**University of Alberta**

# A Multilevel DRAM with Hierarchical Bitlines and Serial Sensing

by

**Lin Fu** ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2004

*Your file Votre référence*
*ISBN: 0-612-95746-2*
*Our file Notre référence*
*ISBN: 0-612-95746-2*

To Liang Fang, my wife.
Without her encouragement and great support, I would not have been able to finish this work.

# Acknowledgements

This research was funded by Micronet R&D, the Canadian Microelectronics Corporation and the University of Alberta. MOSAID Technologies Inc. and ATMOS Corp. (now part of MoSys Inc.) have provided intellectual property (IP) and technical assistance to the MLDRAM project. I would like to thank my supervisor, Dr. Bruce Cockburn, for his direction and patience throughout my M.Sc. research program. I would also like to thank my colleagues for their assistance and support: Jesus Hernandez Tapia, Tyler Brandon, Christian Giasson and John Koob.

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **1T1C** | One Transistor One Capacitor |
| **BL** | Bitline |
| **BSIM3v3** | Berkeley Short-Channel IGFET Model 3 version 3 |
| **CAM** | Content-Addressable Memory |
| **CAS** | Column Address Strobe |
| **CMC** | Canadian Microelectronics Corporation |
| **DRAM** | Dynamic RAM |
| **DRC** | Design Rules Check |
| **EDO** | Extended Data Out |
| **EEPROM** | Electrically Erasable and Programmable Read-Only Memory |
| **EPROM** | Electrically Programmable Read-Only Memory |
| **ERC** | Electrical Rules Check |
| **FPM** | Fast Page Mode |
| **IP** | Intellectual Property |
| **LVS** | Layout Versus Schematic check |
| **MLDRAM** | Multilevel DRAM |
| **PROM** | Mask-Programmable Programmable Read-Only Memory |
| **RAM** | Random-Access Memory |
| **RAS** | Row Address Strobe |
| **SA** | Sense Amplifier |
| **SBL** | Subbitline |
| **SDRAM** | Synchronous DRAM |
| **SPICE** | Simulation Program with Integrated Circuit Emphasis |
| **SRAM** | Static RAM |
| **TSMC** | Taiwan Semiconductor Manufacturing Corporation |
| **WL** | Wordline |

# Chapter 1

# Introduction

Semiconductor memories are essential parts of contemporary digital systems, from the fashionable and compact MP3 player to a full-scale parallel computer. They are widely used to store data values and program instructions and then retrieve them reliably on demand. Traditionally, the memory market has been driven by the tremendous demand for Dynamic RAM (DRAM) [18]. DRAM is used in large quantities as the main memory of most computer systems. Modern high performance microprocessors integrate increasing amounts of fast cache memory on the same chip as the processor. Cache memory is usually implemented using a faster but bulkier memory type called Static RAM (SRAM). In recent years, with the skyrocketing popularity of portable consumer media devices, nonvolatile flash memory has become another major driver of the memory market. In 2003, memory device sales rose to \$33.7 billion, up by 18.5 percent from \$28.4 billion in 2002 [11].

Due to the mission-critical role of the memory, reliability is an essential requirement. After some data is stored in a memory location, exactly the same data must be read out from the same address: any data loss or alteration is unacceptable. Because of the inevitable introduction of defects in the manufacturing process, redundancy techniques are widely used to ensure that most defective memory devices can be fully repaired using spare memory cells. Redundancy involves replacing the bad cells with spare cells so as to increase the yield of memory chips.

In modern digital systems, other properties are becoming important along with capacity, cost, speed and reliability. With further down-sizing in the minimum feature size, many kinds of electronic equipment can be integrated into much smaller packages and thus made portable and/or mobile. The cute mobile wireless telephone and the slim laptop computer have become indispensable tools of modern life. However, one ongoing challenge for portable devices is the limited energy capacity of the battery. Although researchers are continuing to improve battery technologies and portable power sources, by far the best practical strategy is to reduce the power needs of circuits by using low-power design techniques.

To better understand the requirements of the semiconductor memory, it is helpful to review its development history.

## 1.1 Development of Semiconductor Memories

Semiconductor memories were introduced in the late 1960s after the advent of bipolar transistors and large-scale integrated circuits. Before that, the memory unit in digital computers was implemented with delay lines, cathode-ray storage tubes, ferrite cores and thin magnate films, etc. [14]. Because of their high speed, compactness, ruggedness and potential of high-density integration, semiconductor memories came to dominate the memory market.

Three important types of semiconductor memories are CCD, bipolar and MOS memories. CCD memories are serial access memories and are only used in certain special applications, such as scanners and cameras. A competition between bipolar and MOS memories existed in the 1980s. Despite having faster access speeds, bipolar memories eventually lost out to MOS memory because of MOS memory's greater scalability to smaller feature sizes and its lower power requirements (in CMOS especially).

MOS technology has evolved progressively through PMOS, NMOS and finally CMOS [18]. PMOS was easiest to fabricate in the early 1970s, but it was superseded soon by faster NMOS using lightly P-doped substrates. Compared to CMOS circuits, NMOS circuits have the disadvantage of high static power consumption although they have higher gate density. CMOS became dominant in the memory market in the mid-1980s because it has ultra-low static power consumption.

Different memory technologies have been developed to accommodate the various requirements of applications. Generally, semiconductor memories can be categorized into two types: volatile memories and nonvolatile memories. Volatile memories can retain stored data only when the power is applied while nonvolatile memories can retain data even when the power is turned off [18].

Non-volatile memories include Mask-Programmable Programmable Read-Only Memory (PROM), Electrically Programmable Read-Only Memory (EPROM), Electrically Erasable and Programmable Read-Only Memory (EEPROM) and flash memory. ROM is still widely used for font tables, ideogram tables and bootstrap programs. The data is written permanently into the ROM at the factory. Flash memories are becoming the choice of non-volatile storage technology in mass market audio and video applications due to their large capacity and low cost per bit.

Volatile memories include Non-Random-Access Memory and Random-Access Memory (RAM). Non-Random Access Memory is only used in some special applications. One example is Content-Addressable Memory (CAM), which is widely used in cache memory to store and match a supplied address to those addresses previously stored in the memory. CAMs are also used in high-performance routers in data networks. RAMs include SRAM and DRAM. SRAM stores each bit of data in a flip-flop. SRAMs are used to implement the on-chip cache of microprocessors due to their fast access time. SRAMs are also used to implement the main memory of super-computers. In DRAM, data is stored as an electrical charge on each cell capacitor. Refresh (data rewrite) operations are required periodically to replenish the charge that has leaked away from the cell capacitors. That is why DRAM is called dynamic, opposed to static in SRAM. The main sources of the leakage current in the DRAM cell will be explained in the next section after the description of the DRAM cell structure. DRAM is widely used as the main memory in mainstream computers because of its high density and low cost.

Over the past 35 years, DRAM densities per chip have increased by successive factors of four going all the way from 256 bits in the late 1960s to 1 Gb today. Meanwhile, various data access modes, such as page mode, Fast Page Mode (FPM), Extended Data Out (EDO), etc. have been developed to increase the access speed. High-speed Synchronous DRAM (SDRAM) uses a clock signal to synchronize the memory operations at the interfaces and to simplify internal pipelining. The memory array is divided into several independent banks and the pipeline techniques are applied to get around the bottleneck of the relatively slow row access time in each memory array [15].

## 1.2 DRAM Basics

### 1.2.1 DRAM Organization

A simplified block diagram of a DRAM is shown in Figure 1.1. It consists of the memory array and the peripheral circuits. The peripheral circuits include the row and column address latches and decoders, the wordline drivers and the Sense Amplifiers (SA). The row and column address bits, coming from the processor, are latched by the address latches at the falling edge of the Row Address Strobe (RAS) and Column Address Strobe (CAS), respectively. After decoding, the wordline that is uniquely associated with the row address, is activated and all the memory cells enabled by this wordline are opened for access. The resulting signal voltage on each bitline is relatively weak (eg. 80-200 mV) and needs to be amplified to a full-strength rail voltage (e.g. $V_{DD}$ or $V_{SS}$) by the sense amplifier. After that, multiple cells which have the required column address are selected by closing the appropriate transistor switches, and the stored data are transferred to the data bus.

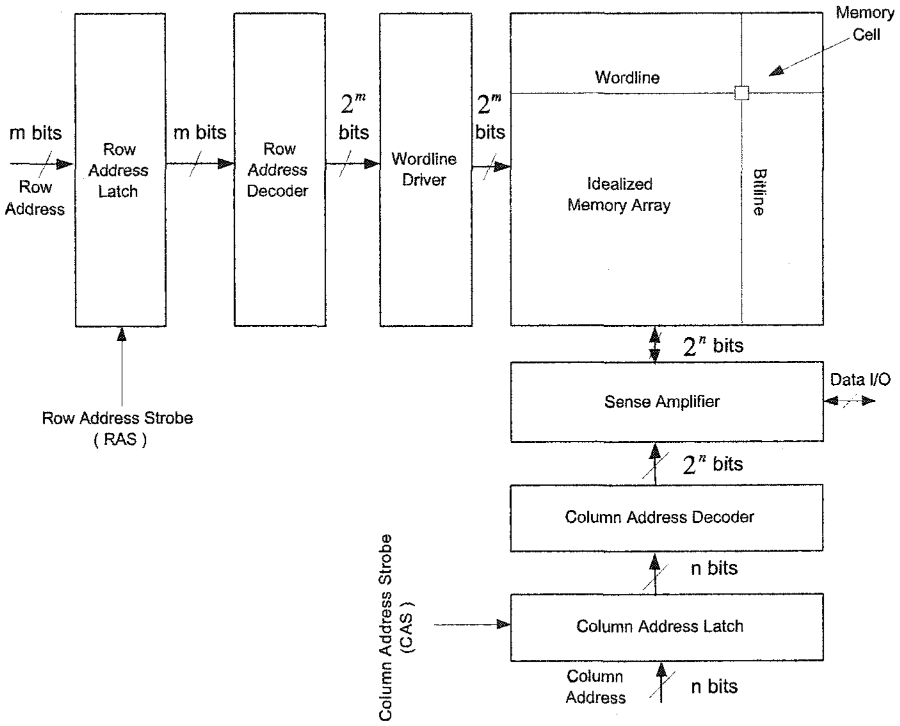


Figure 1.1: DRAM Block Diagram

#### 1.2.1.1 DRAM Array

Two bitline arrangements have been used for the memory array: open bitlines and folded bitlines. The open bitline structure is shown in Figure 1.2. The two bitlines connected with each SA come from two adjacent memory arrays. From the viewpoint of density, it is a good choice because a cell is present at every wordline and bitline intersection. However, the open bitline structure is more sensitive to the unequal noise induced onto the bitlines by the two arrays and parameter mismatches during manufacturing. Note that the two bitlines going to each SA are in different arrays and experience different operating environments. Also, the SA layout must fit within one bitline pitch in the vertical direction.

