with equivalent RC modeling using the Elmore delay model to write an '1' and a '0' is shown in Fig.2.5(a) and Fig.2.5(b) respectively. The write signal delay consists of the write amplifier delay, data bus delay, bit line delay and cell access delay.

According to Elmore's delay model the write access signal delay T_{write} can be written as following:

$$T_{write} = T_{wa_0/1} + T_{wbit\text{-}0/1} + T_{a_0/1}. \qquad (2.46)$$

Where $T_{wa_0/1}$ is the write amplifier & data bus delay.

$T_{wbit_0/1}$ is the write path bit line delay

$T_{a_0/1}$ is the Cell access delay.

According to Fig.2.5(a) the write amplifier and data bus delay to write an '1' T_{wa_1} can be determined as,

$$T_{wa_1} = T_{wb1_1} + T_{wb2_1} + T_{wpas_1} + T_{rwcs}. \qquad (2.47)$$

Where T_{wb1_1} is write buffer-1 delay to write-1 which can be written as,

$$Twb_1 = R_{nb1}\, C_{21}. \qquad (2.48)$$

T_{wb2_1} is the write buffer-2 delay to write-1 which can be written as,

$$T_{wb2_1} = R_{pb2}\, C_{120}. \qquad (2.49)$$

T_{wpas_1} is the write T-gate delay which can be written as,

$$T_{wpas1_1} = (R_{pb2} + R_{eqwpas})C_{12}. \qquad (2.50)$$

T_{rwcs} is the read/write, Chip select gate delay which can be written as,

$$T_{rwcs} = R_{prw}\, C_{241} + (R_{prw} + R_{pcs})C_{17} + R_{redn}\, C_{170}. \qquad (2.51)$$

Also, according to Fig.2.5(b) the write amplifier and data bus delay to write a '0'

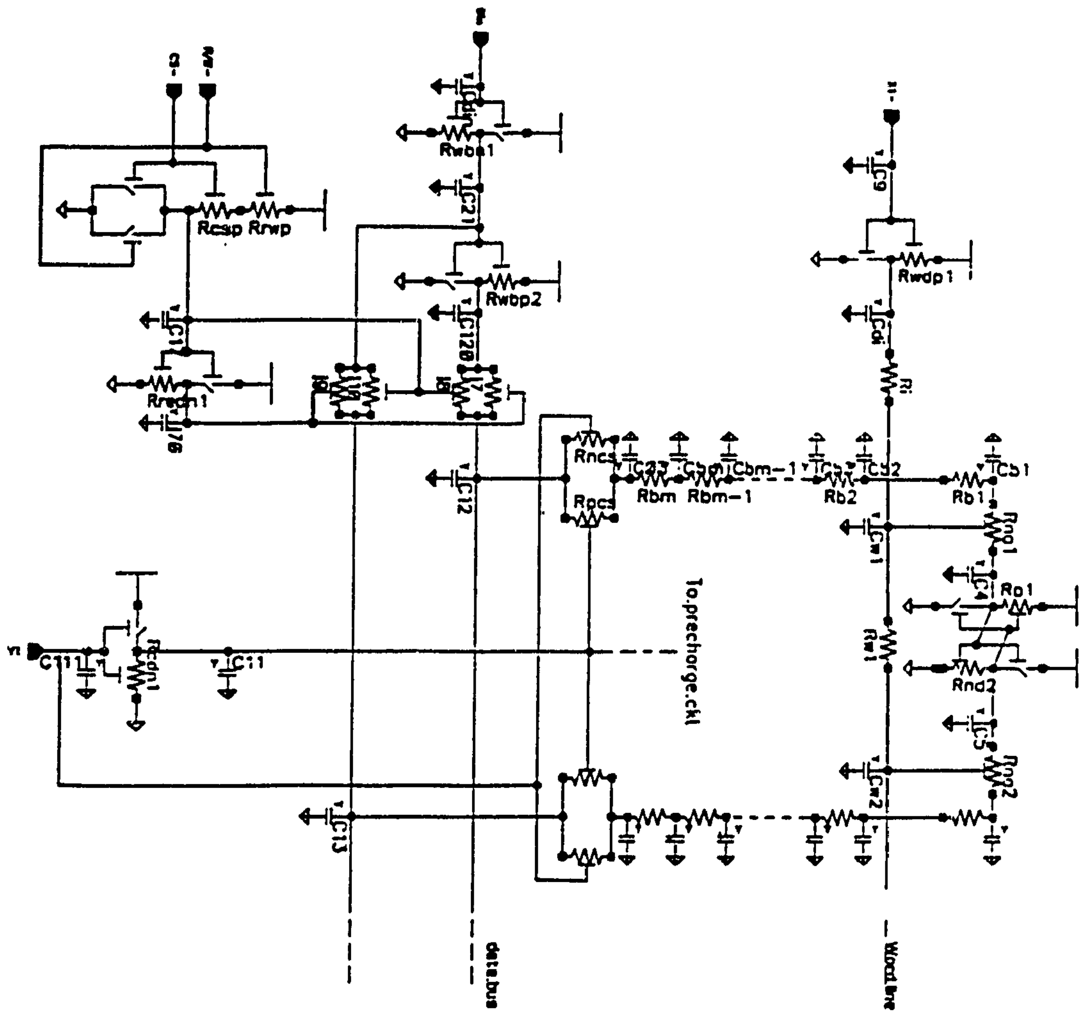


Fig. 2.5(a) Write-1 delay modeling

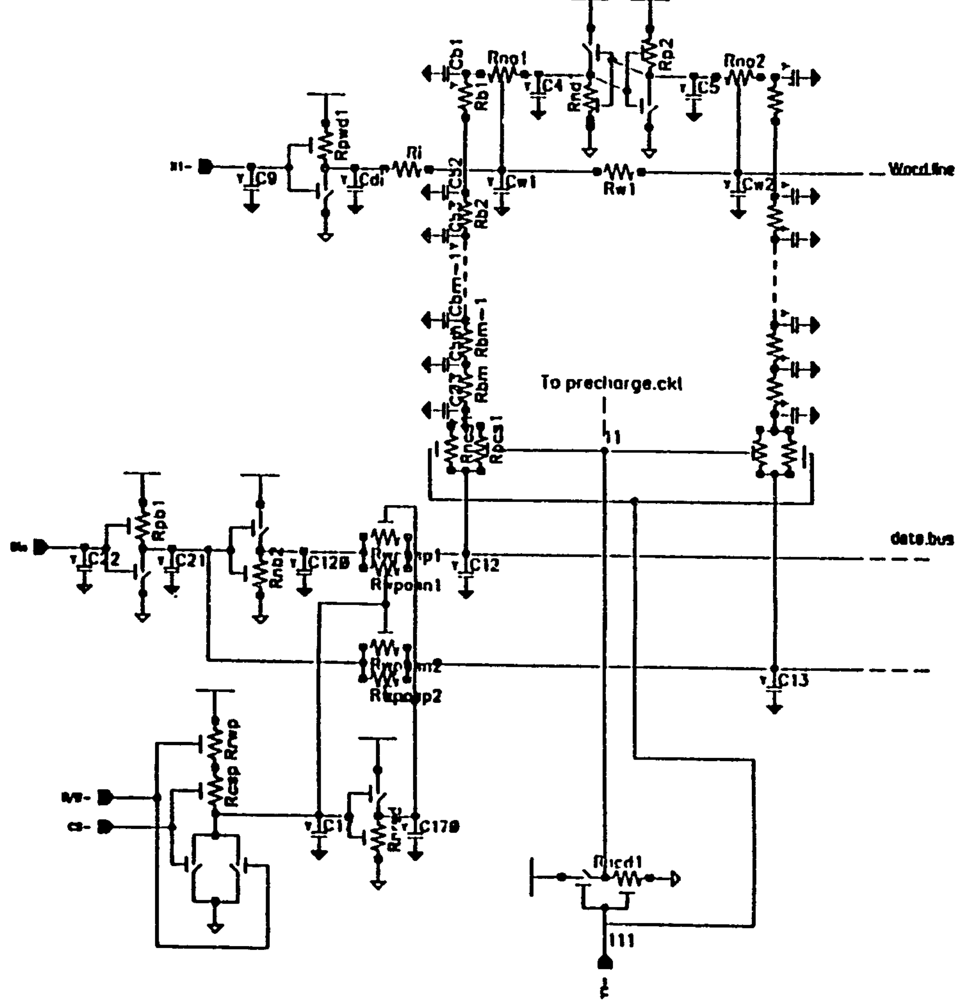


Fig. 2.5(b) Write-0 delay modeling

T_{wa_0} can be determined as,

$$T_{wa_0} = T_{wb1_0} + T_{wb2_0} + T_{wpas_0} + T_{rwcs} \qquad (2.52)$$

Where T_{wb1_0} is the write buffer-1 delay to write-0 which can be written as,

$$T_{wb1_0} = R_{pb1} C_{21}. \qquad (2.53)$$

T_{wb2_0} is the write buffer-2 delay to write-0 which can be written as,

$$T_{wb2_0} = R_{nb2} C_{120}. \qquad (2.54)$$

T_{wpas_0} is the write T-gate delay to write-0 which can be written as,

$$T_{wpas_0} = (R_{nb2}+R_{eqwpas}) C_{12}. \qquad (2.55)$$

The Elmore write path bit line delay model depicted in Fig. 2.5(a) & 2.5(b) which can be determined as following:

$$T_{wbit\ 1} = \{R_{pb2} + (R_{eqwpas} + R_{eqcspas})\} C_{03} + m\{R_{pb2} + (R_{eqwpas} + R_{eqcspas})\} C_b + \frac{m(m+1)}{2} R_b C_b \qquad (2.56)$$

$$T_{wbit\ 0} = \{R_{nb2} + (R_{eqwpas} + R_{eqcspas})\} C_{03} + m\{R_{rnb2} + (R_{eqwpas} + R_{eqcspas})\} C_b + \frac{m(m+1)}{2} R_b C_b \qquad (2.57)$$

Where T_{wbit_1} and $T_{wbit\text{-}0}$ is the bit line delay due to write an '1' and a '0' respectively.

R_{pb2} is the resistance of the write driver-2 PMOS transistor.

R_{nb2} is the resistance of the write driver-2 NMOS transistor.

R_{eqwpas} is the equivalent resistance of the write T-gate transistors.

$R_{eqcspas}$ is the equivalent resistance of the column select T-gate transistors.

R_b, C_b and C_{03} are as defined in Section-2.4.

m is the number of cells connected per bit line or number of bit line segments.

The cell access delay for writing T_{a_1} and T_{a_o} for writing '1' and '0' respectively can be determined as,

$$T_{a_1} = (R_{pb2} + R_{eqwpas} + R_{eqcspas} + m\,R_b + R_{na})C_4 + R_{nd2}\,C_5. \qquad (2.58)$$

$$T_{a_0} = (R_{nb2} + R_{eqwpas} + R_{eqcspas} + m\,R_b + R_{na})C_4 + R_{p2}\,C_5. \qquad (2.59)$$

Where R_{pb2}, R_{nb2}, R_{eqwpas}, $R_{eqcspas}$, m, R_b is defined as above.

R_{na} is the resistance of the cell access transistor.

R_{nd2} is the resistance of the cell driver NMOS transistor.

R_{p2} is the resistance of the cell PMOS transistor.

C_4 and C_5 are as defined in Section 2.4.

Also, according to Fig.2.5(a) and 2.5(b) it can be written,

$$C_{21} = C_{dbpwb1} + C_{dbnwb1} + C_{sbwnpas} + C_{sbwppas} + C_{gpwb2} + C_{gnwb2} \qquad (2.60)$$

$$C_{120} = C_{dbpwb2} + C_{dbnwb2} + C_{sbwnpas} + C_{sbwppas} \qquad (2.61)$$

$$C_{12} = C_{dbwnpas} + C_{dbwppas} + C_{gns} + C_{sbcsnpas} + C_{sbcsppas} + n\,C_{dbus} \qquad (2.62)$$

$$C_{241} = C_{dbrwp} + C_{sbcsp} \qquad (2.63)$$

$$C_{17} = C_{dbcsp} + C_{dbrwn} + C_{dbcsn} + C_{gredp} + C_{gredn} + 2C_{gwnpas}. \qquad (2.64)$$

$$C_{03} = C_{dbcsnpas} + C_{dbcsppas} + C_{dbprp} + C_{ib}/2. \qquad (2.65)$$

$$C_4 = C_5 = C_{dbna} + C_{dbcp} + C_{dbcnd} + C_{gcp} + C_{gcnd}. \qquad (2.66)$$

$C_{db\sim}$, $C_{sb\sim}$, $C_{g\sim}$, etc. are given in appendix-A.

Fig.4.1(c) shows our mathematical modeling result for write access delay and area delay square product (AT^2 of write amplifier) as a function of write buffer size with $\beta_1 = W_p/W_n = 3$ as parameter.

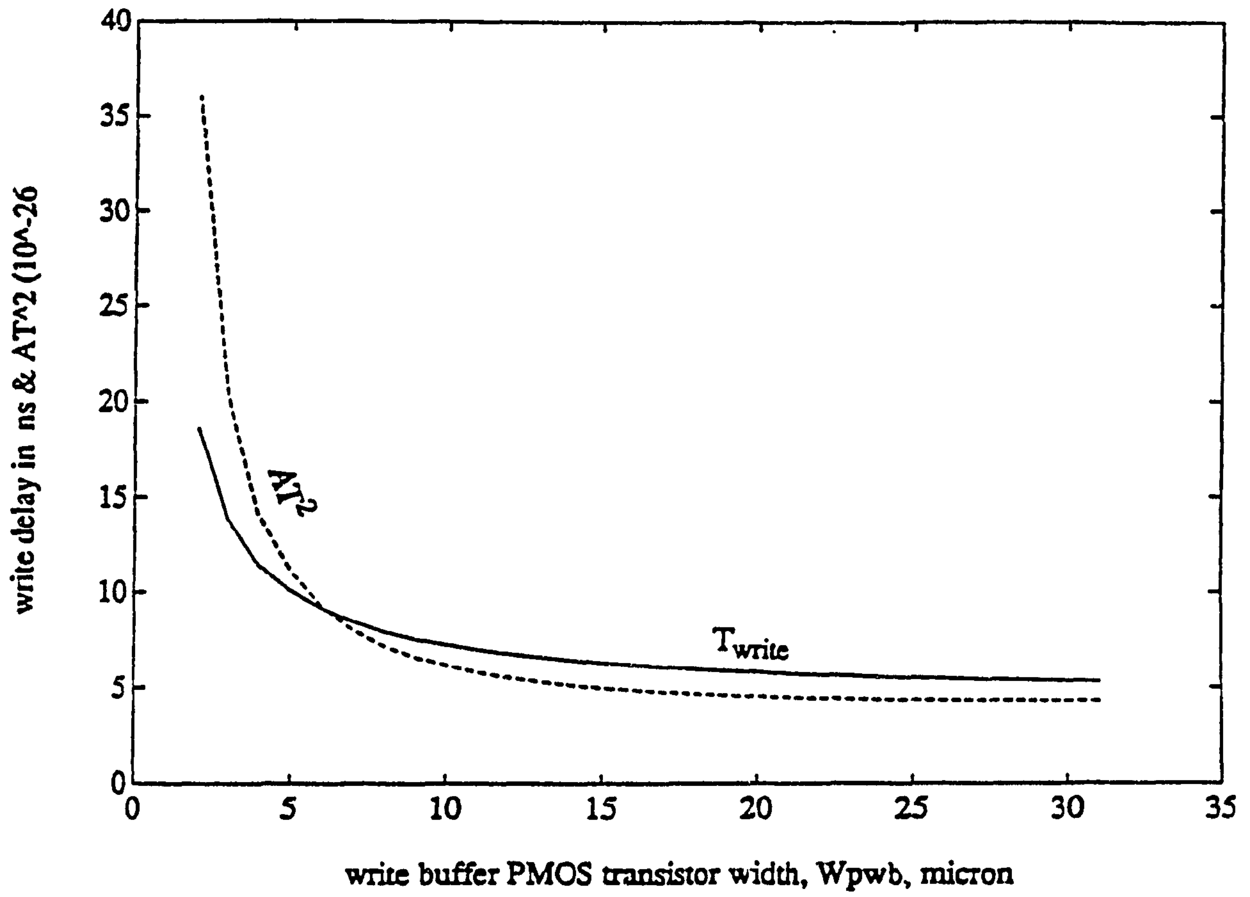


Fig. 2.5(c) Write-1 access delay & AT^2 as a function of write buffer size.

Chapter 3

Area Model of SRAM

3.0 SRAM Area

The SRAM cell array occupies at least 50% of the whole chip area. The next major area contribution is due to the X-Y decoding system. The total area of the SRAM consists of cell a array, an x-y decoder system, read & write amplifiers and other peripheral circuits.

Therefore, the area of the whole SRAM structure A_S can be written as,

$$A_s = \sum_{i=1,m;j=1,n} A_{cell(i,j)} + A_{cd} + A_{rd} + A_{sa} + \sum_{j=1,n} A_{pcj} + A_{pcbus} + 2\sum_{j=1,n} A_{csj} + A_{iop} \qquad (3.1)$$

Where $A_{cell(i,j)}$ is the area of each SRAM cell.

A_{cd} is the total area of column decoder including column driver.

A_{rd} is the total area of the row decoder including row drivers.

A_{sa} is the area of sense amplifiers.

A_{pcj} is the area of each precharge circuit.

A_{pcbus} is the area of the data bus precharge circuit.

A_{csj} is the area of each column select T-gate.

A_{iop} is the area of I/O and other peripheral circuits.

In our present analysis we assume in general,

$$L_d = L_{dn} = L_{sn} = L_{dp} = L_{sp}, \text{ and } L = L_n = L_p \qquad (3.2)$$

Where L_d is the diffusion length of N or P transistor.

L_{dn} is the length of the drain diffusion region of a NMOS transistor.

L_{sn} is the length of the source diffusion region of a NMOS transistor.

L_{dp} is the length of the drain diffusion region of a PMOS transistor.

L_{sp} is the length of the source diffusion region of a PMOS transistor.

L is the channel length of any N or P transistor.

L_n and L_p are the channel lengths of any N and P transistor respectively.

3.1 SRAM Cell Area

The 6T SRAM cell consists of two cross coupled inverters and two access pass-transistors. Therefore, the area of the cell can be approximated to,

$$A_{cell(i,j)} = 2\ (A_{access} + A_{inv}) \tag{3.3}$$

Where A_{access} is the area of cell access transistor.

A_{inv} is the area of a cell inverter.

The top view of the cell access transistors to determine their area is shown in Fig. 3.1(a). The area of each cell access transistor A_{access} can be determined as,

$$A_{access} = W_{na} L_{dn} + W_{na} L_n + W_{na} L_{sn}. \tag{3.4}$$

Where W_{na}, and L_n are the channel width and length of the NMOS cell access transistor.

L_{dn} is the length of the drain diffusion region of the NMOS cell access transistor.

L_{sn} is the length of the source diffusion region of the NMOS cell access transistor.