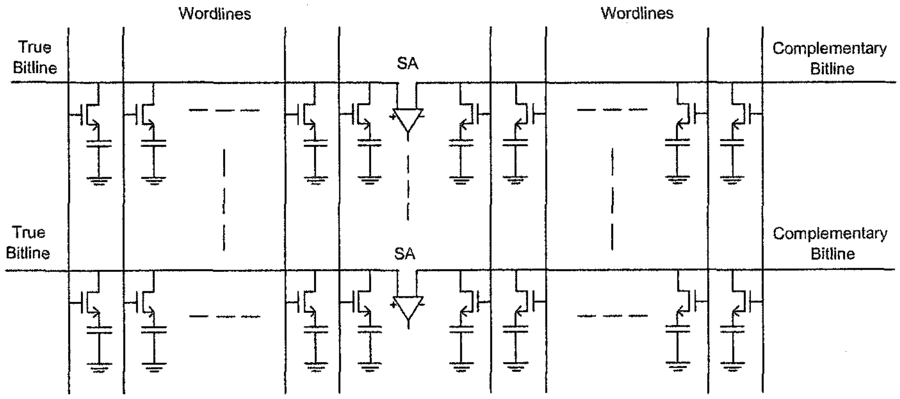


Figure 1.2: Open Bitline Schematics

The folded bitline structure is shown in Figure 1.3. The two bitlines connected to the same SA come from the same array therefore they see very similar electrical environments (e.g. capacitively coupled noise). This makes the folded bitline structure more robust than the above mentioned open bitline structure. However, the physical storage density is inherently lower because there is only one memory cell for every two adjacent bitlines. Nevertheless, this bitline structure is used in most contemporary DRAM designs.

#### 1.2.1.2 DRAM Cell

Early DRAMs used three NMOS transistors to implement each cell. The schematic of the 3T DRAM cell is shown in Figure 1.4. The cell uses the gate capacitance of transistor M3 as the storage node. No special process is required with extra steps to form capacitor structures. Although the area of this 3T DRAM cell is much bigger than that of more recent DRAM cell designs, it can still be useful in embedded DRAM because the memory and logic can use the same manufacturing process.

The DRAM cell can be further simplified to only one transistor and one ca-

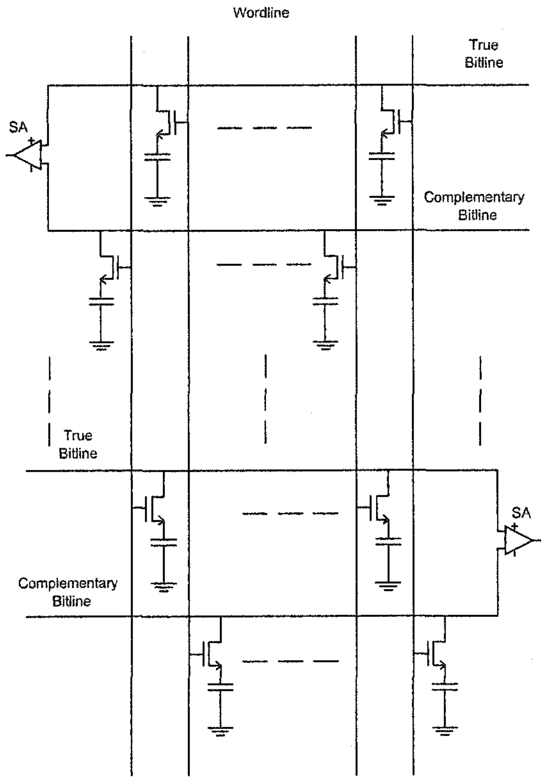
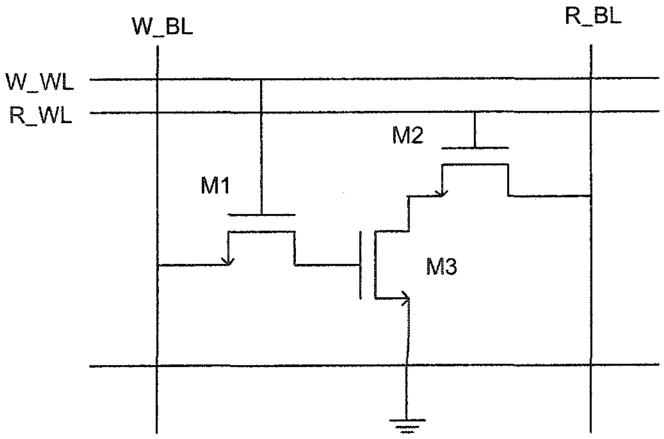


Figure 1.3: Folded Bitline Schematics



W_WL: Write Wordline
R_WL: Read Wordline
W_BL: Write Bitline
R_BL: Read Bitline

M1: Write Access Transistor
M2: Read Access Transistor
M3: Storage Transistor

Figure 1.4: Three-transistor DRAM Cell

pacitor. The One Transistor One Capacitor (1T1C) DRAM cell is the basis of the contemporary DRAM array. The structure of the 1T1C DRAM cell is shown in Figure 1.5. The capacitor stores a charge whose potential (either more positive or more negative with respect to a reference potential) encodes the stored data ('0' or '1'). The transistor acts as the access switch into the memory cell from the bitline. The memory cell is connected to the bitline when the wordline is asserted high and the access transistor is ON, and is disconnected from the bitline when the wordline is de-asserted low and the access transistor is OFF.

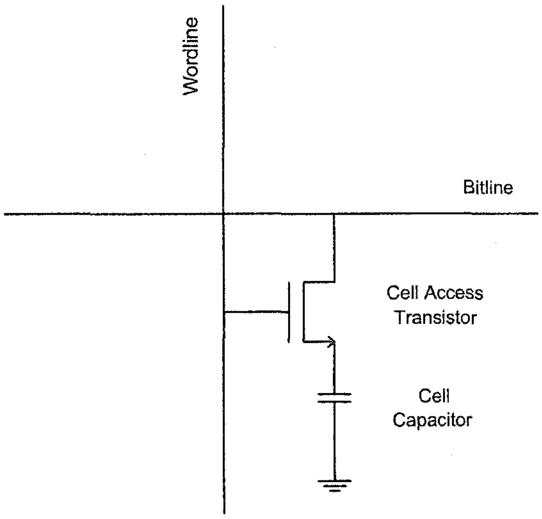


Figure 1.5: 1T1C DRAM Cell

Since the data is stored as a charge on the capacitor, the signal will be depleted when the cell is in standby mode because charge leaks away via finite resistance paths to the substrate. There are four main sources of charge depletion in the DRAM cell that are due to the physical properties of the cell capacitor and the access transistor. Figure 1.6 illustrates these paths in the DRAM cell. The first one is the leakage current between the two plates of the cell capacitor through the imperfectly insulating dielectric, which is inevitable for real capacitors. The second path is the subthreshold current from the storage node to the bitline through the open, but not infinite resistance, access transistor. The third path is junction leakage through the reverse-biased source-to-substrate interface of the source node of the access transistor. The last one is due to α particle radiation (two neutrons and two proton nucleus), which comes from radioactive impurities in the package materials. The collision between the α particle and the atoms in the cell generates many positive and negative carriers, which can disperse and/or change the cell charge signal.

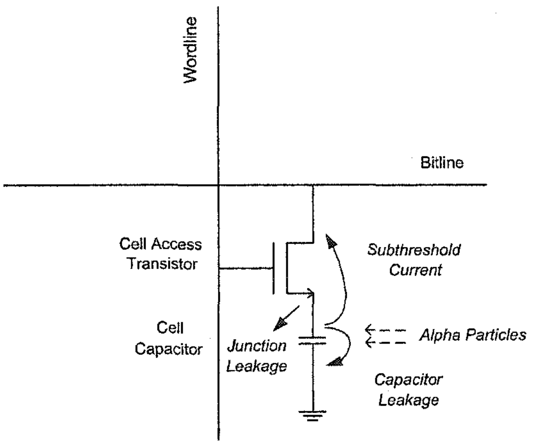


Figure 1.6: Charge Leakage Paths in the DRAM Memory Cell

In order to minimize the area of the memory cell, two adjacent cells need to share the same drain terminal and drain contact to the bitline. The layout of a memory cell pair with a shared-drain connection to the bitline is illustrated in Figure 1.7.

## 1.2.2 DRAM Operation

### 1.2.2.1 Charge Sharing

The operation of the DRAM memory cell is based on the concept of charge sharing, as illustrated in Figure 1.8. Charge sharing refers to a charge redistribution between the cell capacitor and the capacitance of the associated bitline.

Assume that the capacitance of the memory cell is $C_{cell}$ and the capacitance of the bitline is $C_{BL}$. Before charge sharing, the cell voltage is $V_{cell}$ and the bitline

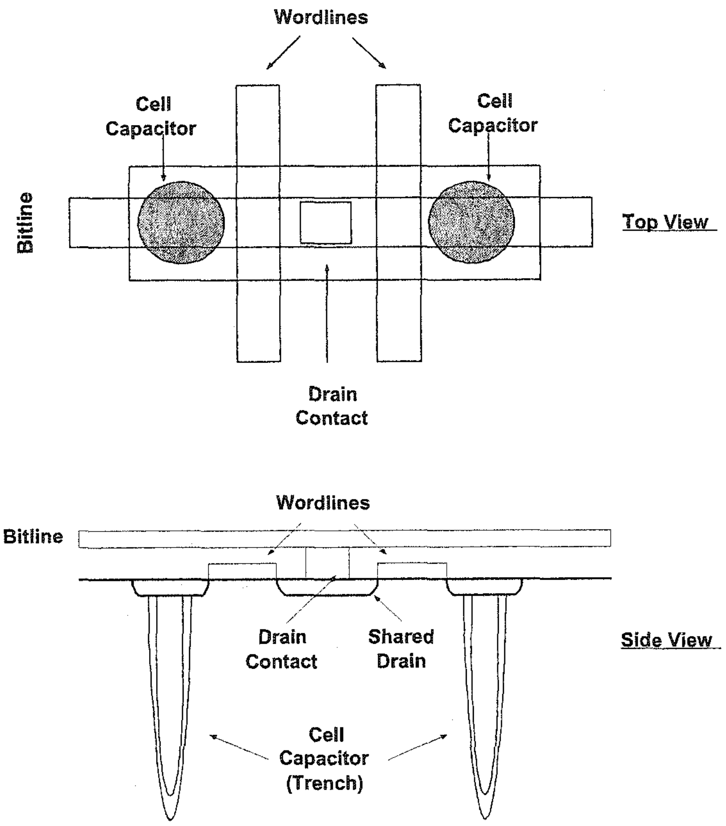


Figure 1.7: Two Adjacent Memory Cells with Shared Drain

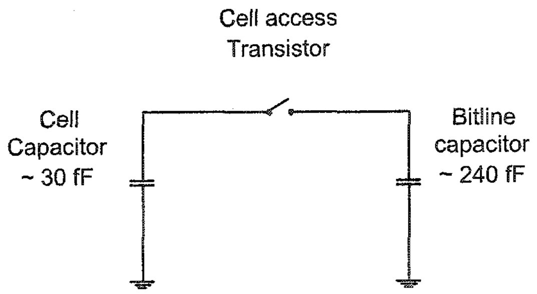


Figure 1.8: Charge Sharing

voltage is precharged to $V_{pre}$, which is usually mid-way between $V_{DD}$ and $V_{SS}$. Typically, $V_{cell}$ is near one of the two supply voltages $V_{DD}$ and $V_{SS}$. Then when the access transistor is closed, the charge stored in the cell is redistributed with the bitline and the resulting signal voltage $V_{signal}$ on the connected cell and bitline is given as follows:

$$V_{signal} = \frac{V_{cell}C_{cell} + V_{pre}C_{BL}}{C_{cell} + C_{BL}}$$

In a typical DRAM chip, the bitline capacitance $C_{BL}$ is much bigger than the cell capacitance $C_{cell}$. If we assume typical values of $C_{BL} = 240\ fF$, $C_{cell} = 30\ fF$, $V_{cell} = 1.8\ V$ and $V_{BL} = 0.9\ V$, the resulting signal voltage after charge sharing is $1\ V$. Here the voltage of the bitline increases by only $0.1\ V$. Such a weak signal can not be used by the external circuit directly: it would get lost in the noise, which is pretty strong in the external circuits. A SA is required to amplify such a small signal to a full-strength rail voltage, either $V_{DD}$ or $V_{SS}$.

#### 1.2.2.2 Basic SA

The schematic diagram of a typical DRAM SA block is shown in Figure 1.9. The SA block contains the isolation transistors controlled by ISO, the precharge and equalization devices controlled by EQ, and the SA circuit itself (formed by transistors M5, M6, M7, M8, M9 and M10). The isolation transistors are pass transistor switches which are used to isolate the SA from the bitline when the bitline needs to be isolated (floated). The precharge and equalization device can precharge the bitline to $V_{pre} = \frac{V_{DD}}{2}$ before the access to the cell. The pass transistor switch between the true bitline and the complementary bitline can connect them together to ensure that they have exactly the same precharge voltage, which is important to the following sensing operation.

The commonly-used SA is actually a pair of cross-coupled inverters, as shown in Figure 1.10. The SA can be used to sense and amplify the weak voltage difference between the bitline and complementary bitline. Since it is an inverter ring, it

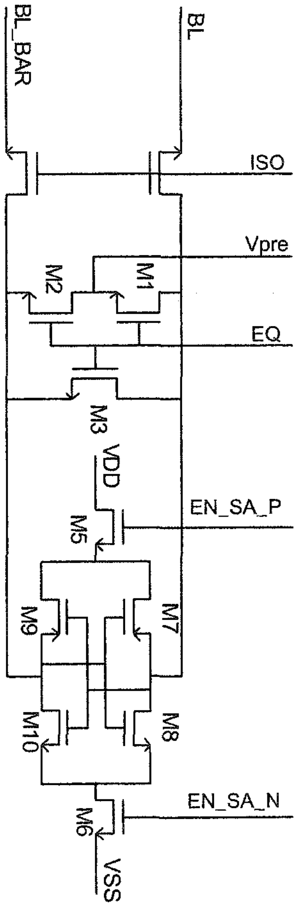


Figure 1.9: Schematic Diagram of the Basic SA Block

can also latch the sensing result. This feature is commonly used to hold a row of data bits, seen by users as a "page".

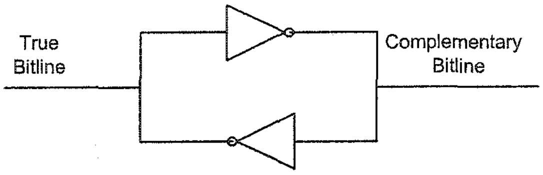


Figure 1.10: Inverter Ring

When sensing a logic '1', after the access of the cell, the voltage of the true bitline BL becomes a little higher than the signal of the complementary bitline

BL_BAR. First, EN_SA_N is asserted, transistor M6 is closed, and the source terminals of transistors M8 and M10 are connected to ground $V_{SS}$. Because the voltages of both BL and BL_BAR are higher than the threshold voltage $V_{th}$ of the transistors, both BL and BL_BAR begin to discharge toward $V_{SS}$ and their voltages begin to decrease. Since BL_BAR is lower in voltage than BL, its voltage will decrease to the value of $V_{th}$ first. Since BL_BAR is connected to the gate of transistor M8, it can open transistor M8 now to stop the discharging of BL. Then EN_SA_P is asserted to close transistors M5 and M9 and thus pull BL up to $V_{DD}$. The bitline waveforms when sensing a '1' are shown superimposed in Figure 1.11.

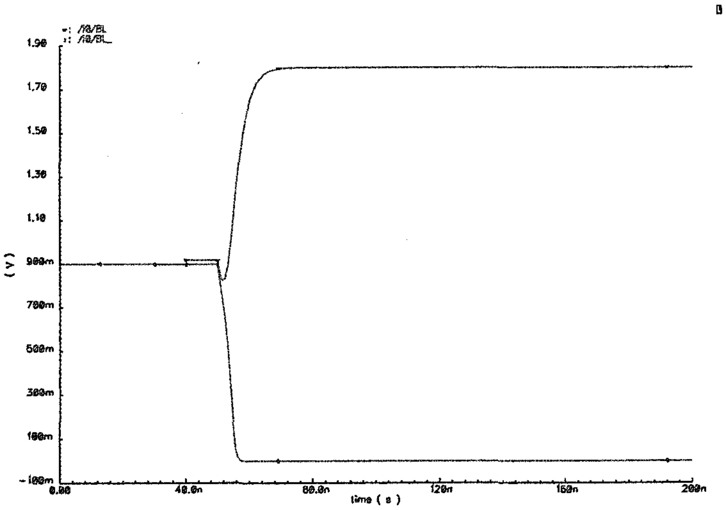


Figure 1.11: Waveform When Sensing a '1'

When sensing a logic '0', the voltage of BL becomes a little lower than BL_BAR after the cell access. When EN_SA_N is asserted, the discharge of both BL and BL_BAR begins. Because BL is lower than BL_BAR, it will reach the value of $V_{th}$ first. Since BL is connected to the gate of transistor M10, it can open transistor M10 now to stop the discharge of BL_BAR. Then EN_SA_P is asserted and transistors M5 and M9 are closed to pull BL_BAR up to $V_{DD}$. The waveform of the sensing logic '0' is shown in Figure 1.12.

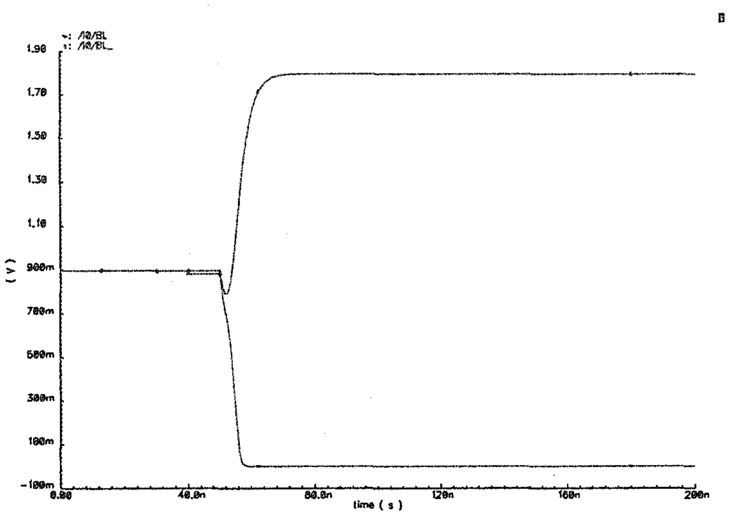


Figure 1.12: Waveform When Sensing a ‘0’

#### 1.2.2.3 SA with Input Offset Cancelation

From the description of the operation of the above SA, it is apparent that the threshold voltage of the left and the right NMOS transistors should be matched with each other to ensure fair sensing. Unfortunately, small variations in the component properties, such as the length and width of the transistor channels, are inevitable in any semiconductor manufacturing process. These variations will change the characteristics of the circuit.

Several SAs with Input Offset Cancellation have been proposed to compensate for these variations [19] [21] [28]. In reference [19], Shunichi Suzuki proposed a new sensing technique built on a dynamic latch amplifier. The block diagram of the SA circuit is shown in Figure 1.13.

The drain and the gate of the NMOS transistors can be connected together to form a diode, which has a forward voltage drop equal to the threshold voltage $V_{th}$ of the NMOS transistor. This configuration is illustrated in Figure 1.14. With such a connection, the true bitline and its complement can be precharged to $V_E + V_{th}$, instead of $\frac{V_{DD}}{2}$ as in the conventional SA. If we assume that $V_E$ is 0.45 V, and $V_{th}$

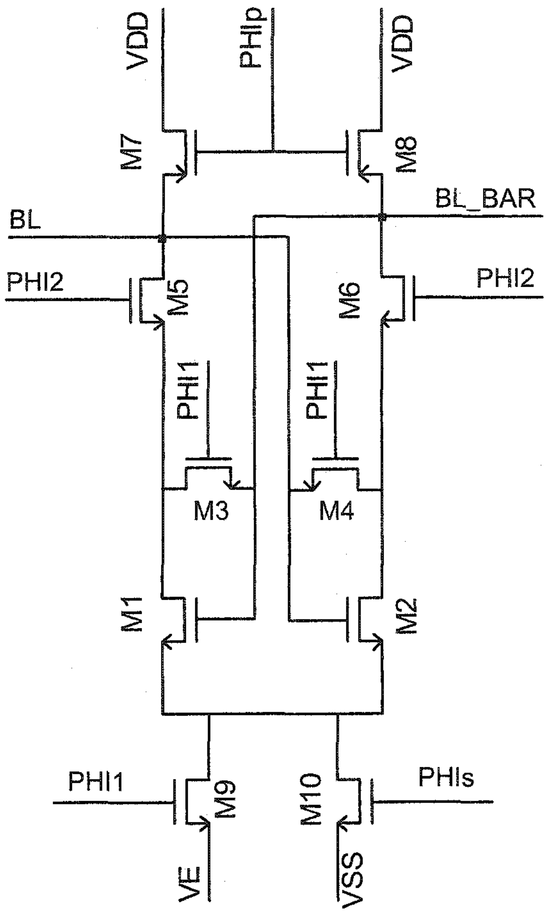


Figure 1.13: Block Diagram of the SA Circuit

also has a value around 0.45 V, then the precharge voltage will be around $\frac{V_{DD}}{2}$. Any variations in the threshold voltages are compensated for by the different precharge voltages. When the transistors M3 and M4 are opened and the transistors M5 and M6 are closed, a dynamic flip-flop amplifier is formed, which can sense the voltage difference between the true and complementary bitlines.

When sensing a logic ‘1’, after a cell access, the voltage of BL becomes a little higher than that of BL_BAR. First, PHIp is asserted high, causing transistors M7 and M8 to be closed, and so the bitline and its complement are precharged to $V_{DD}$. Then PHI1 is asserted causing transistors M3 and M4 to be closed. Thus transistors M1 and M2 are now connected as two diodes. At the same time, the transistor M9

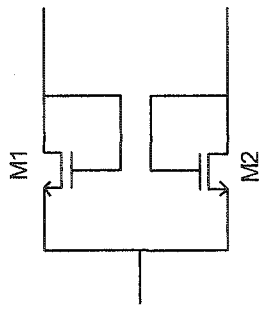


Figure 1.14: Threshold Voltage Compensation using the Diode Connection

is closed and the voltage of the true bitline and its complement drop to $V_E + V_{th}$. Note that the $V_{th}$ can be slightly different for the two primary sensing transistors, M1 and M2. After the adjusted precharge of the bitline pair, the cell is accessed and the voltage of the true bitline rises a little. Then PHI1 is de-asserted and PHI2 is asserted, and sensing begins. Because the voltages of both BL and BL_BAR are higher than the threshold voltage $V_{th}$ of the transistors, both BL and BL_BAR begin to discharge toward the ground potential, and their voltages will begin to decrease. Since BL_BAR is lower in voltage than BL, it will drop to the value of $V_{th}$ first. Since BL_BAR is connected to the gate of transistor M1, it can open transistor M1 now to stop the discharge of BL. Then PHIp is asserted to close transistors M7 and M8, causing BL to be pulled up to $V_{DD}$. The bitline waveforms when sensing a '1' are shown in Figure 1.15.