Using equation (3.2), equation (3.4) becomes,

$$A_{access} = 2\ W_{na} L_d + W_{na} L_n = W_a L_n(1 + 2\ L_d / L_n). \tag{3.5}$$

Assume, $\lambda_1 = 1 + 2\ L_d / L_n.$ (3.6)

Therefore, $A_{access} = \lambda_1\ W_{na} L_n.$ (3.7)

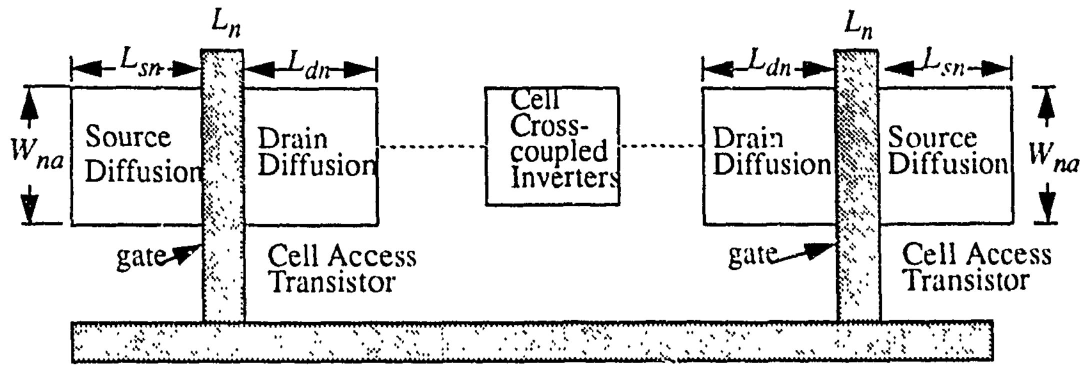


Fig. 3.1(a) Top view of a SRAM cell access transistors

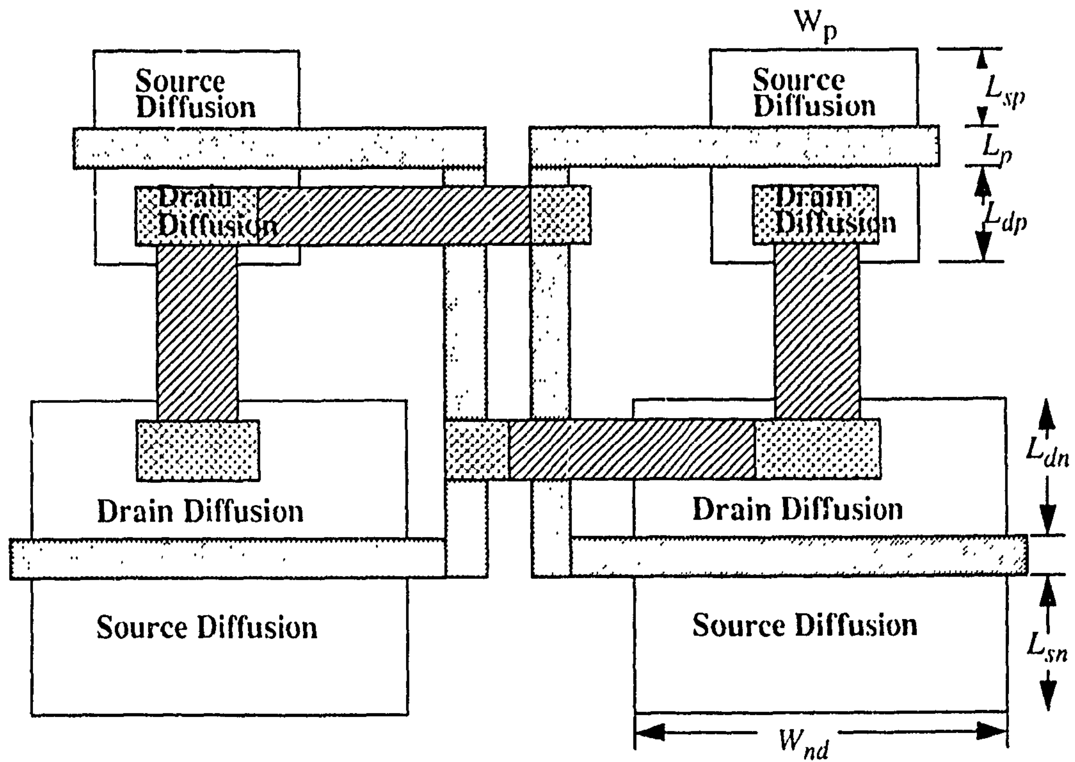


Fig. 3.1(b) Top view of a cross-coupled inverter in a SRAM cell

The top view of the physical representation of the cell cross-coupled inverter is shown in Fig. 3.1(b). As given in equation (3.2), we assume equal source and drain diffusion length and width for the same transistor. Therefore, the area of each cell inverter A_{inv} can be determined as,

$$A_{inv} = W_{nd} L_n + 2\, W_{nd} L_d + W_p L_p + 2\, W_p L_d. \tag{3.8}$$

Where W_{nd}, and L_n are the channel width and length of a cell inverter NMOS driver transistor.

W_p, and L_p are the channel width and length of a cell inverter PMOS pull-up transistor.

L_d is the diffusion length of N or P transistor's.

Thus, from equation (3.6) & (3.8) we can write,

$$A_{inv} = W_{nd} L (1 + 2\, L_d/L) + W_p L (1+2\, L_d/L) = \lambda_1 L (W_{nd} + W_p). \tag{3.9}$$

Using equation (3.7) & (3.9) in (3.3) we have,

$$A_{cell(i,j)} = 2\, \lambda_1 L (W_{na}+W_{nd}+W_p) \tag{3.10}$$

We have for cell stability criterion ratios, $r= W_{nd}/W_{na}$ and $p= W_{na}/W_p$, therefore equation (3.10) becomes,

$$A_{cell(i,j)} = 2\lambda_1 W_{na} L \left(1 + r + \frac{1}{p}\right) \tag{3.11}$$

According to noise margin and cell stability criterion the best value of r ranges from 2 to 3. The cell PMOS transistor width W_p is considered to be minimum sized according to technology and design rule. Assuming r as constant value which ensures the stability of the cell, then the cell area according to equation (3.11) is linearly dependent on cell access transistor width W_{na}. An increase of cell access transistor width will increase the overall cell area and accordingly increase the SRAM array area by an O(mn). Where m and n are the number of rows and columns respectively.

3.2 Decoder Area

The total area of the column or row decoder depends upon the size of the cell array per block, total number of blocks of cell array, the decoder hierarchy and the type of circuits to be used. The total row or column decoder system area can be determined as,

$$A_d = \sum A_{ab} + A_{gd} + \sum A_{sgd} + \sum A_{ld} + \sum A_{di} \tag{3.12}$$

where A_{ab} is the area of each address buffer.

A_{gd} is the area of the global row or column decoder.

A_{sgd} is the area of each sub-global row or column decoder.

A_{ld} is the area of each local row or column decoder.

A_{di} is the area of each row or column driver.

The estimation of A_{gd}, A_{sgd}, and A_{ld} is very straight forward and depends upon the type of implementation. The expression for the area of the row or column driver A_{di} will be similar to that of equation (3.8).

3.3 Sense Amplifier Area

The area of the sense amplifier depends upon the SRAM array size, sensing scheme hierarchy, number of output bits and nature of circuit to be used. The following formula can be used for total sense amplifier area estimation:

$$A_{tsa} = 2^p \, b \, A_{sa} \tag{3.13}$$

Where A_{tsa} = total sense amplifier area.

p=0 to h.

h is the number of sense hierarchy.

b is the number of bits.

A_{sa} is the area of each sense amplifier.

Due to higher differential mode gain of the CMOS Current Mirror Amplifier we choose it for the sensing scheme. The circuit diagram for a conventional Current Mirror Sense Amplifier (CMSA) is shown in Fig. 1.16(a). The area A_{sa} of the CMSA can be determined as,

$$A_{sa} = 2(A_{sp} + A_{sn}) + A_{snc} \tag{3.14}$$

Where A_{sp} is the area of a sense PMOS transistor P_{sp1} or P_{sp2}.

A_{sn} is the area of a sense NMOS transistor N_{sn1} or N_{sn2}.

A_{snc} is the area of a sense NMOS current transistor N_{snc}.

Equation (3.14) can further be written as,

$$A_{sa} = 2(\lambda_1 W_{sp} L_n + \lambda_1 W_{sn} L_n) + \lambda_1 W_{snc} L_n$$

$$= \lambda_1 L_n [2 (W_{sp} + W_{sn}) + W_{snc}]$$

Where λ_1 is defined as in equation (3.6).

W_{sp} is the width of the sense PMOS transistor.

W_{sn} is the width of the sense NMOS transistor.

W_{snc} is the width of the sense NMOS current transistor.

Since the sense PMOS & NMOS transistors are connected in series, to ensure equal current the size of the PMOS transistor W_{sp} needs to be approximated as β times of W_{sn}. Furthermore, since current from both series paths is sunk through the current NMOS transistor, the W_{snc} can be approximated as twice that of W_{sn}. Therefore,

$$A_{sa} = 2 \lambda_1 W_{sn} L (\beta + 2) = \delta_4 W_{sn} \tag{3.15}$$

$$\text{Where } \delta_4 = 2 \lambda_1 L(\beta+2). \tag{3.15.1}$$

3.4 Precharge circuit Area

Each bit line precharge circuit consists of two main PMOS precharge transistors and a power down circuit. The area of the whole bit and bus line precharge circuit can be

determined as,

$$A_{pc} = \sum_{j=1,n} A_{pcj} + A_{pcb} \tag{3.16}$$

Where A_{pcj} is the area of each bit line precharge circuit with power down circuit.

A_{pcb} is the area of the data bus precharge circuit.

A_{pcj} can be determined as,

$$A_{pcj} = 2\ A_{pr} + A_{pdp} + A_{pdn} + A_{pdcn}. \tag{3.17}$$

Where A_{pr} is the area of each main precharge PMOS transistor which can be written as,

$$A_{pr} = W_{pr} L_p + 2\ W_{pr} L_d = \lambda_l\ W_{pr} L_p. \tag{3.18}$$

Where W_{pr} & L_p are the width and length of the precharge PMOS transistor.

A_{pdp} is the area of the power down PMOS transistor which can be written as,

$$A_{pdp} = \lambda_l\ W_{pdp} L_p. \tag{3.19}$$

Where W_{pdp} & L_p are the width and length of the power down PMOS transistor.

A_{pdn} is the area of the power down NMOS transistor which can be written as,

$$A_{pdn} = \lambda_l\ W_{pdn} L_n. \tag{3.20}$$

Where W_{pdn} & L_n are the width and length of the power down NMOS transistor.

A_{pdcn} is the area of the power down control NMOS transistor which can be written as,

$$A_{pdcn} = \lambda_l\ W_{pdcn} L_n. \tag{3.21}$$

Where W_{pdcn} & L_n are the width and length of the power down control NMOS transistor.

A_{pcb}, the area of the data bus precharge circuit can be determined from equation (3.18) as,

$$A_{pcb} = 3\ A_{pcp} = 3\ \lambda_l\ W_{pcp} L_p. \tag{3.22}$$

3.5 Column select T-gate Area

The column selection mechanism is done by using some type of column switch such

as a transmission gate. The area of all the column select T-gates A_{cstg} is very simple which can be determined as,

$$A_{cstg} = 2 \sum_{j=1,n} A_{csj} \qquad (3.23)$$

Where A_{csj} is the area of a column select T-gate which is given by,

$$A_{csj} = A_{csn} + A_{csp} \qquad (3.24)$$

Where A_{csn} & A_{csp} are the area of an NMOS & PMOS column select transistor respectively.

Assume, $L=L_p=L_n$, $\beta=W'_{csp}/W'_{csn}$ and $\lambda_1= 1+ L_d/L$.

Where W'_{csp} and W'_{csn} are the channel width of the column select P & N transistor respectively.

L_p and L_n are the channel lengths of P & N transistor respectively.

Therefore, equation (3.24) can be written as,

$$A_{csj} = \lambda_1 W'_{csn} L(1 + \beta). \qquad (3.25)$$

3.6 I/O peripheral Area

The I/O peripheral circuit may include a write amplifier (write buffers & T-gate), a Read/Write-Chip select- NOR gate, a Read/Write enable/disable buffer and Output drivers which are depicted in Fig.3.6.

The total I/O peripheral area A_{iop} can be determined as,

$$A_{iop} = A_{wb1} + A_{wb2} + 2\, A_{wpas} + A_{rwcs} + A_{rwedb} + b\, A_{ob} \qquad (3.26)$$

Where b is the number of output bit(s).

A_{wb1} is the write buffer-1 area as in Fig. 3.6 which can be written as,

$$A_{wb1} = \lambda_1 L\,(W'_{nb1} + W'_{pb1}). \qquad (3.27)$$

where W'_{nb1} and W'_{pb1} are the channel widths of buffer-1 N & P transistors

respectively.

λ_1 is as defined by equation (3.6).

L is the channel length of any N or P transistor.

A_{wb2} is the write buffer-2 area as in Fig. 3.6 which can be written as,

$$A_{wb2} = \lambda_1 L (W_{nb2} + W_{pb2}). \quad (3.28)$$

where W_{nb2} and W_{pb2} are the channel widths of buffer-2 N & P transistors respectively. λ_1 is as defined by equation (3.6).

L is the channel length of any N or P transistor.

A_{wpas} is the write T-gate area (In_TG1 or In_TG2) as in Fig. 3.6. Assuming equal sized N & P transistors it can be written as,

$$A_{wpas} = 2 \lambda_1 W_{wpas} L. \quad (3.29)$$

Where W_{wpas} is the channel width of write T-gate P or N transistor.

λ_1 is as defined by equation (3.6).

L is the channel length of any N or P transistor.

A_{rwcs} is the read/write-chip select NOR gate area as in Fig. 3.6 which can be written as,

$$A_{rwcs} = \lambda_1 L (W_{rwp} + W_{rwn} + W_{cselp} + W_{cseln}). \quad (3.30)$$

Where W_{rwp} & W_{rwn} are the channel widths of the read/ write P & N transistor respectively.

W_{cselp} &s W_{cseln} are the channel widths of the chip select P & N transistors respectively.

λ_1 is as defined by equation (3.6).

L is the channel length of any N or P transistor.

A_{rwedb} is the read/write enable disable buffer area as in Fig. 3.6 which can be written as,

$$A_{rwcedb} = \lambda_1 \, L \, (W_{rwedp} + W_{rwedn}). \tag{3.31}$$

Where W_{rwedp} & W_{rwedn} are the channel widths of the read/ write enable/ disable P & N transistors respectively.

λ_1 is as defined by equation (3.6).

L is the channel length of any N or P transistor.

A_{ob} is the output buffer area as in Fig. 3.6 which can be written as,

$$A_{ob} = \lambda_1 \, L \, (W_{obn1} + W_{obp1} + W_{obn2} + W_{obp2} + W_{obn3} + W_{obp3} + W_{obn4} + W_{obp4}). \tag{3.32}$$

Where *Wobnl, Wobpl*, etc. are the widths of the output driver N & P transistors respectively. λ_1 and L are as defined above.

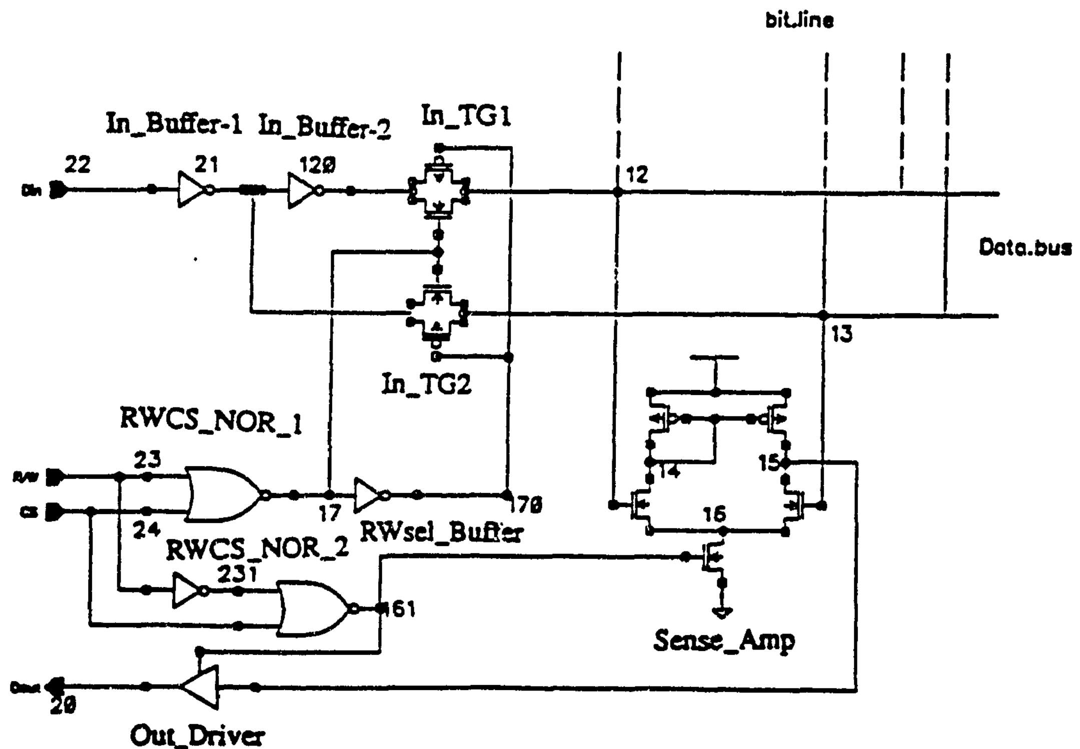


Fig. 3.6 I/O peripheral circuits of a SRAM.

Chapter 4

Power Consumption in SRAM

4.0 Introduction

The power consumption in SRAM depends upon the type of cell (4T/6T), the array size, the type of precharge scheme and other peripheral circuits. In this Chapter, we compare power consumption in SRAM using PDYCP precharge and YCL precharge in terms of mathematical models and SPICE simulations.

There may be two types of power consumption in SRAM:

1. Standby or static power.

2. Active or dynamic power (read or write).

4.1 Standby power

Assume that any cell contains either a 0 or an 1. The total static power dissipation in the SRAM matrix will be sum of the static power for the whole array and other peripheral circuits. Therefore, P_{static} can be determined as,

$$P_{static} = \sum_{i = 1, m; j = 1, n} I_{i,j} \times V_{dd} + P_{peri} \qquad (4.1)$$

Where m and n are the number of rows and columns respectively.

$I_{i,j}$ is the leakage current of each cell.

P_{peri} is the power dissipation in other peripheral circuits.

4.2 Read power

The read operation in general begins with the precharge cycle. In a Y-controlled PMOS bit line load (YCL) or conventional precharge technique all the columns are

precharged high. So, there is dynamic power dissipation during precharge. Now, whe he cell is selected to read, all the other bit lines of unselected cells still remain in the precharge mode. Thus, the dynamic power dissipation in the cell array with a precharge circuit can be estimated as follows:

$$P_{dcp(YCL)} = \sum_{j=1}^{n} \int_{0}^{t_p} P_{dpj}(t)\,dt + \sum_{j=1}^{n-1} \int_{t_p}^{T_{rc}} P_{dpj}(t)\,dt + \sum_{j=1}^{n} \int_{t_p}^{T_{rc}} P_{dc(k,j)}(t)\,dt \quad (4.2)$$

Where n is the total number of columns.

$P_{dpj}(t)$ is the precharge power in any column j for j=1 to n.

$P_{dc(k,j)}(t)$ is the power dissipation of the cell to be read and the other cells in the same row k.

k is any column number and k=1 to m where m is the total number of rows

t_p is the precharge time.

T_{rc} is the READ cycle time.

Also, $P_{dpj}(t) = V_{dd} I_j(t)$ and $P_{dc(k,j)} = V_{dd} I_{(k,j)}(t)$

where V_{dd} is the power supply voltage.

$I_j(t)$ is the current flowing in j th column.

$I_{(k,j)}(t)$ is the current flowing in any cell k, j.

Power down YCL PMOS (PDYCP) circuit dynamic READ power:

Since all the other columns during actual read time avoid precharging with our power down arrangement then, there will be no D.C. power consumption in those columns

as happens in the case of YCL precharge. Therefore, the third term in equation (4.2) will be reduced by $(n-1)P_{dc(k,j)}$ and the second term can be fully left, although there will be negligible power dissipation in the power down PMOS transistor because the size of the transistor is considered to be minimum according to technology and the design rule.

Now, in our case, the above expression (4.2) can be re-written as follows:

$$P_{dcp(PDYCP)} = \sum_{j=1}^{n} \int_{0}^{t_p} P_{dpj}(t)\,dt + \int_{t_p}^{T_{rc}} P_{dc(k,j)}(t)\,dt \qquad (4.3)$$

4.3 Write power

In a YCL or conventional SRAM during a write operation the column which is selected for write will have write power dissipation and rest of the columns will suffer from precharge power dissipation. The total dynamic WRITE cycle power dissipation $P_{dw(YCL)}$ for a conventional YCL precharge scheme can be determined as,

$$P_{dw(YCL)} = P_{dw(i,j)} + \sum_{j=1}^{n-1} \int_{0}^{T_{wc}} P_{dpj}(t)\,dt \qquad (4.4)$$

Where $P_{dw(i,j)}$ is the write power dissipation in the column under read.

P_{dpj} (t) is the precharge power dissipation in any column unselected to write.

i and j are any row and column respectively.

Power down YCL PMOS (PDYCP)

In our power down YCL precharge scheme the unselected columns are not precharged. Therefore, there will be no unwanted precharge power dissipation during a write operation. So, in this case there will be an excellent amount of power saving using the PDYCP scheme. Thus, equation (4.4) becomes,

$$P_{dw(PDYCP)} = P_{dw(i,j)} \qquad (4.5)$$

4.4 Standby mode power dissipation

For a YCL or conventional precharged SRAM, the standby power consists of static power, P_s, in the cell array and other peripheral circuits plus the power dissipation, $P_{dpj}(t)$, due to continuous bit line precharges So, we can write in general,

$$P_{standby} = P_s + \sum_{j=1}^{n} \int_{0}^{T_s} P_{dpj}(t)\,dt \qquad (4.6)$$

Where T_s is the standby cycle time.