When sensing a logic '0', the voltage of BL is bumped a little lower than BL_BAR after the access to the cell and the charge sharing between the cell and bitline. The true bitline and its complement are also precharged to $V_{DD}$ at first. Then PHI1 is asserted and transistors M3 and M4 are closed. This causes the voltage of the bitline and its complement to drop to $V_E + V_{th}$. After the precharge of the bitline pair, the cell is accessed and the voltage of the bitline drops a little. Then PHI1 is de-asserted and PHI2 is asserted and sensing begins. Because the voltage of BL is lower than that of BL_BAR, it will reach the value of $V_{th}$ first. Since BL is connected to the gate of transistor M2, it will open transistor M2 now to stop

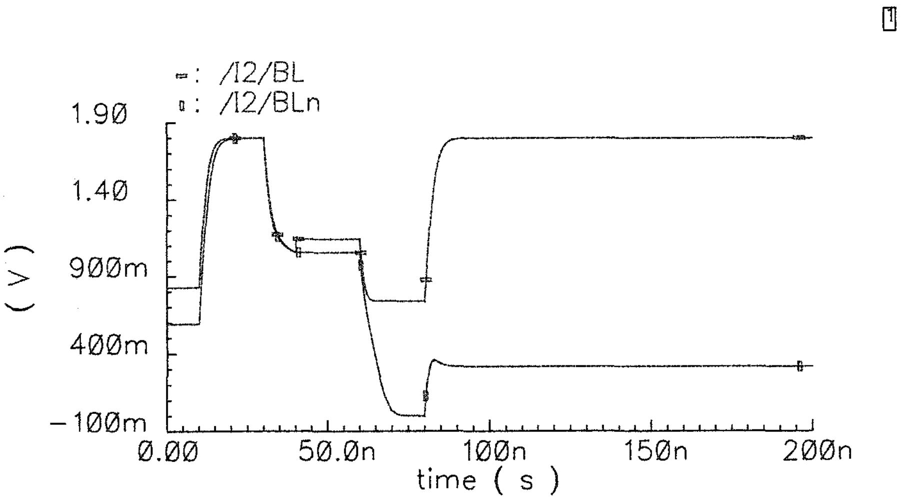


Figure 1.15: Waveform When Sensing a '1'

the discharge of BL_BAR. Then PHIp is asserted, closing transistors M7 and M8 to pull BL_BAR up to $V_{DD}$. The bitline waveforms when sensing a logic '0' are shown in Figure 1.16.

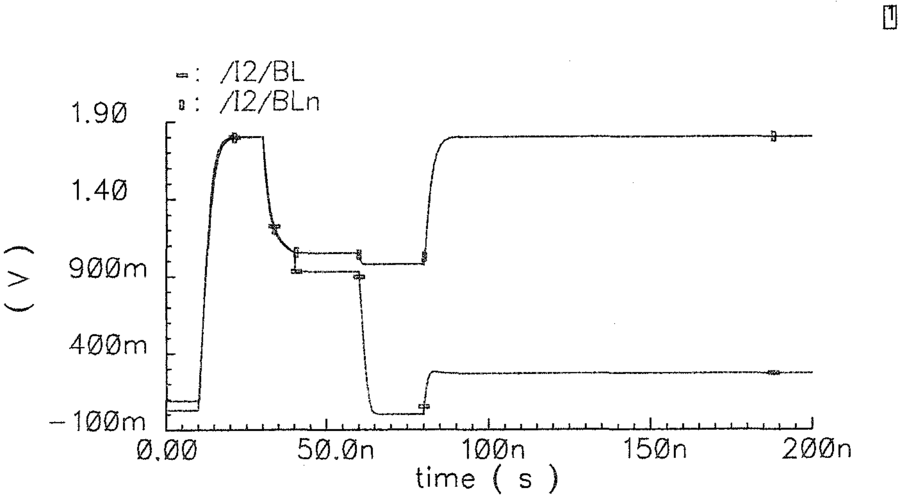


Figure 1.16: Waveform When Sensing a '0'

## 1.3 Introduction to Multilevel DRAM

Generally, there are two ways to increase the storage density of a DRAM. The first way is to reduce the area of the memory cell. Since the cell capacitor occupies a large fraction of the area of the memory cell, three-dimensional capacitor structures, such as the stacked and trench capacitors, have been developed to reduce the size of the cell capacitor [7][5]. Also higher dielectric constant materials (e.g. ONO, $Ta_2O_5$) have been adopted or considered as the insulator instead of the conventional silicon dioxide to further increase the capacitance. Using such materials, a large cell capacitor can be made within a minimum area.

Another technique, called Multilevel DRAM (MLDRAM), has also been investigated by researchers as the means to record more than one bit in each single memory cell. In MLDRAM, instead of only using two nominal voltages, $V_{SS}$ and $V_{DD}$, to represent logic '0' and '1', multiple equally-spaced voltages between $V_{SS}$ and $V_{DD}$ are used to represent binary information with more than one bit. In the next section, the multilevel encoding method will be introduced.

### 1.3.1 Multilevel Encoding

The four-level MLDRAM signalling scheme shown in Figure 1.17 will be used here as an example.

In order to record two bits in one cell, four equally-spaced signalling voltages between $V_{SS}$ and $V_{DD}$ are adopted, and each of these voltages can be written into the cell to represent one possible combination of the two bits. Three reference voltage are used when reading back previously stored multilevel signals, and each of them is located mid-way between two adjacent data voltages. Note that the noise margin in 4-level MLDRAM is reduced to $\frac{1}{3}$ of $\frac{V_{DD}}{2}$, which is the noise margin for conventional DRAMs.

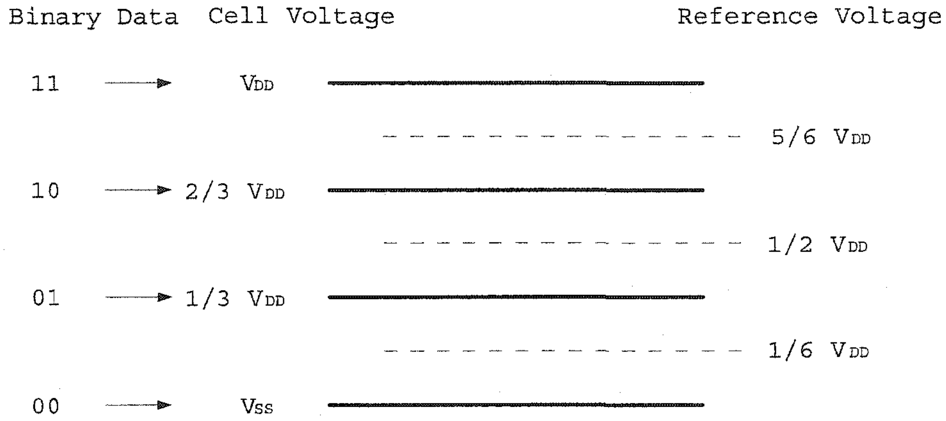


Figure 1.17: 4-Level MLDRAM

### 1.3.2 Challenges in MLDRAM

Multilevel encoding has been proposed in flash memories and used in commercial productions for a few years already [6]. As for MLDRAM, it is still on the way, because of the great technical challenges that exist.

A first major challenge are the reduced noise margins as mentioned before. The data signal becomes much smaller after the charge sharing operation between the capacitance $C_{cell}$ of the memory cell and the capacitance $C_{BL}$ of the bitline. Under the previous assumption about the capacitances and voltages, the attenuated noise margin is only 50 mV. This imposes a stricter requirement on the sensitivity of the SA.

A second major challenge is the area overhead. In order to sense the multilevel voltage, more circuits are required for sensing, coding and decoding. All these additional circuits need to fit into the pitch of the present DRAM array. This constraint complicates the layout work. The area of the extra overhead must not overwhelm the extra density that is gained in the cell array.

## 1.4 Outline of the Thesis

In this thesis, we have so far introduced the basic concepts of DRAM and MLDRAM in Chapter 1. In chapter 2 we will review the previous work on MLDRAM, especially, the progress on MLDRAM technology made at the VLSI lab in the Department of Electrical and Computer Engineering at the University of Alberta.

In chapters 3 and 4, we will first introduce a new serial-sensing MLDRAM, and then describe its implementation and simulation. We also compare it with other published MLDRAM schemes.

This thesis concludes with a discussion of how to make the MLDRAM competitive to conventional DRAM. Suggestions are also given for future research work on MLDRAM.

# Chapter 2

# Previous Work on Multilevel DRAM

With MLDRAM, we can increase the storage density by storing more than one bit in a single memory cell. Instead of only using two nominal voltage levels, $V_{DD}$ and $V_{SS}$, to represent logic '1' and '0', three or more equally-spaced voltages between $V_{DD}$ and $V_{SS}$ are used to encode a word of binary information. Each of such words would represent more than one bit of information. The signalling that could be used in a 4-level MLDRAM was shown in Figure 1.17.

The 1T1C memory cell, which has proven reliability and area efficiency, is a good choice to be reused in MLDRAM. Reuse of the proven technologies can save time and minimize risk in the implementation of an experimental one. DRAM technology will continue to be scaled aggressively due to the large research and development funds being invested by many companies around the world.

In MLDRAM, more voltage levels are inserted between $V_{DD}$ and $V_{SS}$, so the noise margins are reduced. If the same memory cell and power supply voltage are used in an $N$-level MLDRAM, the resulting noise margins are only $\frac{1}{N-1}$ of those in a conventional DRAM [4]. This makes MLDRAM more sensitive to the capacitively coupled noise in the densely packed cell array and cell leakage, which inevitably appear as sensing errors that affect all DRAM chips. Meanwhile, the accuracy requirement on the reference voltage is increased because a small error in the reference voltage can make a sensing operation false. With reductions of the circuit dimension, the operating voltage must also be reduced to avoid dielectric breakdown in the transistors. Since the noise margins are proportional to the supply voltage, this makes the noise margins of DRAM even smaller.

In order to retrieve the multiple bits of information from one or more MLDRAM cells, additional peripheral circuits are required to decode the possible combinations of cell voltage levels.

In MLDRAM, we can retrieve the multiple bits of information either in parallel or in serial. Parallel sensing is inherently faster but more SAs are required to work at the same time and that may increase the peripheral area significantly. In serial sensing, the speed is inherently slower and several copies of the data signal need to be created and held for multiple serial sensing operations. These signal copies are in turn susceptible to being affected by the coupling noise, especially during the time when they are waiting to be sensed.

In order to ensure reliable sensing, accurate reference voltages are required. The reference voltages can be generated globally or locally. With global reference generation, all the memory cells share the same reference generation circuit. Unfortunately, we then could have problems when distributing these reference signals around the chip to the memory cells. In a large scale chip, the delivery routes are quite different for the different cells, which may cause the voltages delivered to different cells to differ from each other. This can cause errors in the sensing operations.

The speed of the MLDRAM is slower than that of the conventional DRAM because of the additional sensing and restoration operations. Even so, MLDRAM may still find suitable applications as a file memory, which can fit into the large speed gap between the fast DRAM main memory and the slow hard disk storage [8]. At the same time, any cost per bit advantage of MLDRAM over DRAM will keep it more competitive as the DRAM technology shrinks further in size.

Over the last decade, researchers have been working on MLDRAM and some good schemes have been proposed. Experimental chips were made to investigate the real problems in the proposed schemes. Research work on MLDRAM has been carried out for the past five years at the VLSI lab at the University of Alberta.

In the following sections, we will introduce the published MLDRAM schemes and we will also discuss their advantages and disadvantages.