Since our PDYCP circuit avoids unnecessary precharge of the columns, the power consumption will be static power, P_s, in the cell array and other peripheral circuits plus power dissipation, P_{dpd}, by the power down PMOS transistor. Therefore, in our case equation (4.6) becomes,

$$P_{standby} = P_s + n\,P_{dpd} \qquad (4.7)$$

Results:

In the above sections we mathematically proved that our PDYCP precharge circuit results in a significant reduction of power dissipation in any mode of operation of SRAM.

Fig.4.4(a) and (b) shows our SPICE simulation results for read and write operations using YCL (Fig. 1.13) and PDYCP (Fig. 1.14) circuits. Moreover, Fig. 4.4(c) shows a comparison of YCL and PDYCP precharge SPICE simulation results for average power dissipation of a cycle (0-150ns) of consecutive read-write operations (0-30ns: read cell 11, 30-60ns: write cell 12, 60-90ns: read cell 12, 90-120ns: write cell 11, 120-150ns: write cell 12) in two cells of different columns. It can be concluded that our PDYCP circuit works as an excellent source of power reduction which is less than 50% of the conventional one.

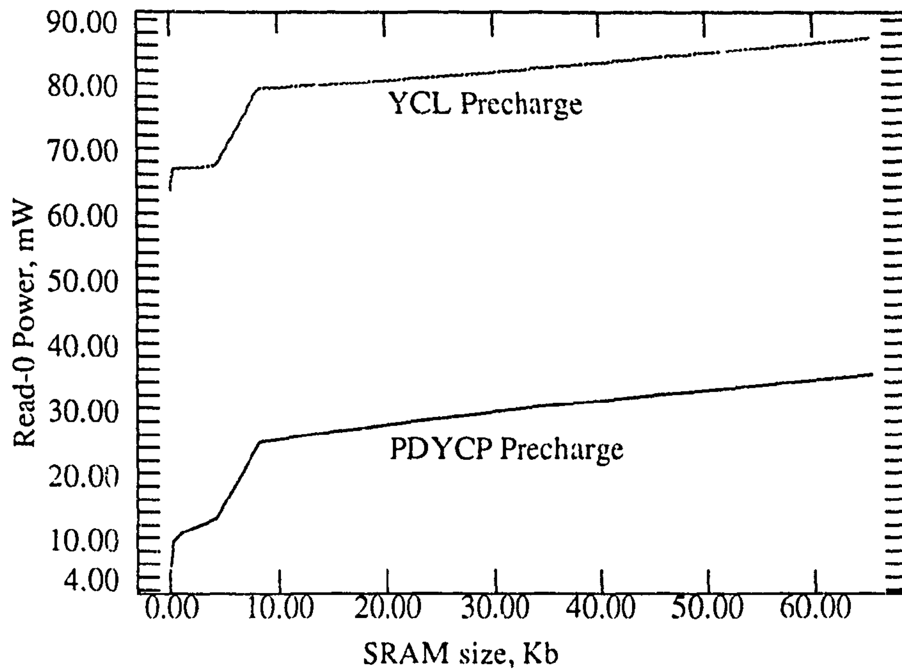


Fig.4.4(a) SPICE Comparison of Read Power dissipation using PDYCP and YCL Precharge circuit

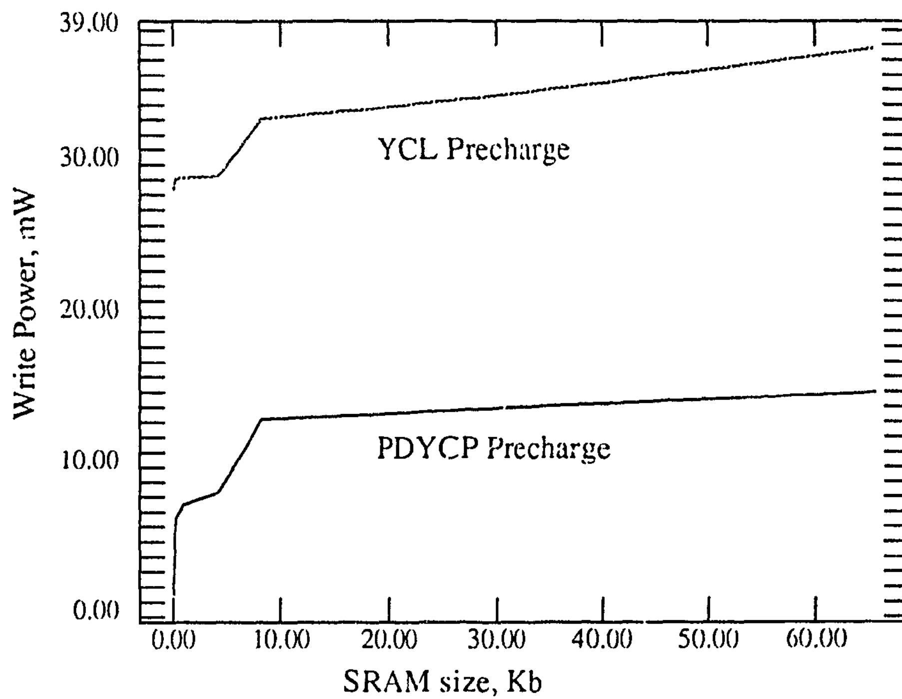


Fig. 4.4(b) SPICE Comparison of Write Power dissipation using PDYCP and YCL Precharge circuit

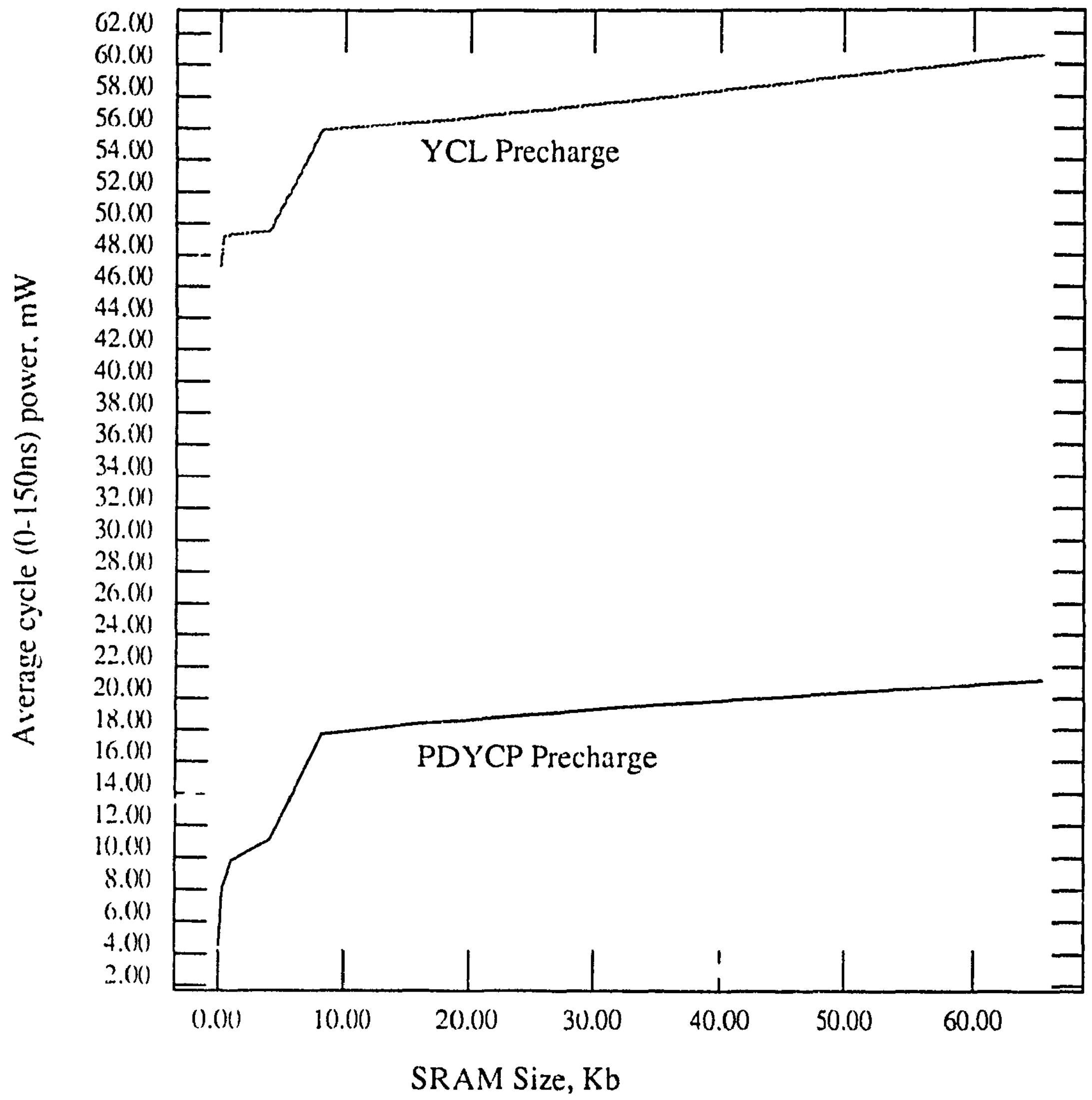


Fig. 4.4(c) SPICE Comparison of average cycle po.ver.

Chapter 5

SRAM Design Optimization

5.1 Introduction

In general, the optimization problem in VLSI design deals with the objective functions such as area, delay and power. The problem can be viewed from two different aspects such as, unconstrained and constrained based optimization. In unconstrained based optimization, minimization of one, for example, area/power maximizes the other, for example, the delay. Since, the unconstrained based optimization is insufficient so it is necessary to use constrained based optimization in order to have a better trade-off between the factors. Therefore, we emphasize on the constrained based optimization in our work. The details of analysis for the optimized design of some circuits in a SRAM are presented in appendix B.

The key approach to constrained based optimality are Khun-Tucker Equations which is explained in the following Section.

5.2 Kuhn-Tucker Equations

The general optimization problem can be formulated as follows:

$$\text{minimize} \quad f(x)$$

$$\text{such that} \quad g_i(x) \le 0, \qquad 1 \le i \le m. \tag{5.1}$$

A region R (region of acceptability) is defined as,

$$R \equiv \{x | g_i(x) \le 0, 1 \le i \le m\} \tag{5.2}$$

The optimal solution of (5.1) as depicted in [31] is given by KT's necessary and sufficient condition theorem as follows:

KT's Necessity Theorem: If f and g_i are differentiable at x^* and for any z, $z^T(\delta g_i(x^*)/\delta x)$ ≤ 0 for all i where $g_i(x^*) = 0$, where z is an element of x pointing into R from x^* and z^T is the transpose of z, then the necessary conditions that x^* be a local minimum of (5.1) is that there exist λ_i^*, $1 \leq i \leq m$, such that,

$$g_i(x^*) \leq 0$$

$$\lambda_i^* g_i(x^*) = 0 \qquad \lambda_i^* \geq 0 \tag{5.3}$$

and,

$$\frac{\delta f(x^*)}{\delta x} + \sum_{i=1}^{m} \lambda_i^* \frac{\delta g_i(x^*)}{\delta x} = 0. \tag{5.4}$$

Where λ_i^* are called the Lagrange multipliers.

If the problem is a so-called convex programming problem, that is $f(x)$ and $g_i(x)$, $1 \leq i \leq m$ are convex functions, then the KT conditions are both necessary and sufficient.

KT sufficiency Theorem: If f and g_i are convex and continuously differentiable, then a sufficient condition that x^* is a global minimum of (5.1) is that there exist multipliers λ_i^* such that the KT equations are satisfied.

5.3 Sizing and Optimization of area, delay, and Power in Logic Blocks

The signal delay, T_d, through a logic gate can be written in terms of Elmore's delay model as follows:

$$T_d = \sum RC \tag{5.5}$$

The minimum delay can be obtained by differentiating Eq. (5.5) with respect to transistor size W. i.e.,

$$\frac{dT_d}{dW} = 0 \tag{5.6}$$

Eq. (5.6) can be solved for $W(T_{dmin})$. The delay obtained in the above case is not

practical, because when the delay time approaches towards minimum, the silicon area of the gate increases rapidly. It is observed that for a 10% improvement in speed at minimum delay, area increases by twice more. So, it is reasonable to set a design goal to reach the optimum with a trade-off between speed, area & power.

Now, to solve our goal directed optimization, assume T_g is the design delay goal for the circuit under optimization. The area of a transistor is a function of width which can be written as,

$$A = f(\alpha W_i) \tag{5.7}$$

Where α is a technology dependent parameter and W_i is the width of the transistor.

Now, it is necessary to find the optimized transistor sizes that will satisfy the following optimization criterion:

$$(1)\ D = T_d - T_g <= 0 \tag{5.8}$$

Equation (5.4) can be rewritten as follows which is known as Lagrange's equation:

$$(2)\quad \frac{dD_i}{dW} + \lambda_i \frac{dA_i}{dW} = 0 \tag{5.9}$$

where , λ_i = Lagrange multiplier(LM). The above criterion is used to obtain the optimum size of the transistor in the circuit under optimization for a specific goal delay. The factor λ_i is used as a slack variable in the optimization criterion which indicates how far the design goal deviates from the minimum. The above described optimization approach provides local optimum for each logic block.

For the Static RAM design, optimization for certain circuits such as a write amplifier, and different T-gates switches are not very essential because their contribution in the chip area is very minimal. In this case the optimum sizes of those circuit transistors are assumed from our numerous SPICE simulation experience.

5.3.1 Optimum Sizing Algorithm of Transistors in a SRAM

A heuristic approach is used as an initial guess of sizing the transistors in the entire SRAM structure. Since, the heuristic approach does not guarantee the optimal solution so, the optimization criteria described in Section-5.2 is used in the following steps to reach the optimum. If for some of the circuits the initial heuristic transistor size satisfies the approximate design goal then those blocks need not be considered for the second step. In the second step the critical path transistor which mostly affects the design goal is selected for optimization by fixing others from standard heuristic value or design data sheet.

The algorithm is straight forward. Its input is the SRAM design specification that is invoked initially. In the second step the design goal of each logic block is set accordingly. In the next step the heuristic initial sizing of all the transistors in the specified SRAM is done as a preliminary step of sizing. Then the major sizing step continues to meet the design goal and optimization criteria. As a bench mark of sizing of the transistor the width of the critical transistor (say, word driver) in the delay path is calculated for minimum delay with $dT/dw=0$ for $\lambda=0$. Next, an initial guess of λ is set (say, $\lambda(W_{min})$), then, a binary search is used to calculate the best value of λ to meet the optimized design goal.The same process is repeated for the other logic blocks.

Before stating the algorithm let us define the following:

i is the number of iteration.

j is the number of circuits under optimization.

k is the number of transistors to be sized in the j th circuit block.

T^j_{dmin} is the minimum delay of circuit j in i th iteration.

$W_{jk}(T^j_{dmin})$ is the width of k th transistor in circuit j for T^j_{dmin}.

T^j_d is the delay obtained for j th circuit block for any lamda.

T^j_g is the goal delay for j th block.

λ is the Lagrange multiplier.

λ_{min} and λ_{max} are minimum and maximum values of lamda respectively.

W is the width of the transistor.

W_{min} is the minimum width of the transistor according to design and process rule.

λ_{oj} is the optimum value of lamda.

W_{ojk} is the optimum size of transistor k for circuit j.

$T^j{}_{do}$ is the optimum delay for circuit j.

The optimization algorithm is given below:

Algorithm OPTIMUM

1) Get the design specification: SRAM array size ?, Goal delays ?

2) Propagate delay goal, $T^j{}_g$ for each circuit under optimization.

3) Use heuristics to size the entire structure.

4) For each circuit under optimization do the following:

a) Use the delay, area and optimization models of the circuit;

b) Assume $\lambda_1 = \lambda_{min} = 0;$

c) Calculate $W_{jk}(T^j{}_{dmin})$ and $T^j{}_{dmin}$;

If $T^j{}_g < T^j{}_{dmin}$ then reassign $T^j{}_g$ and compare again. Else calculate $\lambda_2 = \lambda_{max}$ for minimum transistor size W_{min}.

d) Use binary search to reach the optimum;

Find $D = T^j{}_d - T^j{}_g$ for each λ;

Calculate W_{ijk}, $T^j{}_d$ for each λ and compare $D = T^j{}_d - T^j{}_g \leq 0$.

If $T^j{}_d$ is within 10% range of $T^j{}_g$ then done.

Compute λ_{oj}, W_{ojk} and $T^j{}_{do}$ as optimum.

5) Repeat step 4 for other circuits and generate all optimum values of λ, W and T_d.

5.4.1 Optimum Word Driver Design

The word decoding system contributes a significant amount of time slice in the access delay. As mentioned before we are interested to have an optimized design of the word driver. The major word decoding delay is due to the word line capacitance which depends mostly on the number of cells connected to the word line. Therefore, the optimum design of the word driver depends upon the word line capacitance, driving capability of the driver and moreover on the design goal choice.

As in equation (2.5) the word driver input capacitance C_{din} can be rewritten as,

$$C_{din} = C_{d(i-1)} + \alpha_1 W_p \tag{5.10}$$

Where we assume in equation (2.5), $L = L_n = L_p$, $LD_n = LD_{sn} = LD_{dn}$, $LD_p = LD_{sp} = LD_{dp}$ and $C_{ox} = C_{oxn} = C_{oxp}$. Therefore we have,,,

$$\beta = \frac{W_p}{W_n}$$

$$\delta_1 = 1 + 2\frac{LD_{dn}}{L}$$

$$\delta_2 = 1 + 2\frac{L_{dp}}{L}$$

and
$$\alpha_1 = C_{ox} L\left(\frac{\delta_1}{\beta} + \delta_2\right) \tag{5.11}$$

From equation (2.8) the word driver output capacitance C_{do} can be written as,

$$C_{do} = \alpha_2 W_p + \alpha_3 \tag{5.12}$$

where,

$$\alpha_2 = K_{eq}\left[\left(\frac{1}{\beta}\right) (CJ_n L_{dn} + 2CJSW_n) + CJ_p L_{dp} + 2CJSW_p \right] \tag{5.13}$$

$$\alpha_3 = 2K_{eq}(CJSW_n L_{dn} + CJSW_p L_{dp}) + \frac{1}{2} c_o w_w l_w \tag{5.14}$$

Where, K_{eq}, CJ_n, CJ_p, $CJSW_n$, $CJSW_p$ are as defined in Section 2.1.2.

It is observed from SPICE simulation that as the load changes the driving capability of the driver changes in a nonlinear fashion. Since the Elmore's delay model is very simple and approximate therefore, in order to have the model more accurate and comparable to SPICE it is required to add some fit constants in it. Assume a fixed SRAM block, so the driving capability of the word driver will depend upon the size of the driver. Thus, inserting fit cons ants in the places where the RC delay depends upon the size of the driver transistors thereby we will have a fair approximate SPICE fit model.

Therefore, putting equation (5.10), (5.11) and (5.12) into equation (2.24) and also adding the fit constants, we have,

$$T_{wd} = \left(\frac{1}{6}\right)\left(1 + 2\frac{V_t}{V_{dd}}\right)\tau + \frac{n(n+2)}{2}R_w C_w + b_1 R_{nin}(C_{d(i-1)} + \alpha_1 W_p)$$

$$+ b_2 \frac{\rho_p}{W_p}(\alpha_2 W_p + \alpha_3) + n b_3 \frac{\rho_p}{W_p} C_w \qquad (5.15)$$

where R_{pd} in equation (2.24) is replaced by equation (A.1.9) in appendix-A.

ρ_p is defined by equations (A.1.10) in appendix-A.

$\alpha_1, \alpha_2, \alpha_3$ are given by eq. (5.11), (513) and (5.14) respectively.

b_1, b_2 & b_3 are fit constant as a function of m (rows) or n (columns) and can be defined as,

$$b_1 = a_{11} + n\, a_{12} + n^2\, a_{13}$$

$$b_2 = a_{21} + n\, a_{22} + n^2\, a_{23}$$

$$b_3 = a_{31} + n\, a_{32} + n^2\, a_{33} \qquad (5.16)$$

Where b_1, b_2, and b_3 are fit constants obatined in the initial step of regression between equation (5.15) and the respective word delays from the SPICE

simulation.

n is the number of columns.

a_{11}, a_{12}, etc. are the fit constants obtained in the final step of regression analysis between b's and n's as in equation (5.16).

Details of the regression results are presented in Table B.1 in appendix - B.

Case-1: Delay without Area/Power constraint

The designer has the option of having the fastest possible word driver without caring about the area or power consumption. Now, from equation (B. 4) of appendix B.1.1 we have the word driver PMOS transistor size for minimum word delay,

$$W_{ptm} = \sqrt{\frac{b_2 \rho_p \alpha_3 + b_3 n \rho_p C_w}{b_1 \alpha_1 R_{nin}}} \tag{5.17}$$

Where W_{ptm} is the channel width of the word driver PMOS transistor at minimum delay.

α_1 and α_3 are as defined by eq. (5.11) and (5.14) respectively.

b_1, b_2 and b_3 are fit constants as defined in eq. (5.16)

Case-2: Delay with area/power constraint

The optimum design of the word driver in terms of area, delay and power is highly desirable. The optimality criterion in equation (5.8) & (5.9) can be used to solve the problem.

For CMOS circuit design, the area and power are linearly dependent. Therefore, we assume that an optimum area will result in optimum power. The area of a word driver can be written in terms of width as,

$$A_d = \alpha_4 W_p \tag{5.18}$$

where

$$\alpha_4 = (L+2) + \frac{1}{\beta}(L + 2L_{dn}) \tag{5.19}$$

Assume $L = L_p = L_n$ = constant for design rules.

From equation (B.8) of appendix B.1.1 we have the optimum size of the word driver PMOS transistor,

$$W_{po} = \sqrt{\frac{\rho_p (b_2 \alpha_3 + b_3 n C_w)}{b_1 \alpha_1 R_{nin} + \lambda_w \alpha_4}} \qquad (5.20)$$

The optimum size of the driver NMOS transistor can be obtained as, $W_{no}=W_{po} / \beta$.

Since delay is a function of width, then, optimum delay can be obtained using the above value of W_{po} as,

$$T_{wdo} = f(W_{po}). \qquad (5.21)$$

Table-2 below shows our analytical result for the optimized design of a word driver under variable load conditions and the design goal.

n	T_{wdm} ns	W_{ptm} μ	T_{gwo} ns	W_{po} μ	λ_w
8	0.514	91.85	0.68	17.76	1.28
16	0.609	118.19	0.73	34.61	0.572
32	0.7736	156.78	0.86	65.73	0.288
64	0.982	189.26	1.087	90.068	0.295
128	1.4038	321.11	1.58	132.01	0.299
256	2.3828	491.45	2.52	249.53	0.308

Table 2: Example of Optimized design of a word driver under variable load conditions.

Where as in Table-2,

n = number of cells connected per word line.

T_{wdm} = Minimum word driver delay.

W_{ptm} = Channel width of the word driver PMOS transistor for minimum delay.

T_{wgo} = Optimum word driver delay obtained.

W_{p0} = Optimum channel width of the word driver PMOS transistor. Assume optimum width of NMOS transistor to be W_{po}/β.

As shown in column 3 of table-2, the word driver size at minimum delay is very large, which is quite impractical. Also, if n, the number of cells connected per word line is increased more than 64, it is clear that both the word driver size and delay increases more than twice as much as in the case of n=64. The solution to this problem may be to insert multiple buffer stages in the row. But, it is observed that most of the high-performance and high-density SRAM's [1-7] are limited up to n=64. Instead of increasing n in a block the total number of blocks and the number of rows connected per bit line (m) can be increased. The effect of the increase of m will be observed in the following sections. Finally, comparing column 2 with 4 and 3 with 5, it can be concluded that with a slight increase in delay a significant amount of area or power can be saved.

5.4.2 Optimum Precharge Circuit Design

The precharge delay T_{prech} is depicted in equation (2.25). Here, Since, T_{rampi}, T_{cd} and T_{bit} are constant for a particular array size and are not a function of W_{pr}, then we assume,

$$T_k = T_{rampi} + T_{cd} + T_{bit} \tag{5.22}$$

Using equation (5.22), and adding fit constants into equation (2.25) it can be written as,

$$T_{prech} = T_k + b_{p1} T_{pd} + b_{p2} T_{pr} \tag{5.23}$$

Where b_{p1} and b_{p2} are fit constants which are given by,

$$b_{p1} = c_{11} + m\,c_{12} + m^2 c_{13}$$

$$b_{p2} = c_{21} + m\,c_{22} + m^2 c_{23} \tag{5. 24}$$

Where m is the number of rows.

T_{pd} and T_{pr} are given in equation (2.28) and (2.30) respectively.

$c_{11}, c_{12}, c_{13}, c_{21}, c_{22}, c_{23}$ are regression fit constants as presented in Table B.2 in appendix- B. 2.

Equating the expression for C_x from equation (2.29) into equation (2.28), which can further be extended as follows:

$$T_{pd} = R_{nt1}C_{10} + \frac{n(n+1)}{2}R_w C_w +$$

$$+ (R_{tn1} + R_{tn1})(2\delta_2 C_{ox} W_{pr} L_p + C_{dbnt1} + C_{dbpt1}) \quad (5.25)$$

Where, $C_{gpr} = 2\ \delta_2\ C_{ox}\ W_{pr}\ L_p$.Putting equation (2.31) into equation (2.30) which can also further be extended as,

$$T_{pr} = \frac{\rho_{pr}}{W_{pr}} K_{eq} CJ_p W_{pr} L_d + 2K_{eq} CJSW_p (W_{pr} + L_d) + mC_b + C_{dbcsp} + C_{dbcsn} \quad (5.26)$$

Case-1: Delay without area/power constraint

As before, for minimum delay criteria we can write,

$$\frac{dD}{dW_{pr}} = b_{p1}\frac{dT_{pd}}{dW_{pr}} + b_{p2}\frac{dT_{pr}}{dW_{pr}} = 0 \quad (5.27)$$

Then, putting Eq. (5.25) and (5.26) into (5.27) we have,

$$-b_{p2}\frac{\rho_{pr}}{W_{pr}^2}((2CJSW_p)L_d + mC_b + 2C_{dbcsp}) + 2b_{p1}(R_{tn} + R_{tn1})\delta_2 C_{oxp} L_p = 0 \quad (5.28)$$

Solving equation (5.28) for W_{pr} with $W_{pr} = W_{prm}$ for minimum delay and we have,

$$W_{prm} = \sqrt{\frac{b_{p2}\rho_{pr}(CJSW_p L_d + \frac{m}{2}C_b + C_{dbcsp})}{b_{p1}\delta_2 C_{oxp} L(R_{tn} + R_{tn1})}} \quad (5.29)$$

The above value of W_{pr} can be used to find the minimum value of T_{prech} from equation (5.24).

Case-2:Delay with area/power constraint

From equation (B.12) of appendix B. 1.2 we have the optimum precharge transistor size,

$$W_{pro} = \sqrt{\frac{b_{p2}\rho_{pr}(2CJSW_pL_d + mC_b + 2C_{dbcsp})}{2b_{p1}\delta_2 C_{oxp}L(R_{tn} + R_{ntn1}) + \lambda_{pr}\delta_3}} \quad (5.30)$$

Where the precharge circuit area is given by equation (3.18) as, $A_{pr}=\lambda_1 W_{pr} L_p= \delta_3 W_{pr}$

and, $\delta_3= \lambda_1 L_p$.

λ_{pr} is the Lagrange multiplier for the precharge circuit design.

Now, the optimum precharge delay would be, $T_{prech(opt)}=f(W_{pro})$.

Table-3 below shows our analytical result for the optimized design of a precharge circuit under variable load conditions and design goals.

mxn	T_{prechm} ns	$W_{pr}(T_{prechm})$ μ	T_{prog} ns	W_{pro} μ	λ_{pr}
8x8	0.347	146.39	0.523	17.28	1.386
16x16	0.376	161.61	0.66	18.18	1.148
32x32	0.49	208.93	0.761	40.58	0.734
64x64	0.575	244.91	0.942	36.42	0.388
128x64	0.665	324.49	1.076	57.68	0.368
256x64	0.747	559.9	1.25	83.72	0.320
512x64	1.268	726.23	2.031	101.67	0.856

Table 3: Example of Optimized design of a precharge circuit under variable load conditions.

Where as in Table-3,

n, m = number of cells connected per word line and bit line respectively.

T_{prechm} = Minimum precharge delay.

$W_{pr(Tprechm)}$ = Channel width of the precharge PMOS transistor for minimum delay.

T_{prog} = Optimum precharge delay obtained.

W_{pro} = Optimum channel width of the precharge PMOS transistor.

As shown in table-3, the transistor sizes in column 3 are huge for the minimum delay condition. Our optimization criteria is used to cause a significant reduction in area with little increase in delay. The resulting delays and transistor sizes are shown in column 4 and 5 respectively for different SRAM cell arrays.

5.4.3 Optimum Sense Amplifier Design

The Sense Amplifier (SA) is the key circuit for a Static RAM to have a very fast access speed. For small and medium density SRAMs the fastest possible circuit can be selected without caring about the area of the sense amplifier; the contribution of which in the overall chip area is not very significant. But for a high density SRAM, which might need a multistage SA, it is required to have a design trade-off between the objective functions.

As in Section-2.4,the total access time in a SRAM can be rewritten as,

$$T_{sense} = T_{dc} + T_{prech} + T_{bit} + T_{acell} + T_{ob} + T_{bus} + T_{s0/1} + T_{ssel} \quad (5.31)$$

Assuming all transistors except the sense amplifier are set from heuristics or from design data sheets or optimization, as in previous sections, then T_{dc}, T_{prech}, T_{bit}, T_{acell} & T_{ob} in the above equation are independent of sense amplifier transistor sizes. Therefore,

adding the fit constants for the variable parts of equation (5.31), which are SA transistor width dependent, we can write,

$$T_{sense} = T_{sk} + b_{s1}T_{bus} + b_{s2}T_{s(1/0)} + b_{s3}T_{ssel} \tag{5.32}$$

Where b_{s1}, b_{s2} and b_{s3} are sense fit constants, which are given by,

$$b_{s1}=s_{11}+n\,s_{12}+n^2s_{13}$$

$$b_{s2}=s_{21}+m\,s_{22}+m^2s_{23}$$

$$b_{s3}=s_{31}+m\,s_{32}+m^2s_{33} \tag{5.32.1}$$

Where n and m are the number of rows and columns respectively.

s_{11}, s_{12}, etc. are sense regression constants as presented in Table B.3 of appendix B.2.

Case-1: Delay without area/power constraint - Minimum delay

We have from equation (B. 29) of appendix B.1.3 the sense amplifier NMOS transistor size for minimum delay as,

$$W'_{snm} = \sqrt{\frac{b_{s2}\rho_n\left(6K_{eq}CJSW_nL_d + 3K_{eq}CJSW_pL_d + \frac{3}{2}\left(C_{gobn2} + C_{gobp2}\right)\right)}{\delta_1C_{ox}L\left(b_{s1}nR_{bus} + 4b_{s3}R_{rwp}\right)}} \tag{5.33}$$

Therefore, the minimum sensing delay can be obtained from equation (5.32) using (5.33) as,

$$T_{sense(min)} = f(W'_{snm}). \tag{5.33.1}$$

Case-2: Delay with area/power constraint

The area of the sense amplifier is a function of the width of the SA transistors, which is given by equation (3.15). Using the optimization criteria of equation (5.9), the optimum SA transistor size as determined by equation (B.32) of appendix B.1.3 can be written as,

$$W_{sno} = \sqrt{\frac{b_{s2}\rho_n (6K_{eq}CJSW_n L_d + 3K_{eq}CJSW_p L_d + \frac{3}{2}(C_{gobn2} + C_{gobp2}))}{\delta_1 C_{ox} L (b_{s1} n R_{bus} + 4 b_{s3} R_{rwp}) + \lambda_s \delta_4}} \qquad (5.34)$$

Where W_{sno} is the optimum size of the SA NMOS transistor. The optimum size of the other SA transistors can be obtained as described in Section-3.3.

ρ_n is as defined by equation (A.1.10) in appendix-A.

δ_1 is as defined by equation (5.11).

δ_4 is as defined by equation (3.15).

n is the number of columns.

R_{bus} is the data bus resistance per segment.

λ_s is the Lagrange multiplier for the SA circuit design.

Any other unknown variable in eq. (5.34) is referred in Section-2.4 & Appendix-A.

Table-4 below shows our analytical result for the optimized design of a sense amplifier under variable load conditions and design goals.

mxn	T_{sensem} ns	W_{snm} μ	T_{sgo} ns	W_{sno} μ	λ_s
8x8	6.875	8.28	7.096	5.01	1.5
16x16	8.033	8.86	8.12	6.54	0.75
32x32	10.99	9.12	11.25	7.24	1.125
64x64	15.01	9.72	15.25	6.02	1.875
128x64	21.48	17.52	21.57	15.48	0.375
256x64	26.99	17.83	27.22	13.81	1.125
512x64	32.43	17.91	32.58	14.5	0.75

Table 4: Example of the Optimized design of a sense amplifier under variable load conditions.

Where as in Table - 4,

n= number of cells connected per word line.

m = number of cells connected per bit line.

T_{sensem} = Minimum sensing-0 delay.

W_{snm} = Channel width of the sense amplifier NMOS transistor for minimum delay.

T_{sgo} = Optimum sensing delay obtained.

W_{sno} = Optimum channel width of the sense amplifier NMOS transistor.

Chapter 6
Design and Layout Implementation of a SRAM

6.0 Example: Optimized SRAM Design

In this section the result obtained by our SRAM Modeling and Optimizer Tool (SMOT) for Fig. 1.3(b) will be presented. It is assumed that the transistor sizes of the other circuits except the word driver, precharge and sense amplifier circuits are predetermined from our SPICE simulation results. The inputs of the tool are the SRAM specification and delay goals. The outputs of the tool are the optimized transistor sizes and optimized goal delays. The tool is tested for different SRAM array specifications and design goals. The following is an example of optimized transistor sizes and access delay presented by our tool for a 4kx1bit (64x64) SRAM.

Inputs of the Tool:

The run time inputs of the tool are given below:

Please input the SRAM specification :

Total No. of Rows, m --> 64

Total No. of Columns, n --> 64

Optimized Design of Word Driver :

Please input the design goal delay for m=64, n=64, Twgp --> 1.0e-9

Optimized Design of the Precharge Circuit :

Please input the design goal delay for m=64, n=64, Tprgp --> 1.0e-9

Optimized Design of Sense Amplifier :

Please input the design goal delay for m=64, n=64, Tsgp --> 15.5 e-9

Outputs of the tool:

OPTIMUM TRANSISTOR SIZES OBTAINED FROM THE TOOL :

1. SRAM Cell Design :

Access Transistor width: Wna1=Wna2= 3.200000e-06

Driver transistor width: Wnd1=Wnd2= 8.000000e-06

PMOS Pull-up transistor width: Wp1=Wp2= 2.000000e-06

2. Word Driver Design :

PMOS Transistor width: Wpd= 9.01e-05

NMOS transistor width: Wnd= 3.00e-05

3. Precharge Circuit Design :

Each Bit line Precharge Transistor width: Wpr11=Wpr21 = 2.73e-05

Each Data Bus line Precharge Transistor width: Wpr = 2.73e-05

Each Power down PMOS Precharge Transistor width: Wpt12 = 2.000000e-06

Each Power down NMOS Precharge Transistor width: Wnt11 = 2.000000e-06

4. Column Select T-gate Design :

Column Select T-gate PMOS transistor width: Wcp= 6.400000e-05

Column Select T-gate NMOS transistor width: Wcp= 6.400000e-05

5. Sense Amplifier Design :

PMOS transistor width: Wsp1=Wsp2= 5.39e-06

NMOS transistor width: Wsn1=Wsn2= 5.39e-06

NMOS Current transistor width: Wsnc= 1.08e-05

6. Write Buffer Design (assume buffer 1 & 2 identical) :

PMOS transistor width: Wpw1=Wpw2= 2.200000e-05

NMOS transistor width: Wnw1=Wnw2= 2.200000e-05

7. Data bus T-gates(assume identical T-gates) :

PMOS transistor width: Wpb1=Wpb2= 2.800000e-05

NMOS transistor width: Wnb1=Wnb2= 2.800000e-05

8. Output buffer Design :

First inverter: PMOS & NMOS transistor width: Wpo1=Wno1= 3.600000e-06

Second inverter: PMOS & NMOS transistor width: Wpo2=Wno2= 3.600000e-06

Width of the four transistors at the output end :

PMOS transistor width: Wpo3= Wpo4= 1.200000e-05

NMOS transistor width: Wno3=Wno4= 1.200000e-05

9. Read/Write Column Select Control gates 1 and 2 Design :

PMOS transistor width: Wrwp1= Wrwp2=Wcsp1=Wcsp2= 1.500000e-05

NMOS transistor width: Wrwn1=Wrwn2=Wcsn1=Wcsn2= 5.000000e-06

10. Column Driver Design :

PMOS transistor width: Wcdp= 1.500000e-05

NMOS transistor width: Wcdn= 5.000000e-06

The following Following Results are also forwarded by the Optimizer Tool:

Optimized Access Goal Delay: Tao = 1.555209e-08

Expected Access Delay: Tge = 1.550000e-08

Optimized Precharge Goal Delay: Tao = 1.074334e-09

Optimized Word Driver Goal Delay: Tao = 1.077924e-09

6.1 Layout Implementation

A 4kX1bit (64X64) SRAM layout was designed[1]. The design was implemented in a CMOS4S 1.2μ NT (Northern Telecom) technology environment using the 'MAGIC version- 6' layout tool. Each cell is designed using our SPICE simulation sizes. During the layout design, we tried to keep the diffusion gaps to a minimum depending on the logic function by honoring the design rules. This also saves a considerable amount of layout area. To reduce the resistance and drain diffusion area fewer contacts are added to the drain and source diffusion region. Since the SRAM cell contributes a major portion of the chip area, the basic cell size is kept as minimum as possible considering the design criteria as described in Section 1.4.2, and technology limitations. A basic SRAM cell layout is shown in Fig.6.1. The SRAM cell test circuit layout is depicted in Fig. 6.2(a) which is used for simulation and circuit extraction. The test is used to confirm the functionality of the cell under the worst condition. The loading effect of other cells in the bit or word line of the test circuit is represented by the sum of lumped and distributed capacitances. The cell is tested under various cases such as a READ followed by a WRITE or vice-versa.

The HSPLOT for the preceding conditions are depicted in Fig. 6.2(b). The layout of other peripheral circuits are also designed and tested, which are depicted in Fig. 6.3 to 6.5. Fig. 6.6 through 6.11 depict the performance of our design. A maximum 20ns READ access time and 13ns WRITE access time is observed, which can be verified from those plots.

1. The design was carried out as a VLSI course project with J. Manz, M. Mansour, T. Vince and the author.

Extracted Transistor Sizes from Layout:

The channel length in any circuit is $L_n = 1.2\ \mu$ for $1.2\ \mu$ technology. The widths of the transistors in different circuits used to implement a SRAM are given below:

1. SRAM Cell (Fig. 6.1- Layout, Fig. 1.3(b)- Schematic):

   Access transistor width: $W_{na1} = W_{na2} = 1.6\ \mu$.

   Driver transistor width: $W_{nd1} = W_{nd2} = 3.6\ \mu$

   PMOS pull-up transistor width: $W_{p1} = W_{p2} = 1.6\ \mu$

2. Word Driver (Fig. 6.2- Layout, Fig. 1.3(b)- Schematic)

   PMOS transistor width: $W_{pd} = 90.8\ \mu$.

   NMOS transistor width: $W_{nd} = 30.4\ \mu$.

3. Precharge Circuit (Fig. 6.3- Layout, Fig. 1.3(b)- Schematic)

   Each Bit line Precharge PMOS transistor width: $W_{pr11} = W_{pr21} = 12\ \mu$.

   Each Data bus Precharge PMOS transistor width: $W_{pr} = 12\ \mu$.

4. Column Select T-gates (Fig. 6.3- Layout, Fig. 1.3(b)- Schematic)

   PMOS transistor width: $W_{cp} = 8\ \mu$.

   NMOS transistor width: $W_{cn} = 8\ \mu$.

5. Sense Amplifier (Fig. 6.4- Layout, Fig. 1.3(b)- Schematic)

   PMOS transistor width: $W_{sp1} = W_{sp2} = 15.2\ \mu$.

   NMOS transistor width: $W_{sn1} = W_{sn2} = 3.6\ \mu$.