## 2.1 Furuyama's MLDRAM

In Furuyama's MLDRAM, a 4-level MLDRAM is implemented with global reference generation and parallel sensing. Three different reference voltages are compared with three identical copies of the signal voltage at the same time and one three-bit unary code word, that records the parallel sensing decisions, is obtained as the result. This unary code word is converted into one two-bit binary code word with a decoder, according to the mapping in Table 2.1 [9].

| Binary Bits | Unary Bits |
|---|---|
| 00 | 000 |
| 01 | 001 |
| 10 | 011 |
| 11 | 111 |

Table 2.1: Unary to Binary Conversion Table

A high-level block diagram of Furuyama's MLDRAM is shown in Figure 2.1. The bitline block is divided into three subbitline blocks labeled A, B and C. Each subbitline block has a subbitline pair with the same large number of data cells and two dummy cells connected. Each subbitline block has its own SA for parallel sensing. A more detailed schematic of the subbitline block is shown in Figure 2.2. Switch transistors are placed between adjacent subbitlines to connect or disconnect them according to the control signals from the control logic.

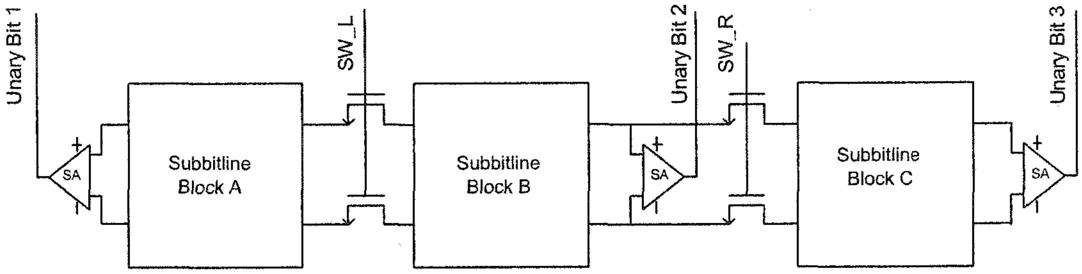


Figure 2.1: Block Diagram of Furuyama's MLDRAM

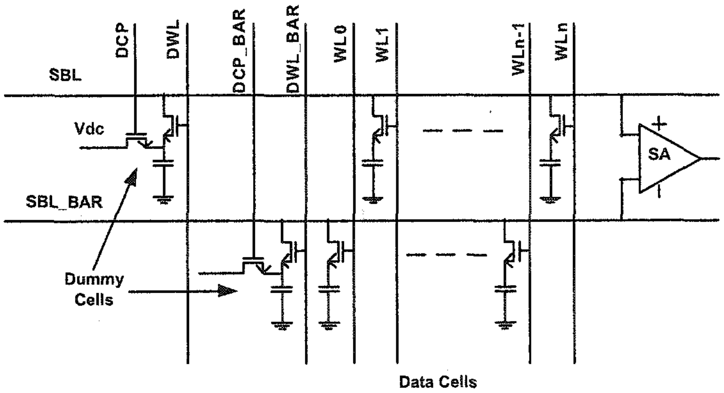


Figure 2.2: Schematic of the Subbitline in Furuyama's MLDRAM

The three reference voltages are generated globally outside the memory core. The reference signals are delivered to the dummy cells in the subbitline blocks. Because of the different locations of the subbitline blocks in the chip, the characteristics of the associated signal distribution interconnection are likely to be slightly different from each other and the resulting reference voltages will consequently be different. Such differences can cause errors in the following sensing operation.

Before the sensing operation, three different reference signals are stored in the dummy cells by activating the DCP signal in the three subbitline blocks. In order to get three identical copies of the data signal, the switch transistors are first closed by activating SW_L and SW_R to connect all the three subbitlines together to form a full bitline that is then precharged to $\frac{1}{2}V_{DD}$ and isolated. Then the wordline of the addressed memory cell is asserted and the charge stored in the memory cell is redistributed along the bitline. As a result, the voltage of all three subbitlines will be bumped together a little bit either up or down. After that, the switch transistors are opened by de-activating SW_L and SW_R, and three identical copies of the data signal are isolated on every subbitline. Now the reference cells on the complementary subbitline are accessed by asserting DWL_BAR, and the reference voltages are formed separately on the three complementary subbitlines.

One tricky problem here is that the charge sharing capacitance of the reference cell is slightly different from that of the data cell. Each reference cell is connected with only one subbitline, but the data cells need to connect with the full bitline, which is a combination of three subbitlines and two switch transistors. In order to get the required reference voltages, the voltage in the reference cell needs to be a little smaller in magnitude (with respect to the bitline precharge voltage) than the predicted one.

Next, the SA is activated, and three unary bits are captured as the results of the sensing operations. The bits are latched and decoded into two binary bits and passed to the outgoing data interface.

In order to restore the full-strength multilevel voltage in the memory cell, we can use charge sharing among the equal subbitline capacitances to create the required signal voltages. After charge redistribution, the wordline is de-asserted and the multilevel voltage, which is equal to the original voltage, is trapped on the data memory cell capacitor. In this way, all of the memory cells along the same wordline can be refreshed at the same time after the read operation.

A write operation to a memory cell is similar to the restore operation of a previously written signal except that some of the restored data is replaced by data supplied by the incoming data interface. After the conversion from the binary to unary, the three resulting unary bits are connected with the appropriate subbitlines for the addressed column(s). Then the data signals are written into the memory cells after charge sharing to create the multilevel data signals.

The parallel sensing operation is fast, but in order to provide parallel sensing, each subbitline needs its own SA. Here, three SAs are required for the three subbitlines. Compared with conventional DRAM, Furuyama's MLDRAM requires more area to implement the two more SAs for each bitline pair. Extra area erodes the goal of increasing the storage density.

## 2.2 Gillingham's MLDRAM

Gillingham's MLDRAM can be characterized as a local reference generation and serial sensing scheme. In order to generate and deliver more accurate reference voltages, local generation is proposed without too much circuit overhead [10].

A schematic of Gillingham's MLDRAM is shown in Figure 2.3. This scheme is also intended for a 4-level MLDRAM. The bitline is divided into two subbitlines on the left and right sides. Between these two subbitlines, a switch matrix is used to connect or disconnect the subbitline according to the control signals (C, Cn, X, Xn) from control logic in the periphery. The memory cells and one SA are connected within each subbitline block.

The sensing of the two binary bits is carried on serially in two steps. First, the Most Significant Bit (MSB) is sensed; then, based on the sensed MSB bit, the reference voltage for the Least Significant Bit (LSB) is generated, which is either $\frac{1}{6}V_{DD}$ or $\frac{5}{6}V_{DD}$. The LSB is then sensed with this reference voltage.

In order to sense the MSB, all the subbitlines are first precharged to $\frac{1}{2}V_{DD}$, and

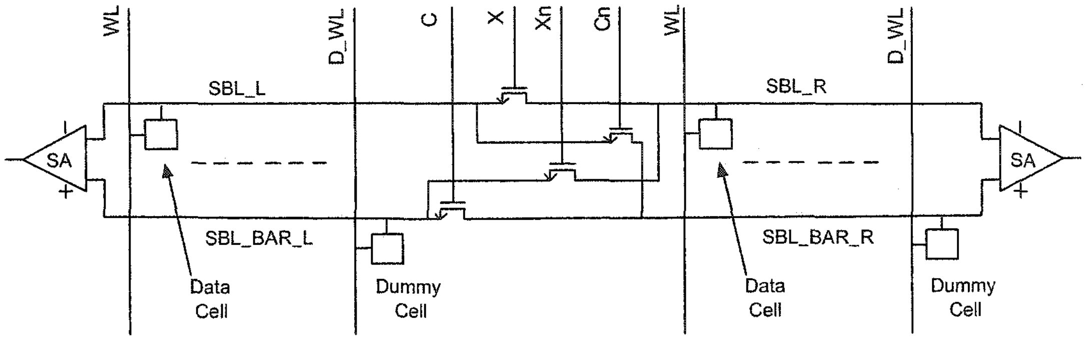


Figure 2.3: Block Diagram of Gillingham's MLDRAM

control signal X is asserted high to connect the two subbitlines (SBL_L and SBL_R) together. Then the wordline is asserted to dump the data cell charge onto the connected subitline. After that, the switch controlled by X is opened to disconnect the two subbitlines. Thus two copies of the data signal are presented on the two subbitlines.

Now control signal C is de-asserted to disconnect the two complementary subbitlines SBL_BAR_L and SBL_BAR_R. Next, the SA on the left is activated and the MSB is evaluated. The MSB is then trapped in the data cell by de-asserting the wordline.

To sense the LSB, we need to determine which reference voltage is to be used, either $\frac{1}{6}V_{DD}$ or $\frac{5}{6}V_{DD}$. Here, a subtle method is used to get the new reference voltage. First, the switches controlled by C and Cn are closed to connect the three subbitlines (the left-top, left-bottom and right-bottom) together, then they are precharged to $\frac{1}{2}V_{DD}$. After cell access of the data cell storing the MSB, the resulting voltage after charge sharing is as follows:

$$V_{ref1} = (S - \frac{V_{DD}}{2})(\frac{C_{cell}}{3C_{SBL}+C_{cell}}) + \frac{V_{DD}}{2}$$

where $C_{cell}$ is the capacitance of the memory cell, $C_{SBL}$ is the capacitance of the subbitline, and $S$ is the voltage of the MSB, which is $V_{DD}$ or $V_{SS}$. If we have the desired reference voltage, $\frac{1}{6}V_{DD}$ or $\frac{5}{6}V_{DD}$, charge-shared onto two subbitlines, it produces the following voltage:

$$V_{ref2} = (R - \frac{V_{DD}}{2})(\frac{C_{cell}}{2C_{BL} + C_{cell}}) + \frac{V_{DD}}{2}$$

where $R$ is the full-strength reference voltage, which equals to $\frac{1}{6}V_{DD}$ or $\frac{5}{6}V_{DD}$, so $R = \frac{2}{3}S + \frac{V_{DD}}{6}$. The subbitline capacitance $C_{SBL}$ is much greater than that of the cell $C_{cell}$. This leads the resulting reference voltage $V_{ref1}$ to be equal to the required reference voltage $V_{ref2}$.

With the new reference voltage now present on SBL_BAR_R and the second copy of the data signal present on SBL_R, the SA on the right side is activated to get the LSB.

In order to restore the correct multilevel data voltage into the memory cell, charge sharing is used to combine the MSB and LSB as follows,

$$V_{cell} = \frac{2}{3}MSB + \frac{1}{3}LSB$$

To implement this operation, first the switch controlled by $C_n$ is closed to connect together the left subbitline and the right complementary subbitline together and the left SA is used to charge these two subbitlines to the full-strength voltage corresponding to the MSB. Then by asserting the signal that closes $X$, charge sharing occurs between the MSB and LSB. Finally, the restored data voltage is trapped in the memory cell by de-asserting the corresponding wordline.

The write operation is similar to the restore operation except that the binary data are supplied by the incoming data interface and the sensed data bits are overwritten.

Gillingham's method can thus generate the required data and reference signals with relatively little circuit overhead. Local reference generation and the use of a dummy cell to generate the LSB reference make it more robust to noise and imbalance during sensing. A global reference generation and distribution system is not required. The drawback here is the low speed of the two-step serial sensing. Also, two SAs are required in this design for each bitline pair.