   NMOS current transistor width: $W_{snc} = 15.2$ m.

6. Write Buffer (Fig. 6.5- Layout, Fig. 1.3(b)- Schematic)

   Driver 1 and 2 PMOS transistors width: $W_{pw1} = W_{pw2} = 22.0\ \mu$.

   Driver 1 and 2 NMOS transistors width: $W_{nw1} = W_{nw2} = 22.0\ \mu$.

7. Data bus T-gates (Fig. 6.5- Layout, Fig. 1.3(b)- Schematic)

   PMOS transistor width: $W_{pdb1} = W_{pdb2} = 27.6\ \mu$.

   NMOS transistor width: $W_{ndb1} = W_{ndb2} = 27.6\ \mu$.

8. Output buffer (Fig. 6.5- Layout, Fig. 1.3(b)- Schematic)

First inverter: PMOS and NMOS transistor width: $W_{po1} = W_{no1} = 3.6\ \mu$.

Second inverter: PMOS and NMOS transistor width: $W_{po2} = W_{no2} = 3.6\ \mu$.

Width of the transistors at the output:

PMOS transistor width: $W_{po3} = W_{po4} = 12\ \mu$.

PMOS transistor width: $W_{no3} = W_{no4} = 12\ \mu$.

9. Read/Write Column Select Control gates 1 and 2 (Fig. 6.5- Layout, Fig. 1.3(b)- Schematic)

PMOS transistor width: $W_{rwp1} = W_{rwp2} = W_{csp1} = W_{csp2} = 12.4\ \mu$.

NMOS transistor width: $W_{rwn1} = W_{rwn2} = W_{csn1} = W_{csn2} = 7.2\ \mu$.

10. Column driver (Fig. 6.2(a)- Layout, Fig. 1.3(b)- Schematic)

PMOS transistor width: $W_{cdp} = 15.6\ \mu$.

NMOS transistor width: $W_{cdn} = 5.2\ \mu$.

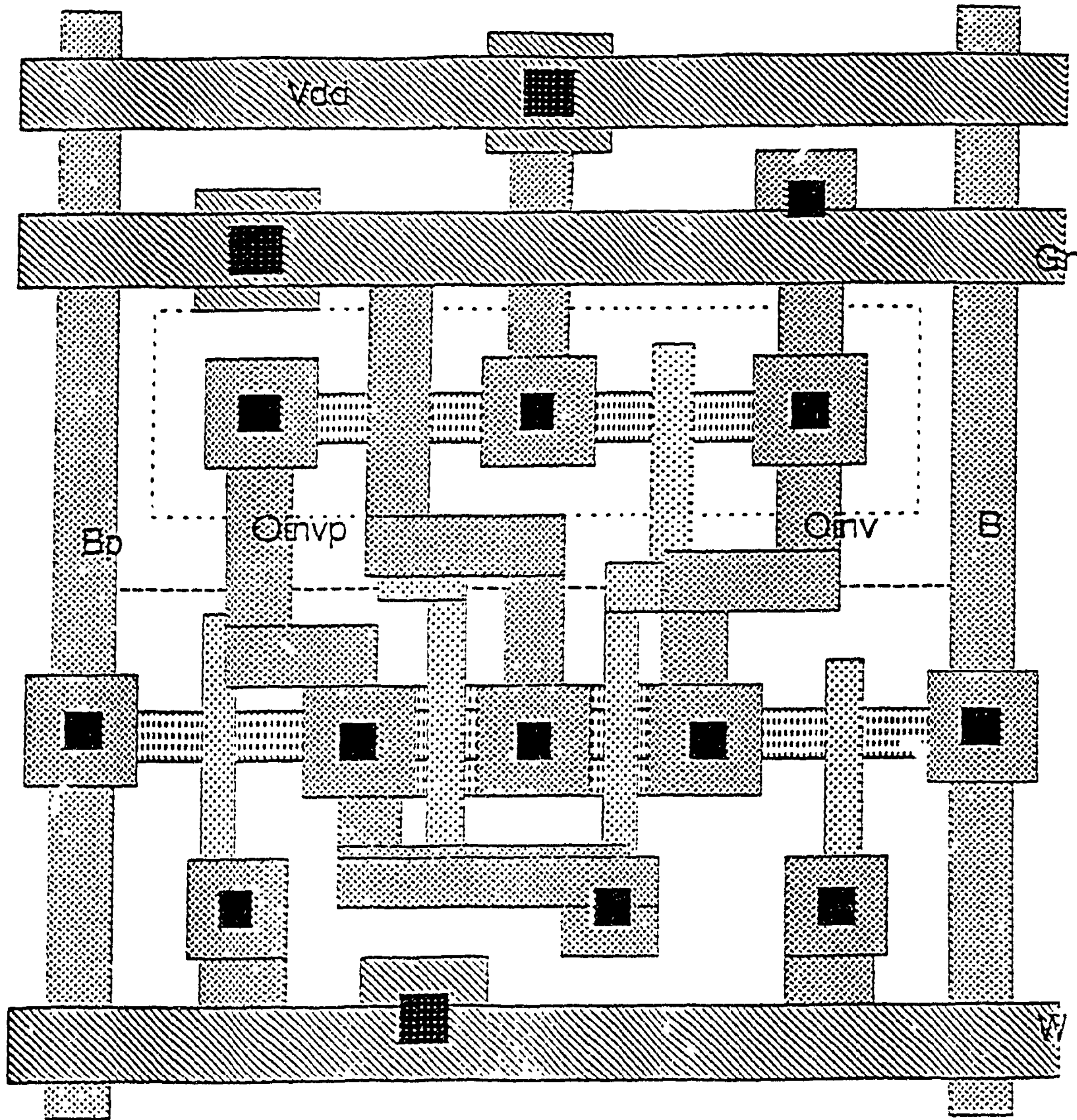


Fig. 6.1 Layout of a SRAM cell.

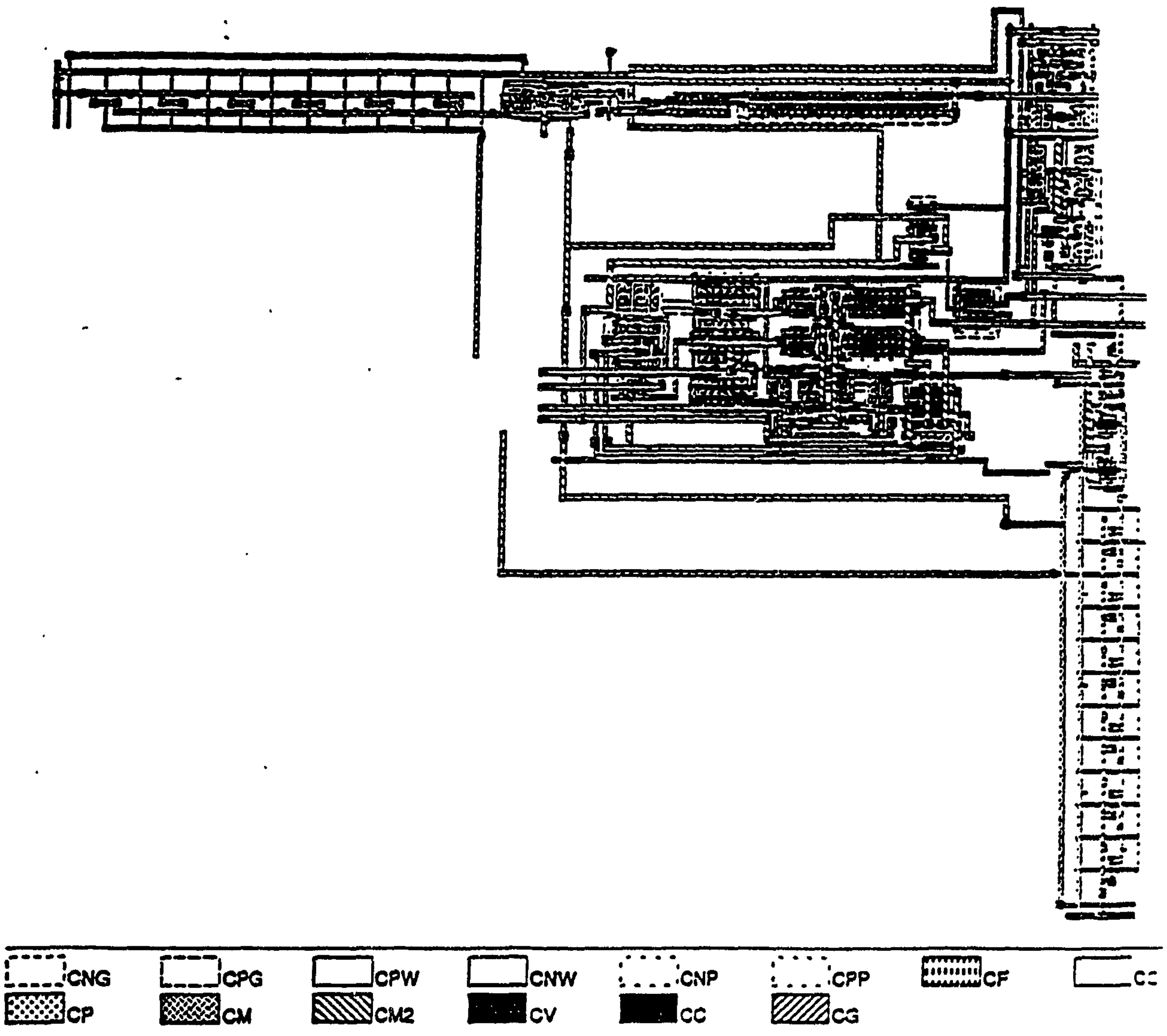


Fig. 6.2(a) Test circuit of a SRAM cell (includes cell, decoder and peripherals)

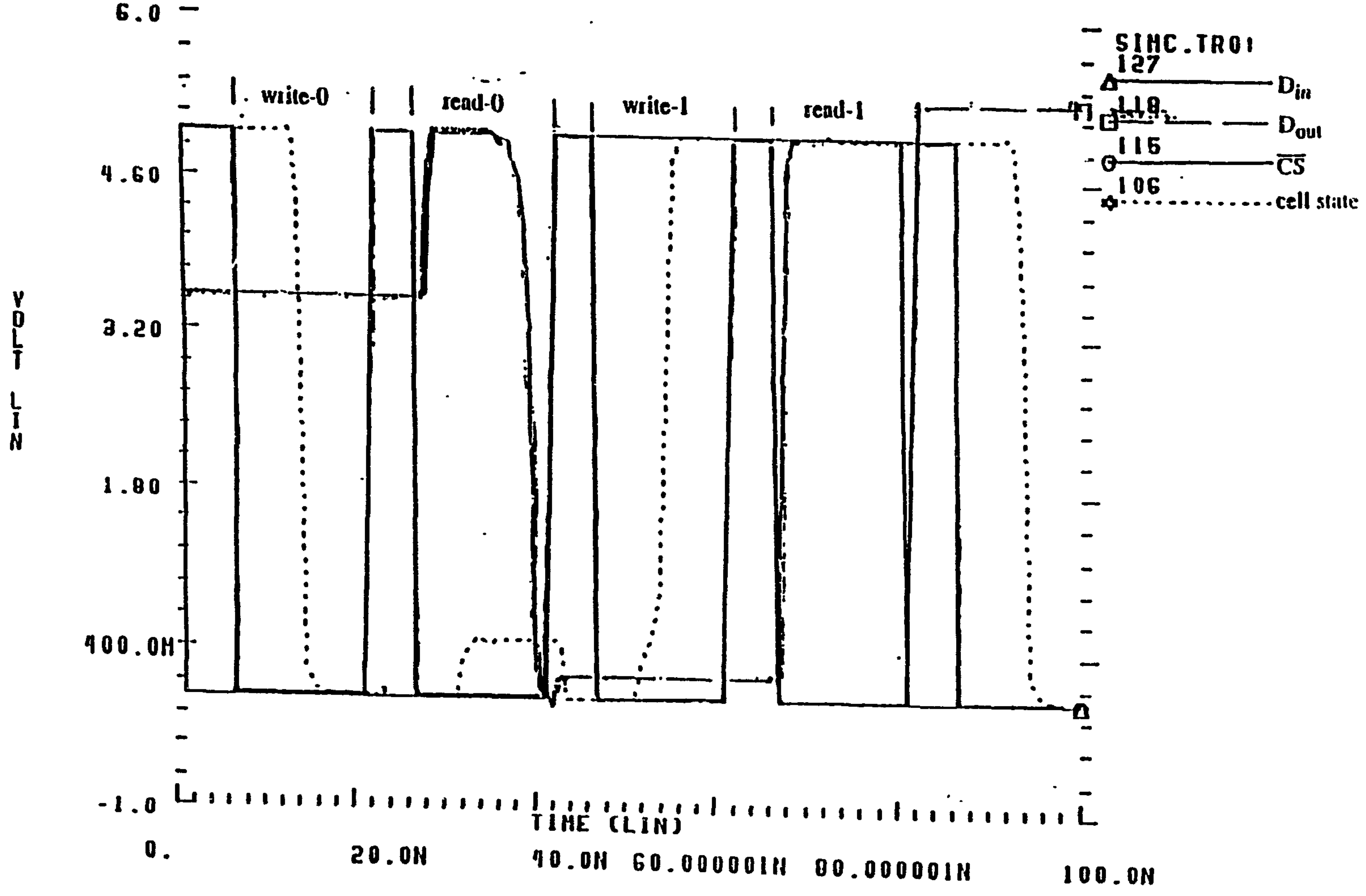


Fig. 6.2(b) HSPLOT for test circuit- alternate READ/WRITE operations.

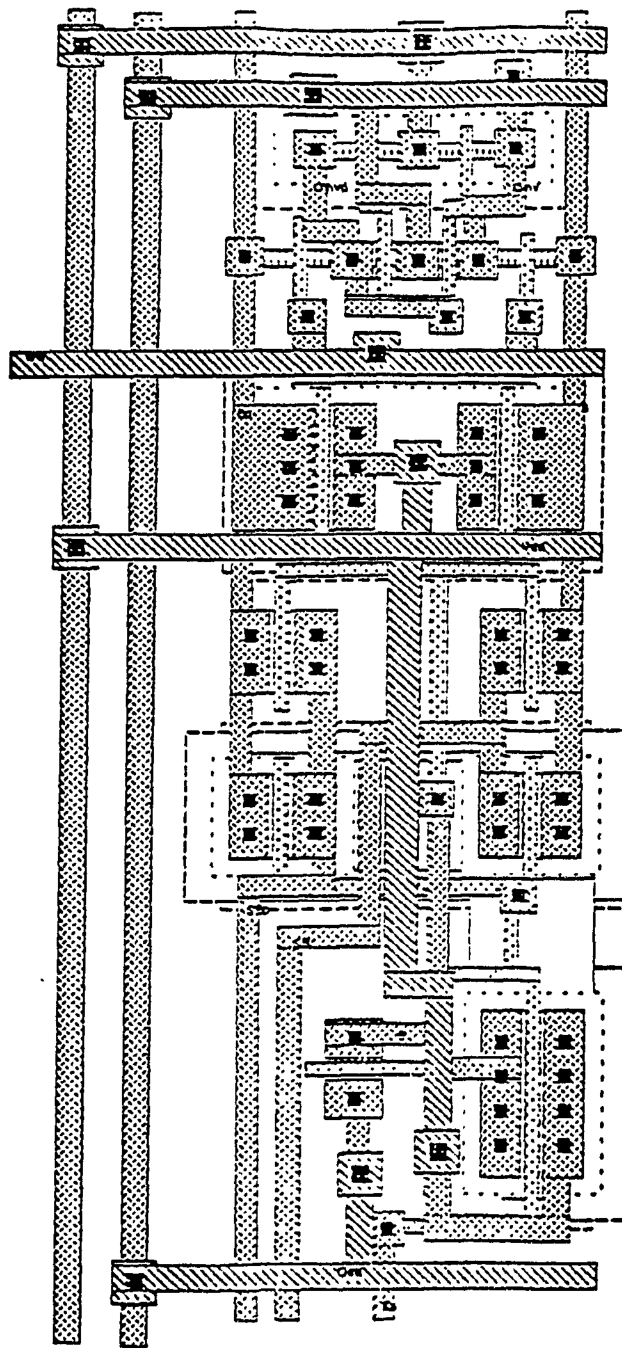


Fig. 6.3 A SRAM cell with precharge and column select circuit layout.

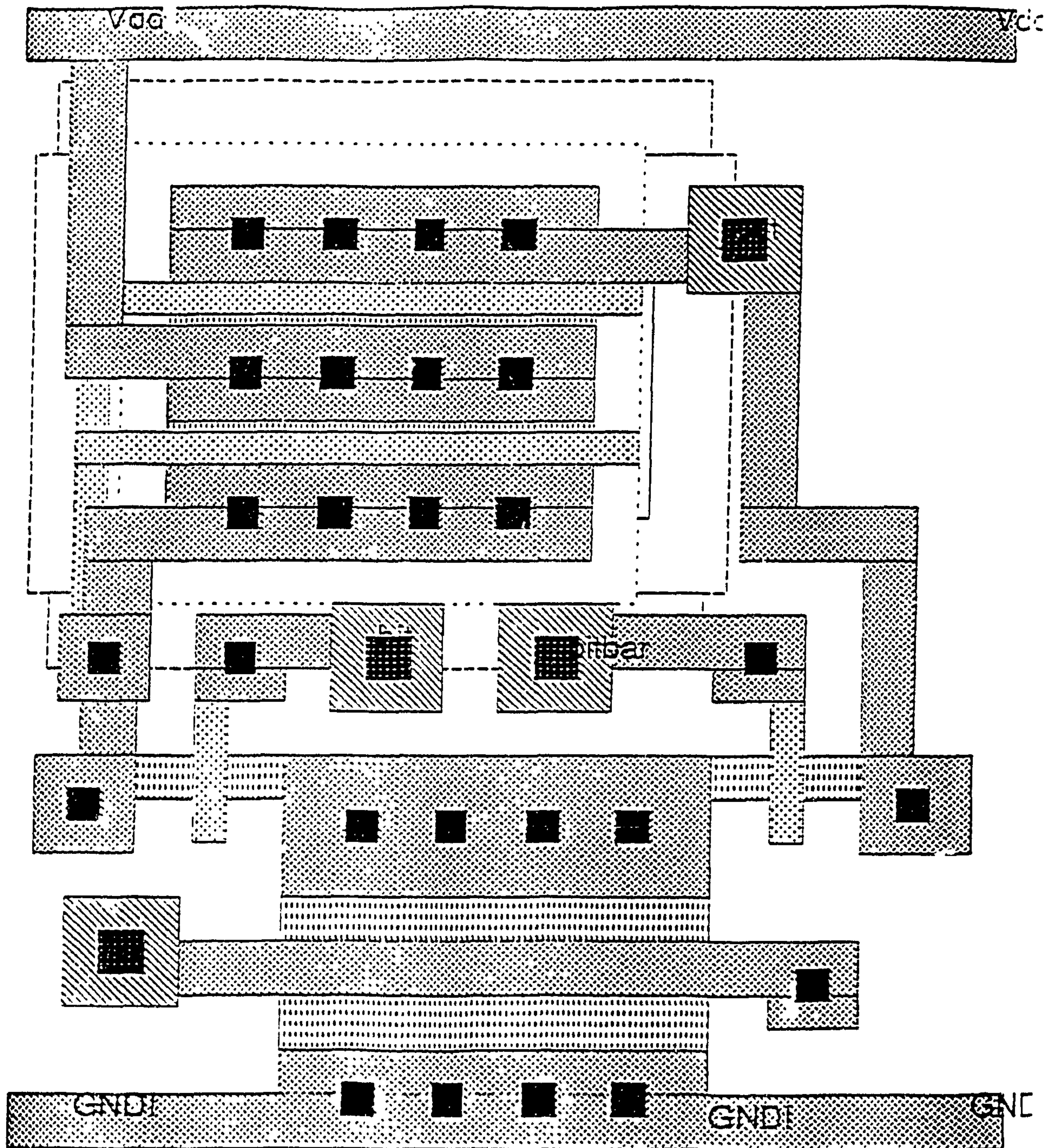


Fig. 6.4 A Current Mirror Sense Amplifier (CMSA) layout.

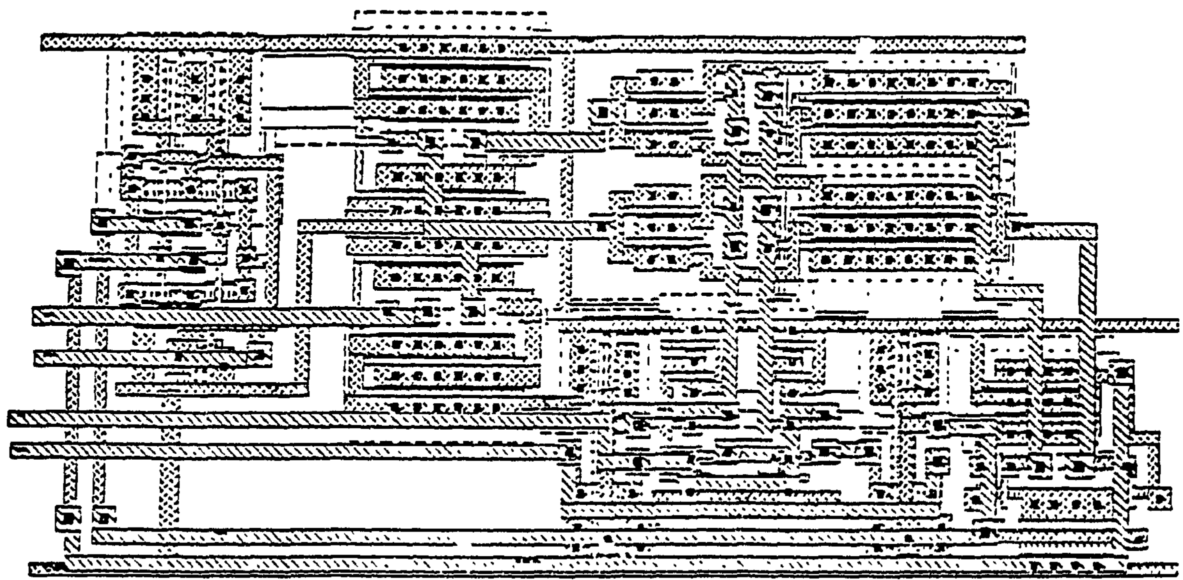


Fig. 6.5 I/O peripheral circuit layout of a SRAM.

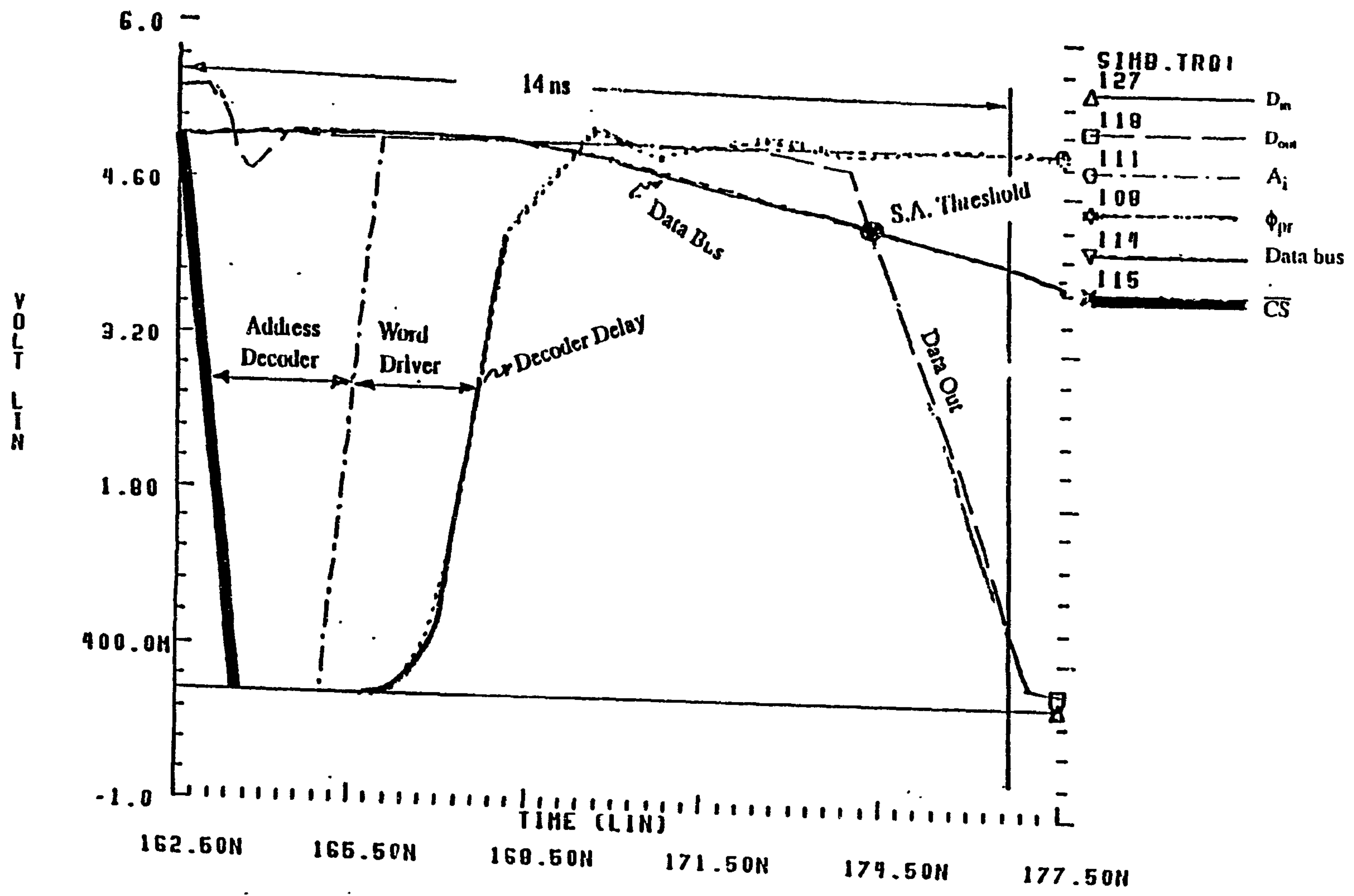


Fig. 6.6 READ-0 timing analysis of SRAM layout.

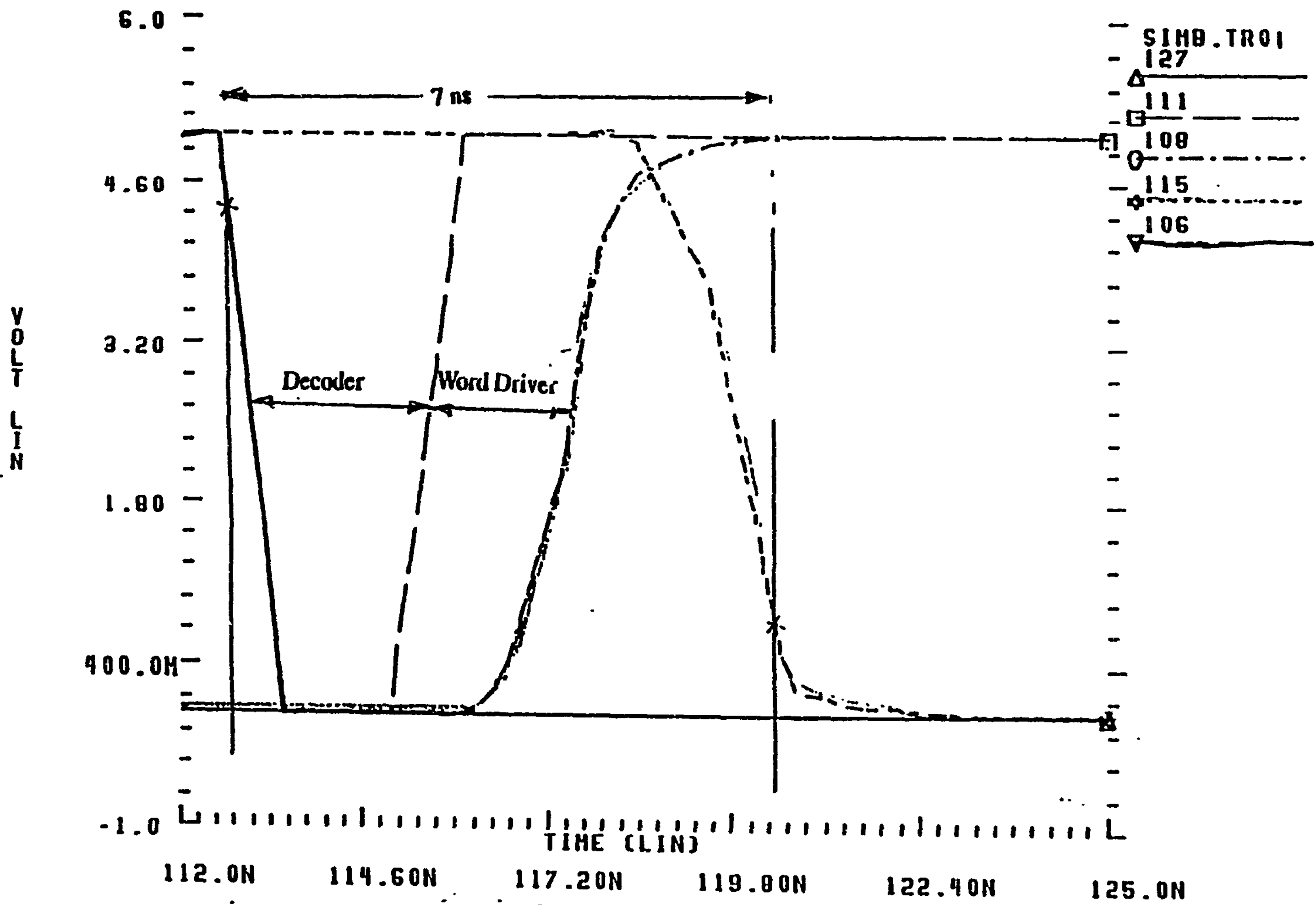


Fig. 6.7 WRITE-0 timing analysis of SRAM layout.

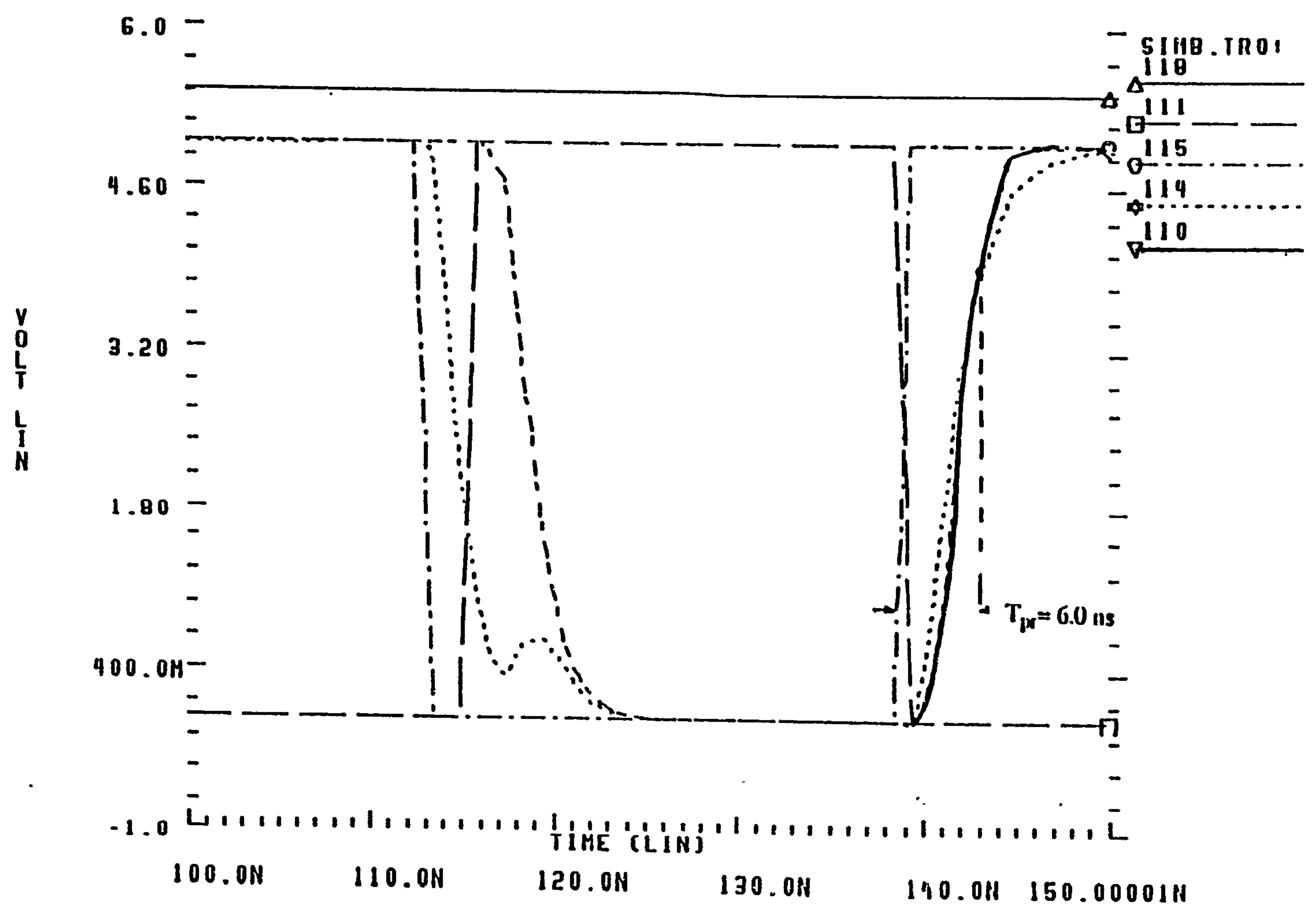


Fig. 6.8 An worst case precharge delay.

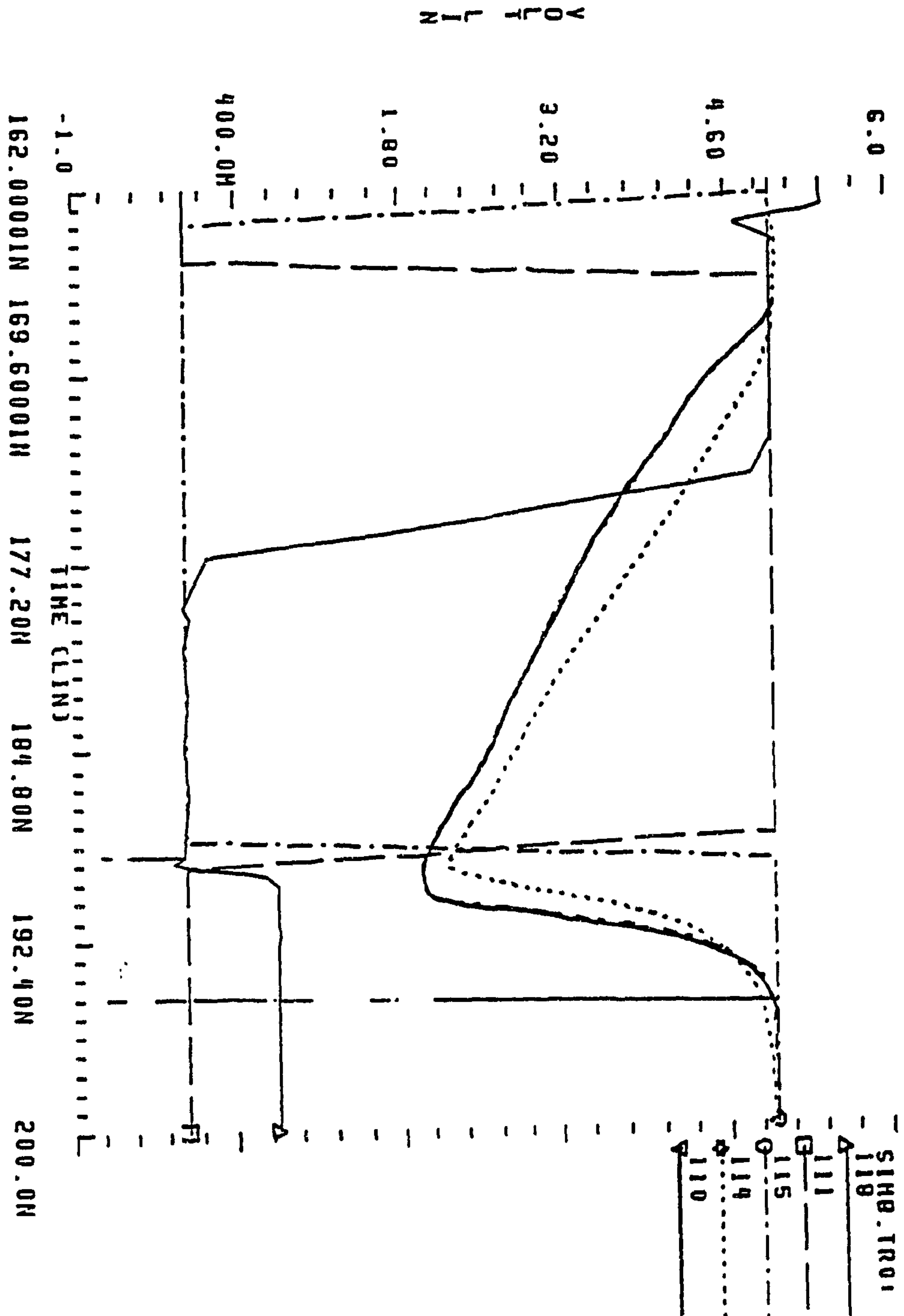


Fig. 6.9 A moderate case precharge delay.

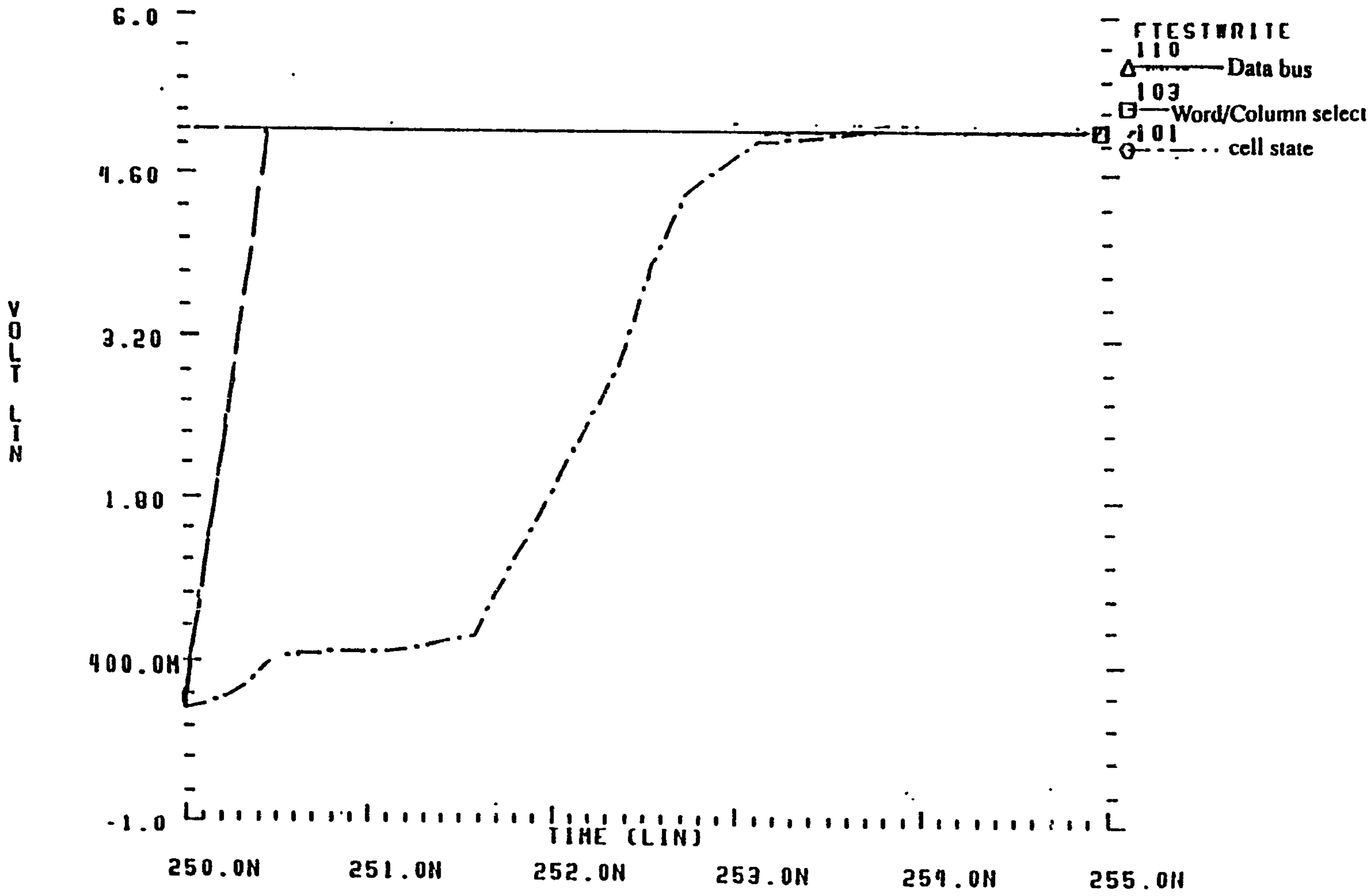


Fig. 6.10 A WRITE-1 timing analysis of the layout.