## 2.3 MLDRAM Research at the University of Alberta

MLDRAM research at the University of Alberta began with Birk's work on MLDRAM. ML1 and ML2 were implementations of Gillingham's scheme. Albert Chan implemented Birk's scheme with ML3 to verify its functionality [4][1]. The succeeding work by Yunan Xiang [26] and Sue Ann Ung [23] were all based on Birk's scheme, with the expandability and testability of Birk's original design being explored in this research. In this section, we will first describe Birk's scheme in detail, and then briefly discuss Xiang and Ung's implementations.

### 2.3.1 Birk's MLDRAM

Birk's MLDRAM design combines local reference generation and parallel sensing. A block diagram of Birk's design is shown in Figure 2.4. Like Furuyama's scheme, the bitline is divided into three equal subbitlines and the adjacent subbitlines can be connected with horizontal switch transistors. Three bitline pairs can also connected together with the reference transistors in the vertical direction.

As shown in Figure 2.4, the three-by-three subbitline array is divided into three sections (labeled as L, C and R) in the horizontal direction, and into three groups in the vertical direction (labeled as T, M and B).

Within each subbitline block, the data cell is connected with the subbitline. The subbitline can also be precharged to three different voltages, $V_{DD}$, $V_{SS}$ and $\frac{V_{DD}}{2}$. With these different possible precharge voltages, the three subbitline can be connected in the vertical direction and three different reference voltage, $\frac{V_{DD}}{6}$, $\frac{V_{DD}}{2}$ and $\frac{5}{6}V_{DD}$, can be formed using charge sharing among the subbitlines. The required

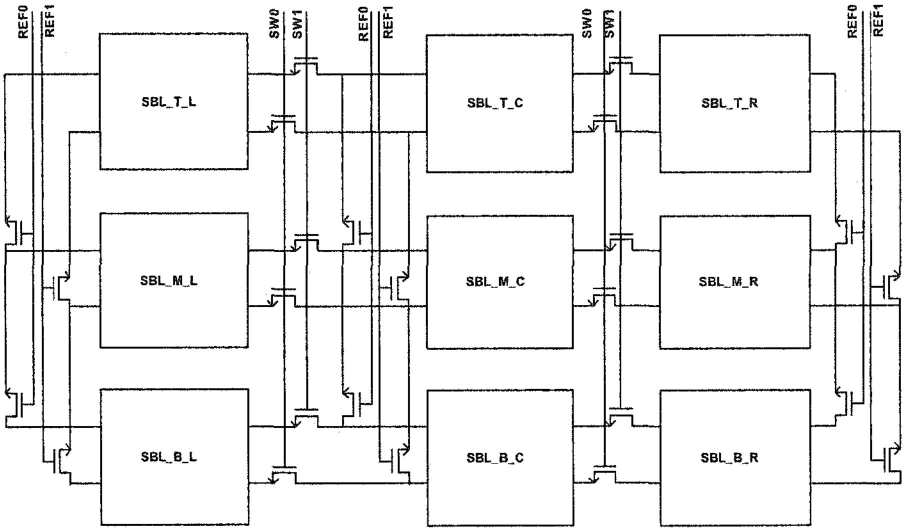


Figure 2.4: Block Diagram of Birk's MLDRAM

precharge voltages are shown in Table 2.2. The three reference voltages are generated by averaging the three columns of voltage in Table 2.2, labeled as L, C and R.

|  |  | Subbitline Sections |  |  |
|---|---|---|---|---|
|  |  | L | C | R |
| Subbitline Pair | T | VSS | VDD | VDD |
|  | M | ½ VDD | ½ VDD | ½ VDD |
|  | B | VSS | VSS | VDD |

Table 2.2: Reference Generation Table

The read operation works in this way. First, the subbitlines are connected together horizontally and precharged to $\frac{1}{2}V_{DD}$, then the addressed wordline is asserted. After charge sharing between the memory cell and the three horizontally-connected subbitlines, the switch transistors are opened, and three identical data voltages are isolated on the three separated subbitlines. Then the three reference voltages are formed in parallel on the complementary bitlines. First, the REF0 and REF1 control signals are asserted to connect the complementary subbitlines vertically. Then the reference wordline is activated to cause charge sharing along the vertically-connected subbitlines. When the REF0 and REF1 signals are de-asserted, three different reference signals are isolated on the complementary subbitlines. After all these steps, the three corresponding SAs are activated to recover the three unary bits, which can be further decoded into two binary bits in the periphery.

The restore operation is achieved by isolating the SAs and closing the switch transistors in the horizontal direction. After charge sharing, the wordline is de-asserted to trap the data signal in the memory cell. The write operation is similar to the restore operation, except that the unary bits are decoded from the binary bits coming from outside the memory core and the SA states of the appropriate subbitlines are overwritten with new data.

The main advantage of Birk's method is the use of local reference generation, which is inherently immune to parameter imbalances in the chip. Parallel sensing is faster than serial sensing. The main disadvantages are the area overhead of the switch matrix and the reference generation circuit. This leads to the investigation of using even more voltage levels, which can further increase the storage density.

### 2.3.2 A Series of Test Chips

Birk's 4-level scheme was implemented in ML3 by Albert Chan [1]. Then Yunan Xiang followed the same scheme of Birk and expanded it to a variable-capacity MLDRAM, called ML5, which can use two, three, four and six data signal levels to store 1, 1.5, 2 and 2.5 bits in each memory cell, respectively. ML5 was developed to investigate the possibility of increasing the numbers of voltage level in Birk's parallel sensing scheme. In order to sense more voltage levels, we need to expand the reference signal generation patterns in Table 2.2 into those shown in Table 2.3 [27][26].

At the same time, the dimensions of the subbitline array were expanded from

|  |  | Subbitline Section |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  | A | B | C | D | E |
| Subbitline Pair | 1 | VSS | VSS | ½ VDD | VSS | VDD |
|  | 2 | VSS | VSS | VDD | VDD | VDD |
|  | 3 | ½ VDD | ½ VDD | ½ VDD | ½ VDD | ½ VDD |
|  | 4 | VSS | VSS | VSS | VDD | VDD |
|  | 5 | VSS | VDD | ½ VDD | VDD | VDD |

Table 2.3: Expanded Reference Generation Table

3-by-3 to 5-by-5. The control signal sequence for ML5 is similar to that of ML3. The operation mode can be changed by small changes in the control signals. In the test chip, these control signals were mostly supplied from the external tester.

Silicon verification for a test chip is usually a challenge, especially when probing internal signals, which are generally non-accessible from the output pins. In ML6, Ung included built-in probes in her test chip to permit measurements of the voltage inside a real MLDRAM chip.

In the following chapters, another new MLDRAM scheme with hierarchical bitlines and serial sensing will be introduced. Layouts of the most critical aspects of the design were completed to demonstrate feasibility.

# Chapter 3

# A MLDRAM with Hierarchical Bitlines and Serial Sensing

A MLDRAM with hierarchical bitlines and serial sensing was proposed recently by J. H. Tapia et al. at the University of Alberta [3]. The new design attempts to reduce the relatively large area overhead in Birk's scheme, which comes from the complex switch matrix and multiple SAs required for parallel sensing. Such area overhead can outweigh the area reduction obtained by storing multilevel in one memory cell.

In order to reduce the area overhead of multiple SAs, serial sensing is used instead and only one SA is required, which is shared by all the subbitlines connected to the same bitline. This one SA per bitline structure also gives us the possibility to stagger adjacent SAs on opposite sides of the cell array. This arrangement also frees up more area to implement more complex SAs, such as input offset compensating SAs that are able to cancel a major source of imbalance in the conventional SA. The new MLDRAM design uses a hierarchical and multidivided bitline to reduce the capacitance ratio of the bitline to the memory cell and thereby increase the noise margins in the proposed MLDRAM.

In the serial 3-level MLDRAM, three equally-spaced voltage levels are used to represent the binary data bits and two reference voltage levels are used to sense the data voltage. The cell and reference voltages of the 3-level MLDRAM are shown in Figure 3.1. Two memory cells are used together to code each 3-bit binary data word, so that each cell can store 1.5 bits. The same strategy can be generalized to encode other fractional bits per cell (eg. six signal levels encodes 2.5 bits in a cell). The fractional bits are recovered as whole bits by encoding two more cells together.

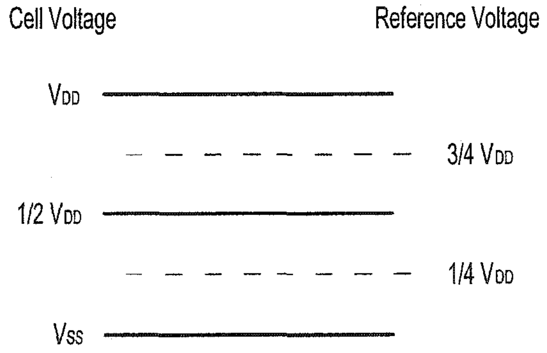


Figure 3.1: 3-Level MLDRAM

## 3.1 Structure and Operation of the Serial MLDRAM

### 3.1.1 Structure of the Serial MLDRAM

The block diagram of the new MLDRAM is given in Figure 3.2. The 3-bit binary input code word is first converted into two 2-bit unary codes, then each of these two unary codes is transferred to one side of a pair of cell arrays, which we will call a buddy cell array pair. The bitline between the buddy cell arrays can be connected together in the horizontal direction by the switches between them. During the sensing operation, one buddy array can hold the first bit of the unary code in its SA, while the second array continues to sense and recover the second bit. During the data output phase, the unary code pair is first transferred over the internal data bus to the decoder and then converted into binary bits.

In order to reduce the ratio of the bitline to cell capacitance, a multidivided and hierarchical bitline is proposed for the new MLDRAM [12]. In the memory array, subbitlines run alongside (and probably beneath) a long unbroken bitline. For each

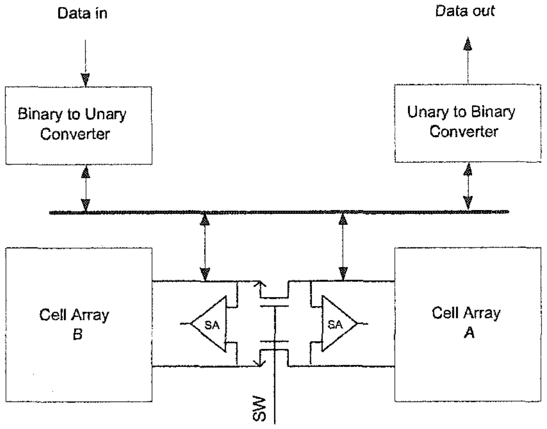


Figure 3.2: Block Diagram Showing Two Buddy Arrays

subbitline, one so-called ENSBL transistor switch is used to control the connection between the subbitline and the bitline. The subbitlines associated with the same bitline can be concatenated end-to-end using so-called SW transistors. The subbitline can also be precharged to a bus carrying $V_{DD}$ or $V_{SS}$, through a PRESBL transistor. Only one SA is associated with the bitline pair at one side of the memory array. All the subbitlines connected with the same bitline pair share this one SA. In the proposed MLDRAM, the multi-divided bitline consists of 16 subbitlines, and each subbitline section contains 32 wordlines, 2 dummy wordlines and 2 reference wordlines. A schematic of the proposed hierarchical bitline is shown in Figure 3.3.

A schematic of the subbitline is shown in Figure 3.4. Each data or reference cell is first connected to a short subbitline through the cell access transistor and then on to a long bitline through the ENSBL transistors.