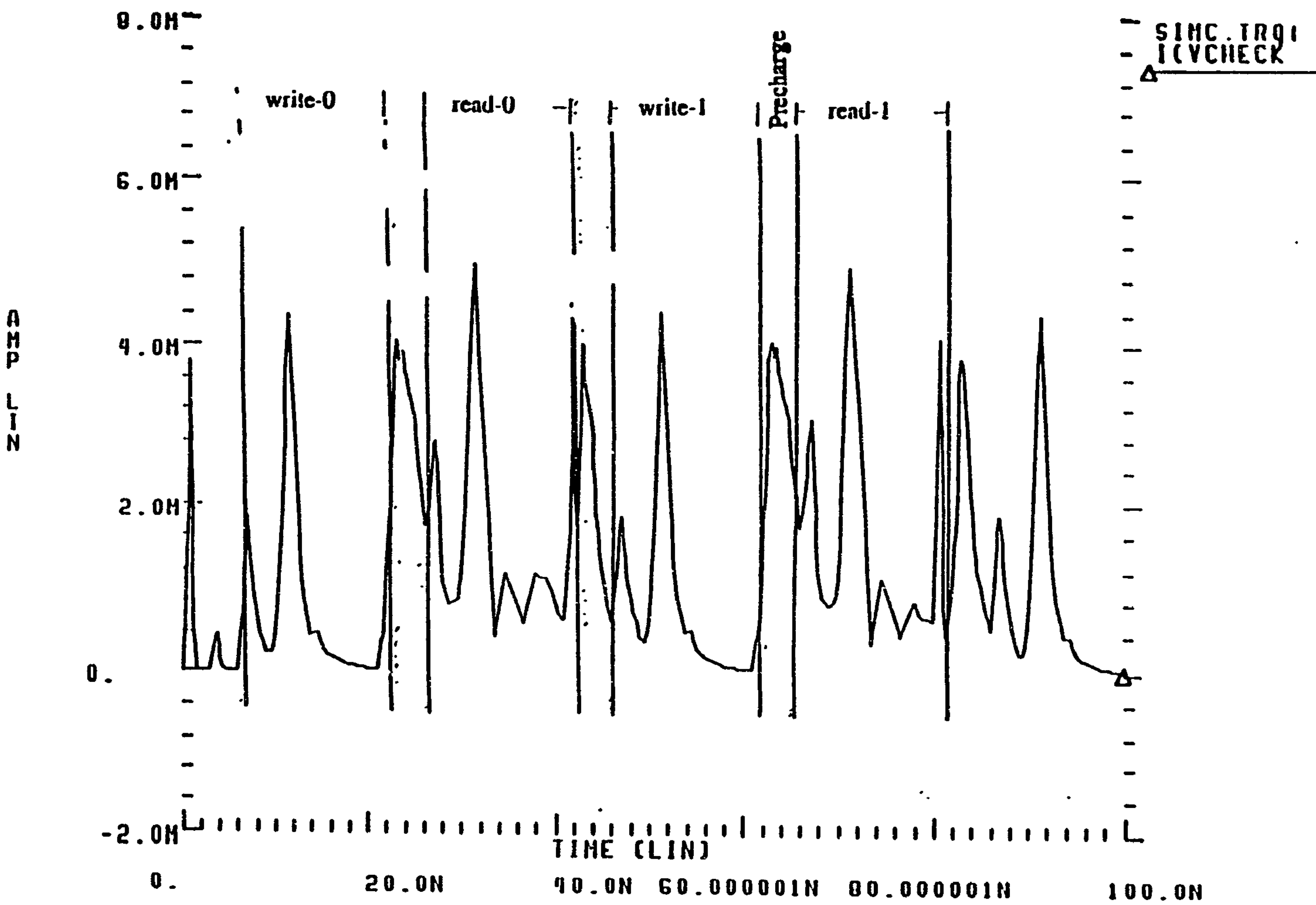


Fig. 6.11 Current consumption in different read/write cycle in SRAM layout.

Chapter 7
Conclusion

In this thesis, we presented a performance analysis and design of an optimized SRAM based on Elmore's RC delay model. Our approximate model is fitted with SPICE results for any SRAM array size to develop a generalized model. Optimum transistor sizes for different delay goals & various SRAM configurations have been attempted. The results confirm our analysis and SPICE simulation results as shown in Chapter-2. As given in Tables - 2, 3 & 4, it can be inferred that our optimization algorithm provides excellent results. A design aid tool for the optimized design of a SRAM is developed which is written in 'C'.

An extensive survey of different types of Decoding, Precharging, and Sensing techniques & circuits have been presented with their major advantages and disadvantages. A novel precharge technique called Power Down Y-Controlled PMOS (PDYCP) load precharge has been presented, which reduces a good amount of power consumption with little area overhead under any operating mode of SRAM.

Our optimization algorithm is able to handle variable user specifications of SRAM and can output the best trade-off. Since our analysis and algorithm gives the optimum sizes of different transistors in a SRAM, it might be an excellent guide to the designer who will be able to design an optimized SRAM in over night without loosing expensive time in SPICE simulation. The model can further be extended for dual & multi-port SRAMs. Based on our analysis and optimized parameters, a module generator for automatic generation of SRAM layouts can hereafter be developed.

REFERENCES

[1] S. Aizaki, T. Shimizu, M. Ohkawa, K. Abe, A. Aizaki, M. Ando, O. Kudoh, and I. Sasaki, "A 15-ns 4Mb CMOS SRAM, " IEEE JSSC , vol. 25, No. 5, pp.1063-1067, Oct . 1990.

[2] K. Sasaki, K. Ishibashi, K. Shimohigashi, T. Yamanaka, N. Moriwaki, M. Honjo, S. Ikeda, A. Koike, S. Meguro, and O. Minato, "A 23-ns 4Mb CMOS SRAM with 0.2 u A standby current," IEEE JSSC, Vol. 25, No. 5, pp.1075-1081, Oct . 1990.

[3] T. Hirose, S. Kawshima, H. Itoh, N. Suzuki,and T. Yabu, "A 20-ns 4Mb CMOS SRAM with Hierarchical Word Decoding Architecture ," IEEE JSSC , vol. 25, No. 5, pp. 1068-1074, Oct. 1990.

[4] H. Shimada, S. Hanamura, K. Ueda, T. Oono, O. Minato, and Y. Sakai,"A 18-ns 1-Mbit CMOS SRAM ," IEEE JSSC , vol. 23, No. 5, pp. 1073- 1077, Oct. 1988.

[5] K. Sasaki, S. Meguro, T. Masayoshi, T. Masuhara, M. Kubotara, and H. Toyoshima, "A 15-ns 1-Mbit CMOS SRAM ," IEEE JSSC , vol. 23, No. 5, pp.1067-1071, Oct . 1988.

[6] M. Matsui, T. Otani, J. Tsujimoto, H. Iwai, A. Suziki, K. Sato, M. Isobe, K. Hashimoto, M. Saitoh, H. Shibata, H. Sasaki, T. Matsuno, J. Matsunaga, and T. Iizuka, "A 25-ns 1-Mb CMOS SRAM with Loading -Free Bit lines," IEEE JSSC , vol. 22, No. 5, pp. 733-740, Oct. 1987.

[7] T. Wada, T. Hirose, H. Shinohara, Y. Kawai, K. Yuzuriha, Y. Kohno, and S. Kayano, "A 1-Mb CMOS SRAM Using Tripple Polysilicon I," IEEE JSSC , vol. 22, No. 5, pp. 727-732, Oct. 1987.

[8] W. C. H. Gubbels, C. D. Hartgring, R. H. W. Salters, J. A. M. Lammerts, M. J. Tooher, P. F. P. C. Hens, J. J. J. Bastiaens, J. M. F. Van Dijk, and M. A. Sprokel, "A 40-ns/100-pF Low-Power Full-CMOS 256K(32KX8) SRAM," IEEE JSSC , vol. 22, No. 5, pp. 741-747, Oct. 1987.

[9] Remi Cissou, and Remy Chapelle, "A High-Speed 64-kbit CMOS SRAM ," IEEE JSSC , vol. 21, No. 3, pp. 390-395,June 1986.

[10] T. Sakurai, J. Matsunaga, M. Isobe, T. Ohtani, K. Sawada, A. Aono, H. Nozawa, T. Iizuka, S. Kohyama, "A Low Power 256 kbit bit CMOS Static RAM with Dynamic Double Word Line," IEEE JSSC , vol. 19, No. 5, pp. 578-585, Feb. 1984..

[11] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano, "A Divided Word -Line Structure in the Static RAM and its Application to a 64K Full CMOS RAM ," IEEE JSSC , vol. 18, No. 5, pp. 479-485, Oct. 1983.

[12] K. Ochii, K. Hasimoto, H. Yasuda, M. Masuda, T. Kando, H. Nazawa, and S. Kohyama, "An Ultralow 8KX8-Bit Full CMOS SRAM with Six-Transistor Cell," IEEE JSSC , vol.-17, No.5, pp. 798-803, Oct. 1982.

[13] E. Seevinck, F. J. List, and J. Lohstroh, "Static-Noise Margin Analysis of MOS SRAM, " IEEE JSSC, vol.22, No.5, pp. 748-754, Oct. 1987.

[14] J. Lohstroh, E. Seevinck, and J. D. Groot, "Worst-Case Static Noise Margin

Criteria for Logic Circuits and Their Mathematical Equivalence," IEEE JSSC , vol.18, No.6, pp. 803-807, Oct. 1983.

[15] G. A. Sai-Halasz, M. R. Wordman, and R. H. Dennard, "Alpha-Particle -Induced Soft Error Rate in VLSI Circuits" IEEE JSSC , vol.17, No.2, pp. 355-361, April 1982.

[16] K. Anami, M. Yoshimoto, H. Shinohara, Y. Hirata, and T. Nakano, "Design Consideration of a Static Memory Cell," IEEE JSSC, vol. 18, No. 4, pp. 414-418, Aug. 1983.

[17] N. Hedenstierna and K.O. Jeppson, "CMOS circuit speed and buffer optimization," IEEE Trans. on CAD, Vol.6, No. 2, pp.-270-281, March 1987.

[18] J. Rubinstein, P. Penfield, J. R. and M. A. Horowitz, et al. , "Signal Delay in RC Tree Networks," IEEE Trans. on CAD, vol. 2, pp.202-210, July 1983.

[19] Maciej J. C., "Layer Assignment for VLSI Interconnect Delay minimization," IEEE Trans. on CAD, vol. 8, No. 6, pp.701-707, June 1989.

[20] T. M. lin, and C. A. Mead, "Signal Delay in General RC Networks," IEEE Trans. on CAD, vol. 3, pp. 331-349, Oct. 1984. 9

[21] A. J. Al-Khalili, Y. Zhu, D. Al-Khalili, "A module Generator for Optimized CMOS buffers," IEEE Trans. on CAD, vol. 9, No. 10, pp. 1028-1046, Oct. 1990.

[22] J. P. Fishburn,and A. E. Dunlop, "A Posynomial Programming Approach to transistor sizing," IEEE ICCAD, pp. 326-328, 1985.

[23] N. Weste, and K. Eshraghian, "Principles of CMOS VLSI Design," Addision-wesley, 1985.

[24] M. Annaratone, "Digital CMOS Circuit Design," Kluwer Academic Pub., 1986.

[25] Lance Glasser, and D. W. Dobberpuhl, "The Design and Analysis of VLSI Circuits," Addision-wesly, 1988.

[26] B. Hoppe, N. Gerd, S. Doris and S. Will, "Optimization of High-speed CMOS Logic Circuits with Analytical Models for Signal Delay, Chip Area, and Dynamic Power Dissipation,", IEEE Trans. on CAD, vol. 9, No. 3, pp. 237-247, March 1990.

[27] Lynnee M. B., Steven P.M., J. Allen, "Macromodeling CMOS circuits for Timing Simulation," IEEE Trans. on CAD, Vol. 7, No. 12, pp. 1237-1247, December 1988.

[28] H. Shimada, Y. Tange, K. Tanimoto, M. Shiraishi, N. Suzuki, and T. Namura, "A 46ns 1-Mbit CMOS SRAM," IEEE JSSC,vol.23, No. 1, pp-53-58, Feb. 1988.

[29] Hiroaki Okuyama, T. Nakano, S. Nishida, E. Aono, H. Satoh, and S. Arita, "A 7.5ns 32kX8 CMOS SRAM," IEEE JSSC, vol. 23,No. 5, pp-1055-1059, Oct. 1988.

[30] D.A. Hodges and H.G. Jackson, "Analysis and Design of Digital Integrated Circuits," New York, McGraw-Hill, 1983.

[31] Robert K.B., Gary D. H., Alberto L. S., "A Survey of optimization techniques for Integrated-Circuit Design", Proceedings of the IEEE, Vol. 69, No. 10, pp. 1334-1364, Oct. 1981.

APPENDIX

Appendix A

SRAM Modeling

In this appendix the basic resistance and capacitance equations for a MOS transistor will be derived, which will further be used for the R, C modeling of a SRAM.

A.1 Capacitance and Resistance calculation of a MOS Transistor

Gate Capacitance Estimation:

Fig. A.1.1 shows the model of the parasitic capacitance of a MOS transistor. The gate capacitance C_g of a MOS transistor can be written as,

$$C_{gn/p} = C_{gbn/p} + C_{gsn/p} + C_{gdn/p} \,. \qquad (A.1.1)$$

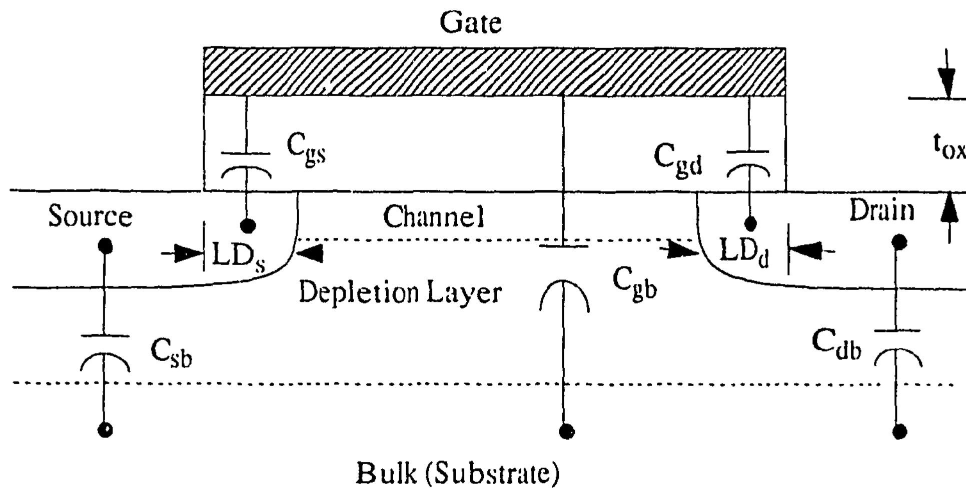


Fig. A.1.1 Parasitic Capacitance of a MOS Transistor

Where $C_{gbn/p}$ is the gate-bulk/substrate capacitance of n or p channel transistor.

$C_{gsn/p}$ is the gate-source capacitance of n or p channel transistor.

$C_{gdn/p}$ is the gate-drain capacitance of the n or p channel transistor.

The parameters in equation A.1.1 can be represented in terms of devices size and SPICE parameters as follows:

$$C_{gbn/p} = C_{oxn/p} W_{n/p} L_{n/p}.$$

$$C_{gsn/p} = C_{oxn/p} W_{n/p} LD_{sn/p}$$

$$C_{dn/p} = C_{oxn/p} W_{n/p} LD_{dn/p} \qquad (A.1.2)$$

Where $C_{oxn/p}$ is the gate oxide capacitance of the n or p channel transistor.

$W_{n/p}$ and $L_{n/p}$ are the Channel width and length.

$LD_{sn/p}$ is the lateral diffusion of the gate-source overlap.

$LD_{dn/p}$ is the lateral diffusion of the gate-drain overlap.

Diffusion Capacitance Estimation:

The area and peripheral components of diffusion capacitance are shown in fig. A.1.2. The diffusion capacitance of a MOS transistor is a function of the "base" area and "sidewall" periphery [23].

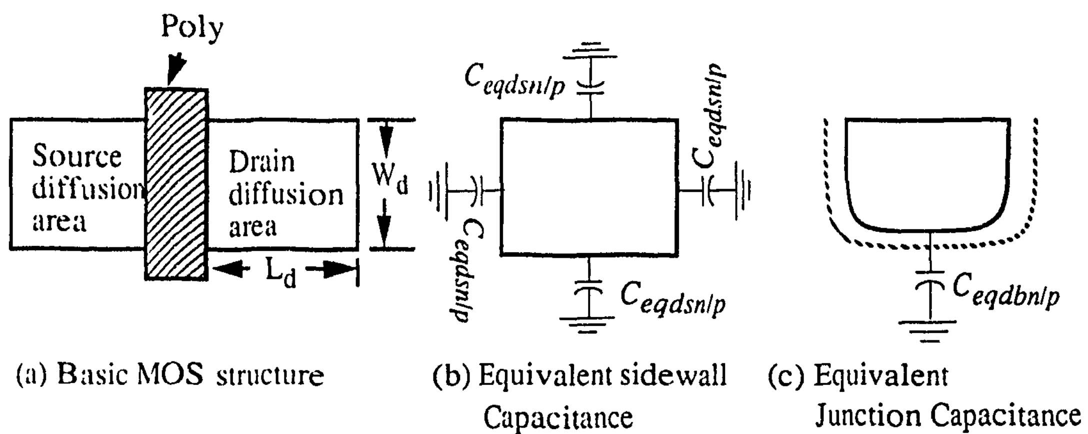


Fig. A.1.2 Area and peripheral components of diffusion capacitance.

Now, C_{db}, the drain-bulk capacitance of an n or p MOS transistor can be determined by,

$$C_{dbn/p} = C_{eqdbn/p} Ad_{n/p} + C_{eqdsn/p} Pd_{n/p} . \quad (A.1.3)$$

Also, the source-bulk capacitance can be determined by,

$$C_{sbn/p} = C_{eqsbn/p} As_{n/p} + C_{eqssn/p} Ps_{n/p}. \quad (A.1.4)$$

Where $C_{eqdbn/p}$ and $C_{eqsbn/p}$ are the n or pMOS transistor's bottom equivalent junction capacitance of drain and source diffusion per unit area respectively.

$C_{eqdsn/p}$ and $C_{eqssn/p}$ are the n or pMOS transistor's sidewall equivalent junction capacitance of drain and source diffusion region per unit length of junction perimeter respectively.

$Ad_{n/p}$ and $As_{n/p}$ are the area of drain and source diffusion regions respectively.

$Pd_{n/p}$ and $Ps_{n/p}$ are the perimeter of the drain and source diffusion area respectively.

K_{eq}, a dimensionless constant is used to relate $C_{eq\sim}$ to $CJ_{n/p}$ (zero bias junction bottom capacitance per unit area) and $CJSW_{n/p}$ (zero bias junction sidewall capacitance per unit length) [30]. Therefore, we can determine the following:

$$C_{eqdbn/p} = K_{eq} CJ_{n/p}$$

$$C_{eqdsn/p} = K_{eq} CJSW_{n/p} \quad (A.1.5)$$

Also, assume $C_{eqsbn/p} = C_{eqdbn/p}$ and $C_{eqssn/p} = C_{eqdsn/p}$.

According to fig. A.1.2, the area and perimeter of the drain diffusion region can be determined by,

$$Ad_{n/p} = W_{dn/p} L_{dn/p}$$

$$Pd_{n/p} = 2(W_{dn/p} + L_{dn/p}) \quad (A.1.6)$$

Assume $As_{n/p} = Ad_{n/p}$ and $Ps_{n/p} = Pd_{n/p}$.

Where $W_{dn/p}$ and $L_{dn/p}$ are the drain width and length respectively of the n or pMOS transistor.