Within such a hierarchical bitline configuration, the previous mentioned ratio of the capacitance of the bitline and cell needs to be generalized to the ratio of the capacitance of the external circuit and the cell. The capacitance of the external circuit, $C_{ext}$, thus includes several components, which is different from the situation in the conventional DRAM. The major components are :

(1) The parasitic capacitance of the bitline, $C_{BL\_p}$, which contains the parallel

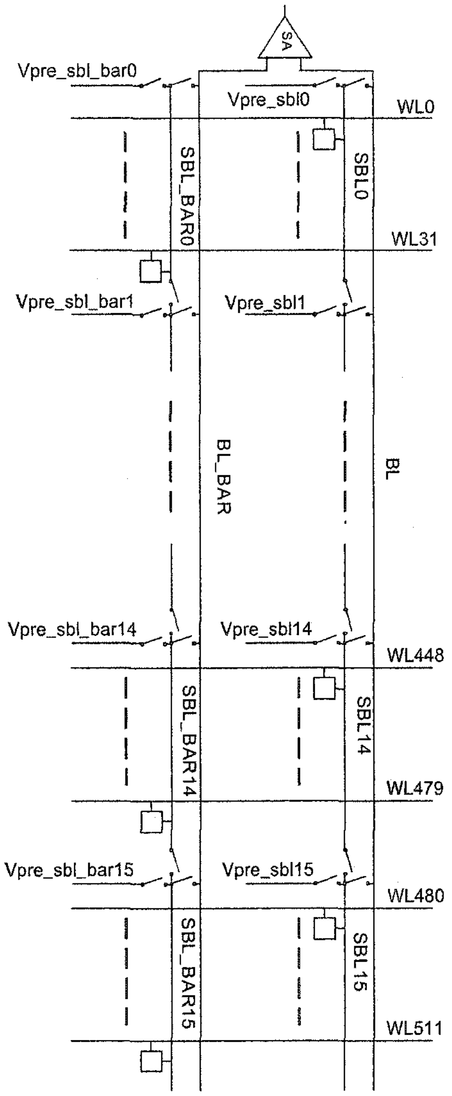


Figure 3.3: Schematic Configuration of the Bitline in the New Design

plate and fringe capacitance between the bitline and its nearby conductors (adjoining bitlines, underlying subbitlines, wordline straps and control signal wires). For a given manufacturing technology, this capacitance is linearly proportional to the length of the bitline and inversely proportional to the distance between the bitline and other conductors in its vicinity. In the new MLDRAM, more area has to be used to implement the switch transistors between the adjacent subbitlines, and also the required reference cells will increase the area of the cell array. So the bitline,

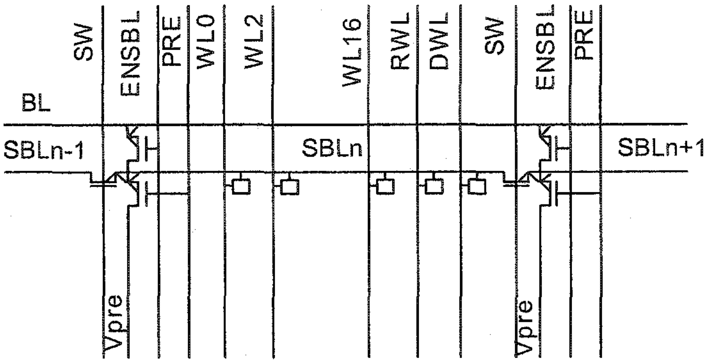


Figure 3.4: Schematic of the Subbitline in the New Design

which is parallel with all those subbitlines, will be longer than that in a conventional DRAM, assuming the same number of memory cells.

(2) The junction capacitance of the ENSBL transistors connected with the bitline, $C_{BL\_j}$. Since the number of the ENSBL transistors is much smaller than the number of cell access transistors connected to a bitline in a conventional DRAM, the junction capacitance $C_{BL\_j}$ will be decreased.

(3) The parasitic capacitance of the subbitline, $C_{SBL\_p}$, which is proportional to the length of the subbitline and inversely proportional to the distance between the subbitline and other conductors in its vicinity assuming a given manufacturing technology.

(4) The junction capacitance of the cell access transistors connected with the subbitline, $C_{SBL\_j}$. In the new multidivided and hierarchical bitline scheme, the number of the cell access transistors is much smaller (eg. 16 or 32 instead of 256 or 512) than that in the conventional DRAM, so this junction capacitance will be decreased.

The external capacitance is the sum of all of the above capacitances. The total capacitance of the external circuits can be minimized by optimizing the number of the subbitlines and the length of each subbitline.

### 3.1.2 Operation of the Serial MLDRAM

In the proposed MLDRAM, charge sharing is used to accurately generate the various required analog voltages, which can be written into both the data and reference memory cells. As shown in Figure 3.1, three analog voltages are required to represent the three possible two-bit unary codewords, 00, 01 and 11. If each unary bit, 1 or 0, is encoded by a voltage of, $V_{DD}$ or $V_{SS}$ respectively, on two separate identical subbitlines, then after closing the switch transistor between them and causing charge sharing, three possible voltages, $V_{DD}$, $\frac{V_{DD}}{2}$ and $V_{SS}$, can be formed. Since the capacitance of the subbitline can change a little bit due to inevitable process variations, more subbitlines could be used to generate a more accurate signal voltage by benefiting from averaging.

In the new MLDRAM, two reference voltages, $\frac{V_{DD}}{4}$ and $\frac{3V_{DD}}{4}$, are required. They are generated in parallel and the generation process is similar to the above data voltage generation method. After generation, the reference voltage is stored in the reference cell for the following sensing operation.

At the beginning of the sensing operation, the data and reference signals are dumped at the same time onto complementary sides of a bitline and subbitline pair. The resulting differential signal is attenuated with the subbitline first, then with the bitline. Because the reference cell is identical in capacitance to the data cell, the capacitance of the bitline and its complementary must be matched. The noise injection should also be identical because the data and reference cell are accessed at the same time.

After the sensing of the first unary bit, the switch transistor between the bitline and its corresponding buddy bitline in the buddy array is closed. The first bit is then transferred to the buddy array and the buddy SA can latch the first bit. Then the switch transistor is opened, and the data array continues with the sensing of the second unary bit.

At the end of the second sensing operation, the two unary bits are transferred to the unary-to-binary decoder. They are decoded into three output binary bits along with another unary bit pair from another array. In the memory array, these two bits are also used to restore the cell voltage. The restore operation is similar to the write operation except that the two unary bits are supplied by the SAs in the data array and its buddy array instead of being supplied by the data input interface.

## 3.2 Schematic Simulation Results

Our simulations were carried out with 0.18-$\mu$m logic CMOS models because device models for a real DRAM process were unavailable to us. In this simulation, two unary bits are first written into the data cell, then read out with relaxed timing to verify the functionality of the proposed MLDRAM.

Three bitline pairs were used in each simulation for the three possible data voltages. The parameters are listed in Table 3.1. Since the bitline is placed in the higher metal layer above the subbitline metal layer, the parasitic capacitance of the bitline will be assumed to be smaller [3]. The waveforms of the three bitline pairs are superimposed as shown in Figure 3.5.

In this simulation, the first memory cell on the third subbitline of each bitline will be accessed. The three possible written signal levels are $V_{SS}$, $\frac{V_{DD}}{2}$ and $V_{DD}$, which represent '00', '01' and '11', from the highest to the lowest voltage. At the beginning of the simulation, all of the bitlines and subbitlines are precharged to $\frac{V_{DD}}{2}$, the ENSBL transistors for subbitlines 3, 4, 5 and 6 are closed, and the access transistor of the addressed cell and the reference cell of the subbitline 2 are closed. At 80 ns, the bitlines are connected to the data bus and the first bit of the two unary bits is written to the bitlines. At 200 ns, the ENSBL transistors for subbitlines 3, 4, 5 and 6 are opened and the first bit is trapped on subbitlines 3, 4, 5, 6 and the

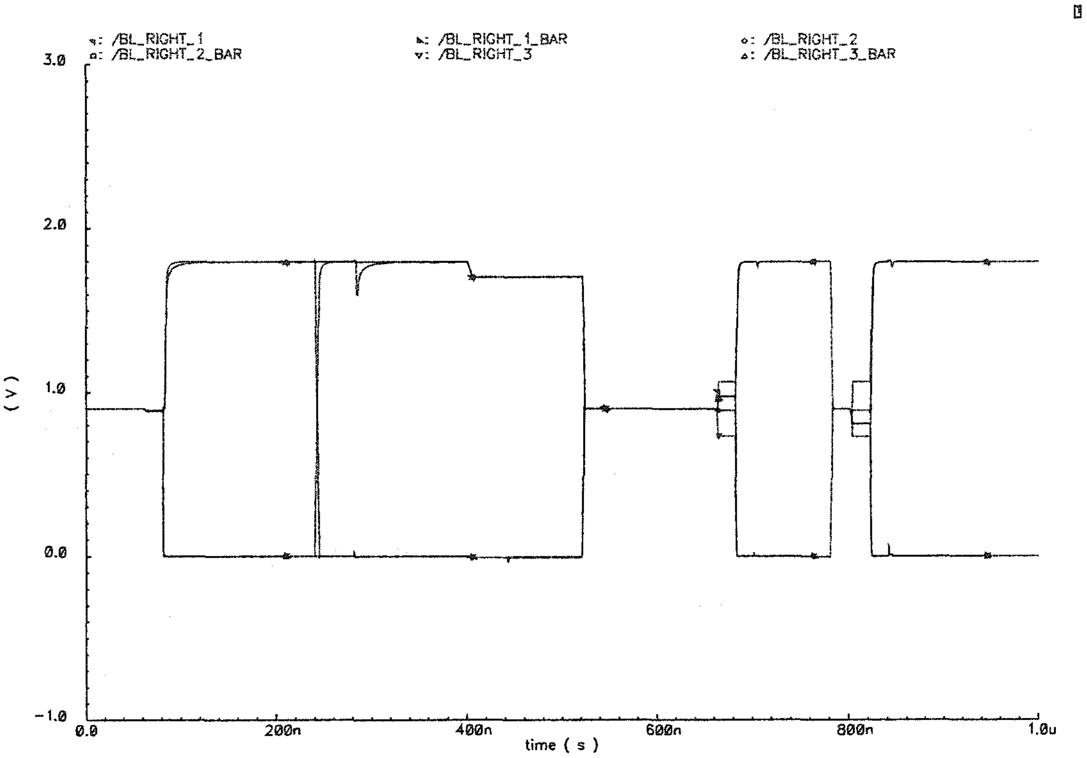


Figure 3.5: Waveforms of the New Design With Three Bitline Pairs

| Parameter | Value |
|---|---|
| Operating Voltage | 1.8 V |
| Data Cell Capacitance | 30 fF |
| Reference Cell Capacitance | 30 fF |
| Subbitline Capacitance | 5.39 fF |
| Subbitline to Bitline Capacitance | 1 fF |
| Bitline Capacitance in one subbitline | 165 aF |

Table 3.1: Simulation Parameters

addressed cell is connected to subbitline 3. At 240 ns, the second bit is written onto the bitline and then trapped on subbitlines 1, 2, 7, 8 and the reference cell on subbitline 2. At 440 ns, all the above mentioned subbitlines are shorted together to create the required analog data signals. At 480 ns, the address wordline is de-asserted, and the data signal is trapped in the memory cell. The waveforms of the control signals are shown in Figure 3.6.