Resistance Estimation:

The channel resistance of an n or pMOS transistor can be represented by,

$$R_{n/p} = \frac{K_r}{\beta_{n/p}(V_{gs} - V_t)} \tag{A.1.7}$$

Where K_r is a constant multiplication factor for resistance.

$\beta_{n/p}$ is the transistor gain factor.

V_{gs} is the gate to source voltage.

V_t is the threshold voltage.

$\beta_{n/p}$ can be determined by,.

$$\beta_{n/p} = \frac{\mu_{n/p}\varepsilon}{t_{ox}}\left(\frac{w_{n/p}}{L_{n/p}}\right) \tag{A.1.8}$$

From A.1.7 and A.1.8 we can write,

$$R_{n/p} = \frac{\rho_{n/p}}{W_{n/p}} \tag{A.1.9}$$

Where,

$$\rho_{n/p} = \frac{K_r t_{ox} L_{n/p}}{\mu_{n/p}\varepsilon(V_{gs} - V_t)} \tag{A.1.10}$$

Any transmission gate (TG) equivalent resistance can be defined as,

$$R_{eqtn/p\sim} = R_n \,//\, R_p. \tag{A.1.11}$$

A.2 SRAM Node Capacitances

In this section various node capacitances for Fig. 1.3(b), which are used for SRAM modeling and analysis in chapter- 2 are derived.

SRAM cell storage node capacitance:

$$C_4 = C_5 = C_{dbna} + C_{dbcp} + C_{dbcnd} + C_{gcp} + C_{gnd}. \quad (A.2.1)$$

Other node capacitances (refer to chapter -2):

$$C_{03} = C_{06} = C_{dbprp} + C_{dbcsppas} + C_{dbcsnpas}. \quad (A.2.2)$$

$$C_{14} = C_{gps1} + C_{gps2} + C_{dbps1} + C_{dbns1}. \quad (A.2.3)$$

$$C_{15} = C_{dbps2} + C_{dbns2} + C_{gobp2} + C_{gobn2}. \quad (A.2.4)$$

$$C_{16} = C_{dbsc} + C_{sbns1} + C_{sbns2}. \quad (A.2.5)$$

$$C_{17} = C_{dbpcs} + C_{dbncs} + C_{dbnrw} + C_{gredp} + C_{gredn} + 2\,C_{gwnpas}. \quad (A.2.6)$$

$$C_{170} = C_{dbpred} + C_{dbnred} + 2C_{gwppas}. \quad (A.2.7)$$

$$C_{171} = C_{dbobp1} + C_{dbobn1} + C_{gobp4}. \quad (A.2.8)$$

$$C_{150} = C_{dbobp2} + C_{dbobn2} + C_{gobp3} + C_{gobn3}. \quad (A.2.9)$$

$$C_{200} = C_{sbobn4} + C_{dbobn3}. \quad (A.2.10)$$

$$C_{201} = C_{sbobp4} + C_{dbobp3}. \quad (A.2.11)$$

$$C_{20} = C_{dbobp4} + C_{dbobn4}. \quad (A.2.12)$$

$$C_{231} = C_{dbrwdn} + C_{dbrwdp} + C_{grwp} + C_{grwn}. \quad (A.2.13)$$

$$C_{161} = C_{dbcsp} + C_{dbcsn} + C_{dbrwp} + C_{gsc} + C_{gobp1} + C_{gobn1}. \quad (A.2.14)$$

$$C_{21} = C_{dbpwb1} + C_{dbnwb1} + C_{sbwnpas} + C_{sbwppas} + C_{gpwb2} + C_{gnwb2} \quad (A.2.15)$$

$$C_{120} = C_{dbpwb2} + C_{dbnwb2} + C_{sbwnpas} + C_{sbwppas} \quad (A.2.16)$$

$$C_{12} = C_{dbwnpas} + C_{dbwppas} + C_{gns} + C_{sbcsnpas} + C_{sbcsppas} + n\,C_{dbus} \quad (A.2.17)$$

$$C_{241} = C_{dbrwp} + C_{sbcsp}. \quad (A.2.18$$

Appendix B

Optimized SRAM Design

In this section of the appendix the optimized design of three important circuits in a SRAM such as word driver, precharge and sense amplifier circuits will be presented based on the optimization criteria described in section 5.3.

B.1.1 Optimized Word Driver Design

Case-1: Delay without area/power constraint

In this case we put equation (5.15) into equation (5.6) which results in,

$$\frac{dD}{dW_p} = b_1\alpha_1 R_{nin} + b_2\frac{\rho_p}{W_p}\alpha_2 - b_2\frac{\rho_p}{W_p^2}(\alpha_2 W_p + \alpha_3) - b_3 n\frac{\rho_p}{W_p^2}C_w = 0 \quad \text{(B. 1)}$$

or,

$$W_p^2(b_1\alpha_1 R_{nin}) - b_2\rho_p\alpha_3 - b_3 n\rho_p C_w = 0 \quad \text{(B. 2)}$$

or,

$$W_p^2(b_1\alpha_1 R_{nin}) = b_2\rho_p\alpha_3 + b_3 n\rho_p C_w \quad \text{(B. 3)}$$

For minimum delay assume $W_p = W_{ptm}$. Therefore,

$$W_{ptm} = \sqrt{\frac{b_2\rho_p\alpha_3 + b_3 n\rho_p C_w}{b_1\alpha_1 R_{nin}}} \quad \text{(B. 4)}$$

Case- 2: Delay with area/power constraint

We have from equation (5.9),

$$\frac{dD}{dW_p} + \lambda_w\frac{dA}{dW_p} = 0 \quad \text{(B. 5)}$$

Putting equation (5.15) and (5.19) into eqn. (B. 5) which results in,

$$b_1\alpha_1 R_{nin} + b_2\frac{\rho_p}{W_p}\alpha_2 - b_2\frac{\rho_p}{W_p^2}(\alpha_2 W_p + \alpha_3) - b_3 n\frac{\rho_p}{W_p^2}C_w + \lambda_w\alpha_4 = 0 \qquad \text{(B. 6)}$$

or,

$$W_p^2(b_1\alpha_1 R_{nin} + \lambda_w\alpha_4) - b_2\rho_p\alpha_3 - b_3 n\rho_p C_w = 0 \qquad \text{(B. 7)}$$

For optimum delay assume $W_p = W_{po}$. Therefore,

$$W_{po} = \sqrt{\frac{b_2\rho_p\alpha_3 + b_3 n\rho_p C_w}{b_1\alpha_1 R_{nin} + \lambda_w\alpha_4}} \qquad \text{(B. 8)}$$

The above equations (B. 4) and (B. 8) are used in section 5.4.1 to design an optimized word driver for SRAM.

B.1.2 Optimized Precharge Circuit Design

Case-1: Delay without area/power constraint

Putting equation (5.25) and (5.26) into eqn. (5.27) we have,

$$-b_{p2}\frac{\rho_{pr}}{W_{pr}^2}((2CJSW_p)L_d + mC_b + 2C_{dbcsp}) + 2b_{p1}(R_{tn} + R_{tn1})\delta_2 C_{oxp}L_p = 0 \qquad \text{(B. 9)}$$

For minimum delay assume $W_{pr} = W_{prm}$ and we have from (B. 9),

$$W_{prm} = \sqrt{\frac{b_{p2}\rho_{pr}(CJSW_p L_d + \frac{m}{2}C_b + C_{dbcsp})}{b_{p1}\delta_2 C_{oxp}L(R_{tn} + R_{tn1})}} \qquad \text{(B.10)}$$

Case-2: Delay with area/power constraint

Putting equation (5.27) and (3.19) into eqn. (5. 9) we have,

$$-b_{p2}\frac{\rho_{pr}}{W_{pr}^2}((2CJSW_p)L_d + mC_b + 2C_{dbcsp}) + 2b_{p1}(R_{tn} + R_{tn1})\delta_2 C_{oxp}L_p + \lambda_{pr}\delta_3 = 0 \qquad \text{(B. 11)}$$

For optimum precharge delay assume $W_{pr} = W_{pro}$ and we have from (B. 11),

$$W_{pro} = \sqrt{\frac{b_{p2}\rho_{pr}(2CJSW_pL_d + mC_b + 2C_{dbcsp})}{2b_{p1}\delta_2C_{oxp}L(R_{tn} + R_{tn1}) + \lambda_{pr}\delta_3}} \qquad \text{(B. 12)}$$

The above equations (B. 10) and (B. 12) are used in section 5.4.2 to design an optimized precharge circuit for a SRAM.

B.1.3 Optimized Sense Amplifier Circuit Design

We have equation (5.32) for the case of read-0 as,

$$T_{sense} = T_{sk} + b_{s1}T_{bus} + b_{s2}T_{s0} + b_{s3}T_{ssel} \qquad \text{(B. 13)}$$

T_{bus} is given by equation (2.35) which can be written as,

$$T_{bus} = \frac{n(n+1)}{2}R_{bus}C_{bus} + nR_{bus}(2C_{dbwpp} + C_{gns}) \qquad \text{(B. 14)}$$

or,

$$T_{bus} = T_{bus1} + nR_{bus}C_{gns} \qquad \text{(B. 15)}$$

Where,

$$T_{bus1} = \frac{n(n+1)}{2}R_{bus}C_{bus} + 2nR_{bus}C_{dbwpp} \qquad \text{(B. 16)}$$

Eqn. (B. 15) can further be expanded as,

$$T_{bus} = T_{bus1} + nR_{bus}\delta_1C_{oxn}L_nW_{sn} \qquad \text{(B. 17)}$$

Where,

$$C_{gns} = \delta_1C_{oxn}L_nW_{sn} \qquad \text{(B. 18)}$$

T_{s0} is given by equation (2.43) which can be rewritten as,

$$T_{s0} = (R_{nsc} + R_{ns})\, C_{15} + R_{nsc}\, C_{16} \qquad \text{(B. 19)}$$

Where C_{15} and C_{16} are defined by eqn. (A.2.4) and (A.2.5) in appendix A.2.

Eqn. (B. 19) can further be expanded as follows:

$$T_{s0} = (R_{nsc} + R_{ns})\,(C_{dbps2} + C_{dbns2} + C_{gobp2} + C_{gobn2}) + R_{nsc}(C_{dbsc} + C_{sbns1} + C_{sbns2})$$

Assume $C_{sbns1} = C_{sbns2}$ and equating the equivalent values of R's and C's above we have,

$$T_{s0} = \left(\frac{\rho_n}{W_{snc}} + \frac{\rho_n}{W_{sn}}\right)(K_{eq}CJ_pW_{sp}L_d + 2K_{eq}CJSW_p\,(W_{sp} + L_d) +$$

$$K_{eq}CJ_nW_{sn}L_d + 2K_{eq}CJSW_n\,(W_{sn} + L_d) + C_{gobp2} + C_{gobn2}\,) +$$

$$\frac{\rho_n}{W_{snc}}\,[K_{eq}CJ_nW_{snc}L_d + 2K_{eq}CJSW_n\,(W_{snc} + L_d) +$$

$$2K_{eq}CJ_nW_{sn}L_d + 4K_{eq}CJSW_n\,(W_{sn} + L_d)\ ] \qquad \text{(B. 20)}$$

We also assume, $W_{sn} = W_{sp}$, and $W_{snc} = 2\,W_{sn}$, then equation (B.20) becomes,

$$T_{s0} = \left(\frac{3\rho_n}{2W_{sn}}\right)(K_{eq}CJ_pW_{sn}L_d + 2K_{eq}CJSW_p\,(W_{sn} + L_d) +$$

$$K_{eq}CJ_nW_{sn}L_d + 2K_{eq}CJSW_n\,(W_{sn} + L_d) + C_{gobp2} + C_{gobn2}\,) +$$

$$\frac{\rho_n}{2W_{sn}}\,[4K_{eq}CJ_nW_{sn}L_d + 2K_{eq}CJSW_n\,(4W_{sn} + 3L_d)\ ] \qquad \text{(B. 21)}$$

or,

$$T_{s0} = (\frac{3\rho_n}{2})\left[K_{eq}CJ_pL_d + 2K_{eq}CJSW_p\left(1+\frac{L_d}{W_{sr}}\right)+\right.$$

$$K_{eq}CJ_nL_d + 2K_{eq}CJSW_n\left(1+\frac{L_d}{W_{sn}}\right)+\frac{1}{W_{sn}}(C_{gobp2}+C_{gobn2})\ ] +$$

$$\frac{\rho_n}{2}\left[4K_{eq}CJ_nL_d + 2K_{eq}CJSW_n\left(4+\frac{3L_d}{W_{sn}}\right)\right] \qquad \text{(B. 22)}$$

T_{ssel} is given by eqn. (2.41) as,

$$T_{ssel}=R_{wcn}\,C_{231} + (R_{rwp}+R_{csp})\,C_{161}. \qquad \text{(B. 23)}$$

Where C_{231} and C_{161} are given eqn. (A. 2.13) and (A. 2.14) respectively in appendix A. 2.

Eqn. (B. 23) can further be expanded as,

$$T_{ssel}= R_{wcn}\,C_{231} + (R_{rwp} + R_{csp})\,(C_{dbcsp}+C_{dbcsn} +$$

$$C_{dbrwp} + C_{gsc} + C_{gobp1} + C_{gobn1}) \qquad \text{(B. 24)}$$

Assume, $W_{redp}= W_{csp}$, $R_{rwp}= R_{csp}$,

and $C_s= C_{dbcsp}+C_{dbcsn} + C_{dbrwp} + C_{gobp1} + C_{gobn1})$

Then eqn. (B. 24) becomes,

$$T_{ssel}= R_{wcn}\,C_{231} + 2\,R_{rwp}\,(C_s + C_{gsc}) \qquad \text{(B. 25)}$$

We have, $C_{gsc}= \delta_1\,C_{oxn}\,W_{snc}$ and $W_{snc}= 2\,W_{sn}$. Then eqn. (B. 25) can be expanded as,

$$T_{ssel}= R_{wcn}\,C_{231} + 2\,R_{rwp}\,(C_s +2\,\delta_1\,C_{oxn}\,W_{sn}\,L_n). \qquad \text{(B. 26)}$$

Case- 1: Delay without area/power constraint

Putting eqn. (B. 17), (B. 22) and (B. 26) into (B. 13) and applying eqn. (5.6) on it for the minimum delay condition, we have,

$$\frac{dT_{sense}}{dW_{sn}} = b_{s1}(nR_{bus}\delta_1 C_{oxn}L_n)$$

$$-b_{s2}\left[\frac{3\rho_n}{2W_{sn}^2}(2K_{eq}CJSW_pL_d + 2K_{eq}CJSW_nL_d + C_{gobp2} + C_{gonn2}) + \right.$$

$$\left.\frac{3\rho_n}{W_{sn}^2}K_{eq}CJSW_nL_d\right] + 4b_{s3}R_{rwp}\delta_1 C_{oxn}L_n = 0 \qquad \text{(B. 27)}$$

Assume, $L = L_n = L_p$ and $C_{ox} = C_{oxn} = C_{oxp}$ and (B.27) becomes,

$$W_{sn}^2(b_{s1}\, nR_{bus}\delta_1 C_{ox}L + 4b_{s3}R_{rwp}\delta_1 C_{ox}L) =$$

$$b_{s2}[\,(3\rho_n)\,(K_{eq}CJSW_pL_d + K_{eq}CJSW_nL_d)$$

$$+ 3\rho_n K_{eq}CJSW_nL_d + \frac{3}{2}(C_{gobp2} + C_{gonn2})\,] \qquad \text{(B. 28)}$$

Putting $W_{sn} = W_{snm}$ for minimum sensing delay and we have from eqn. (B. 26),

$$W_{snm} = \sqrt{\frac{b_{s2}\rho_n(6K_{eq}CJSW_nL_d + 3K_{eq}CJSW_pL_d + \frac{3}{2}(C_{gobn2} + C_{gobp2}))}{\delta_1 C_{ox}L(b_{s1}nR_{bus} + 4b_{s3}R_{rwp})}} \qquad \text{(B. 29)}$$

Case-2: Delay with area/power constraint

Putting eqn. (B. 17), (B. 22) and (B. 26) into (B. 13) then using eqn. (B. 13), (3.15) and applying eqn. (5.9) for optimum delay condition, we have,

$$\frac{dT_{sense}}{dW_{sn}} + \lambda_s \frac{dA_{sa}}{dW_{sn}} = b_{s1}\,(nR_{bus}\delta_1 C_{oxn} L_n)$$

$$-b_{s2}\left[\frac{3\rho_n}{2W_{sn}^2}\,(2K_{eq}CJSW_pL_d + 2K_{eq}CJSW_nL_d + C_{gobp2} + C_{gonn2}) + \right.$$

$$\left.\frac{3\rho_n}{W_{sn}^2}K_{eq}CJSW_nL_d\right] + 4b_{s3}R_{rwp}\delta_1 C_{oxn}L_n + \delta_4\lambda_s = 0 \qquad \text{(B. 30)}$$

Assume, $L = L_n = L_p$ and $C_{ox} = C_{oxn} = C_{oxp}$ and (B. 30) becomes,

$$W_{sn}^2\,(b_{s1}\,nR_{bus}\delta_1 C_{ox}L + 4b_{s3}R_{rwp}\delta_1 C_{ox}L + \delta_4\lambda_s) =$$

$$b_{s2}\,[\,(3\rho_n)\,(K_{eq}CJSW_pL_d + K_{eq}CJSW_nL_d)$$

$$+ 3\rho_n K_{eq}CJSW_nL_d + \frac{3}{2}\,(C_{gobp2} + C_{gonn2})\,\Big] \qquad \text{(B. 31)}$$

Putting $W_{sn} = W_{sno}$ for optimum sensing delay and we have from (B. 31) as,

$$W_{sno} = \sqrt{\frac{b_{s2}\rho_n\,(6K_{eq}CJSW_nL_d + 3K_{eq}CJSW_pL_d + \frac{3}{2}\,(C_{gobn2} + C_{gobp2}))}{\delta_1 C_{ox}L\,(b_{s1}nR_{bus} + 4b_{s3}R_{rwp}) + \lambda_s\delta_4}} \qquad \text{(B. 32)}$$

The above equations (B. 29) and (B. 32) are used in section 5.4.3 to design an optimized sense amplifier for a SRAM.

B.2 Regression Results for SRAM Modeling

In this section the values obtained for fit constants by regression analysis between the analytical delay models and SPICE simulation are given. The word and bit line capacitances increase as the SRAM array size is increased. In our design the basic SRAM cell size is assumed to be fixed. Therefore, as the number of cells per row or column changes the other peripheral circuit design needs to be changed to have an optimized design.

In our analysis we classified the SRAM array (*m x n*) into three different groups such as, Small Scale: (m < 16) & (n < 16), Medium Scale: 16 <= (m & n) <=64, Large Scale: (m > 64) & (n <= 64), where m and n are the number of rows and columns respectively.

The regression fit constants for the word line delay estimation are given by equation (5.16).

The values obtained for the polynomials of equation (5.16) are as follows:

Fit constants	Small Scale	Medium Scale	Large Scale
a_{11}	0.0	1.639	2.2787
a_{12}	0.2084	0.0068	-0.0019
a_{13}	0.0	0.0002	0.0
a_{21}	0.0	2.498	5.1368
a_{22}	0.944	0.6261	0.2398
a_{23}	0.0	-0.0061	0.0
a_{31}	0.0	3.5901	3.786
a_{32}	0.5072	0.0043	0.001
a_{33}	0.0	0.0	0.0

Table B.1: Word delay fit constants.

The regression fit constants for the precharge delay estimation are given by equation 5.24. The values obtained are as follows:

Fit Constants	Small scale	Medium scale	Large scale
c_{11}	0.0	0.0	0.3922
c_{12}	0.0	0.0168	-0.0009
c_{13}	0.0025	-0.0002	0.0
c_{21}	0.0	0.0	4.4957
c_{22}	0.0	0.5207	-0.0014
c_{23}	0.1007	-0.0072	0.0

Table B.2: Precharge delay fit constants

The regression fit constants for Read-0 delay are given by equation 5.32.1. The data bus delay obtained is very minimal and we assumed b_{s1}= 1.0. The other values obtained are as follows:

Fit Constants	Small scale	Medium scale	Large scale
s_{21}	0.0	-4.45	86.1046
s_{22}	0.0	2.0239	0.7053
s_{23}	0.3114	-0.0198	-0.0008
s_{31}	0.0	11.7449	26.6
s_{32}	0.0	0.4003	0.1239
s_{33}	0.2333	-0.0004	0.0001

Table B.3: Sensing delay fit constants.