The reference voltage will be generated on the complementary bitline, but neither the bitline nor the complementary bitline needs to be involved in the generation process. At 520 ns, the subbitlines are charged directly using the PRESBL_ transistors, not over the bitline. The subbitlines will be charged to various voltages based on their position in the bitline block, as illustrated in Figure 3.7.

At 560 ns, the odd SW_BAR transistors, except transistor 9, are closed to create two groups of subbitlines, each of them containing 8 subbitlines. After charge sharing on these two bitline groups, two reference voltages are created as depicted in Figure 3.7. The reference voltages are then trapped into four reference cells. Averaging the signals from four reference cells can generate a more accurate reference voltage than relying on one reference cell alone. At 660 ns, the cells holding the first reference voltage are connected with the first subbitline group and the complementary bitline, and generate the reference voltage for the first sensing step. At the same time, one copy of the data voltage in the reference cell on subbitline 2 is connected with the precharged bitline to create the attenuated data signal. At 680 ns, the SA is activated to sense the first unary bit which is then transferred to the SA of the buddy array. After the first sensing operation, the bitlines and subbitlines are precharged again to $\frac{V_{DD}}{2}$. The second sensing step uses the second copy of the data signal and the second reference voltage to get the second bit of the two unary bit. The waveforms of the control signals are shown in Figure 3.8.

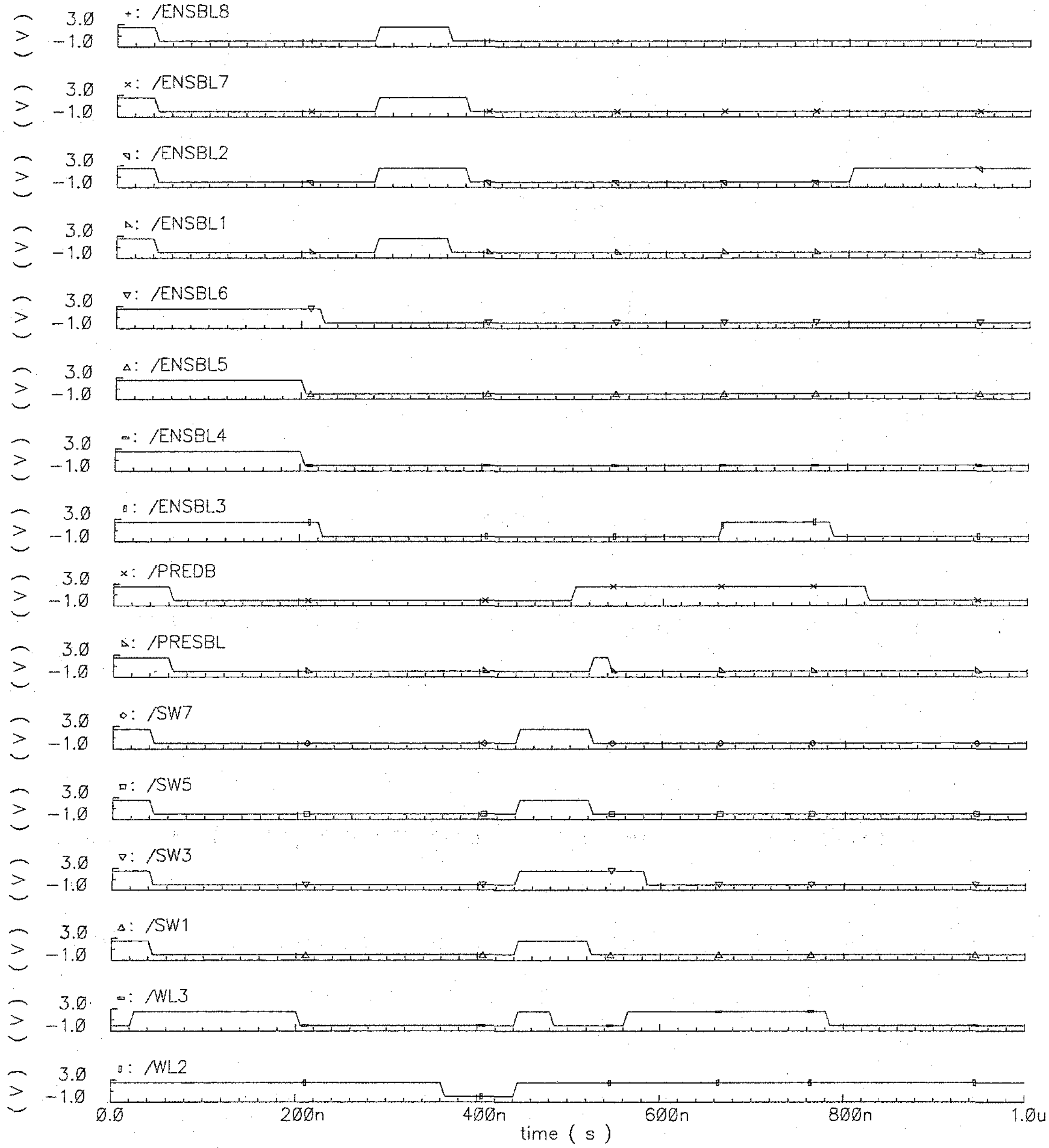


Figure 3.6: Waveforms of the Control Signals for Data Generation

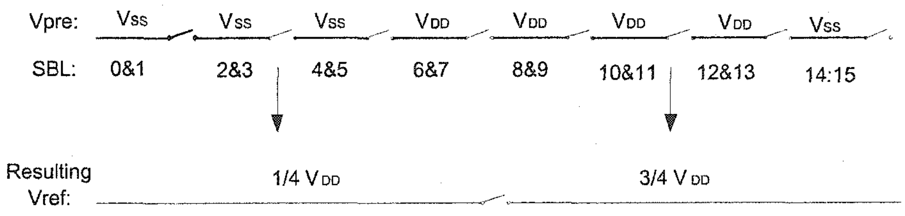


Figure 3.7: Reference Generation Using Charge Sharing

## 3.3 Discussion

Various methods are used to make the new MLDRAM work more reliably in the presence of noise, offset, charge injections and small device parameter variations.

(1) The addressed cell capacitance is balanced with that of the reference cell and noise injection due to wordline activation will be canceled by noise from the activation of the reference wordline. (2) The reference is generated with the averaging of voltages from multiple subbitlines with attached reference cells. This makes the reference voltage more accurate. (3) The capacitance of the bitline is decreased with the proposed multidivided and hierarchical bitline, which makes the noise margins bigger. But the capacitance used in the above simulation is based on many assumptions and on simplified hand calculations and further simulations with parasitic extraction from a real layout need to be done to verify and revise the assumed values. This work will be continued in the next chapter.

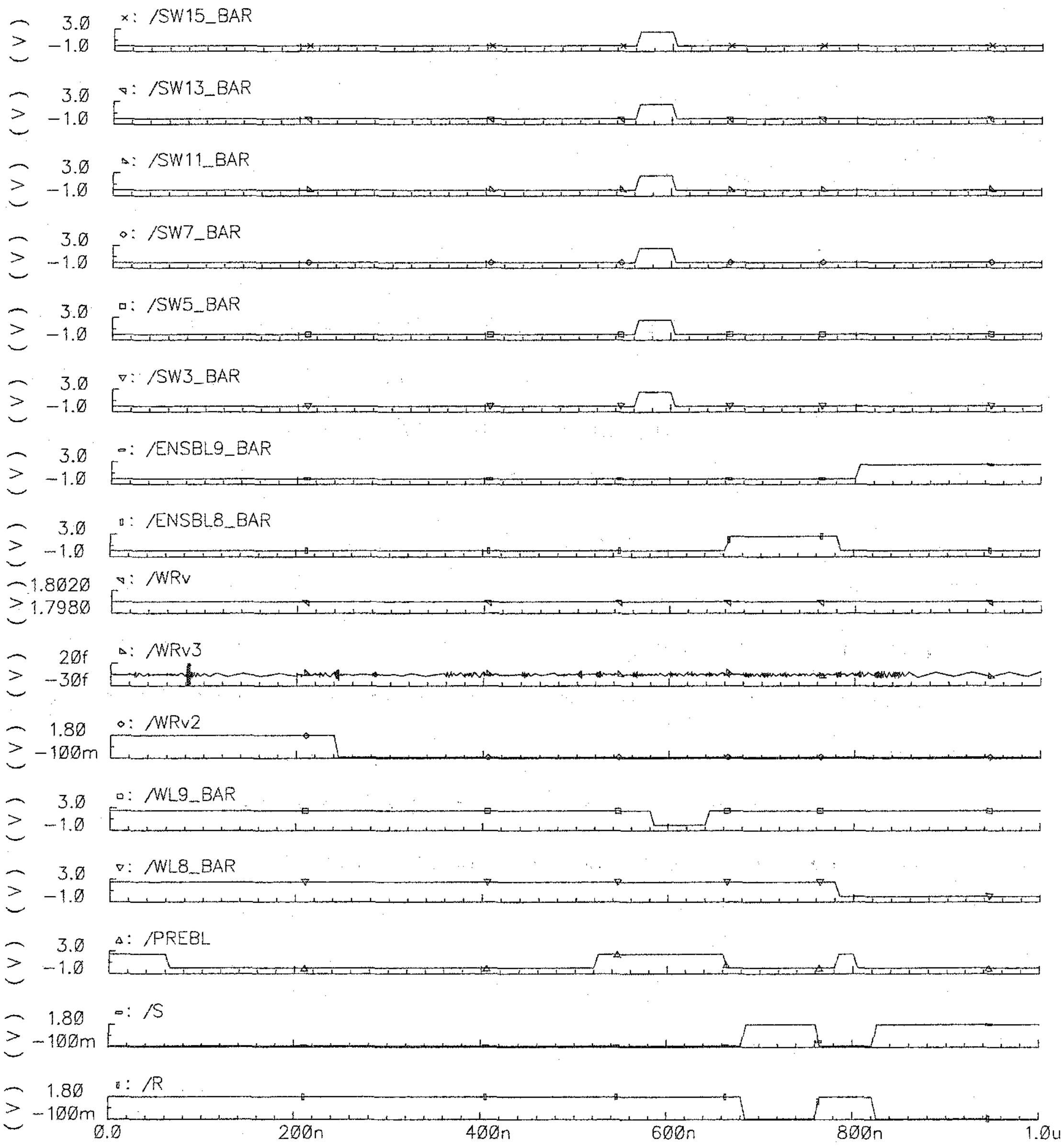


Figure 3.8: Waveforms of the Control Signals for Reference Generation and Sensing

# Chapter 4

# Implementation of the Serial MLDRAM

Memory design continues to be one of the greatest challenges for chip designers. In order to pack more memory cells into a smaller space on a die, the memory cell needs to be very compact. With scaling of the cell dimensions, the various parasitic capacitances and resistances are becoming increasingly important. It is more difficult to ensure the reliable operation of memory chips than ever before.

A memory chip can be divided into two parts, the control logic and the memory core, as depicted in Figure 4.1. The design methods for these two parts are quite different. The control logic comprises standard logic gates and drives various control signals to the memory core. Since the control logic only occupies a small part of the DRAM chip and does not need too tight layout constraints, it can be synthesized with standard cells to save design time and cost. The memory core must be compact, and has a regular architecture. Once the basic cell layout is finished, it can be repeated millions of times to form the memory array. Much experience and extensive simulations are required to produce a competitive cell layout and layouts for the pitch-limited peripheral circuits [2].

A typical memory design flow is shown in Figure 4.2 [2]. First, the schematic of the design is captured using the Cadence schematic composer tool, then the design is simulated at the schematic level. The schematic of the design needs to be modi-

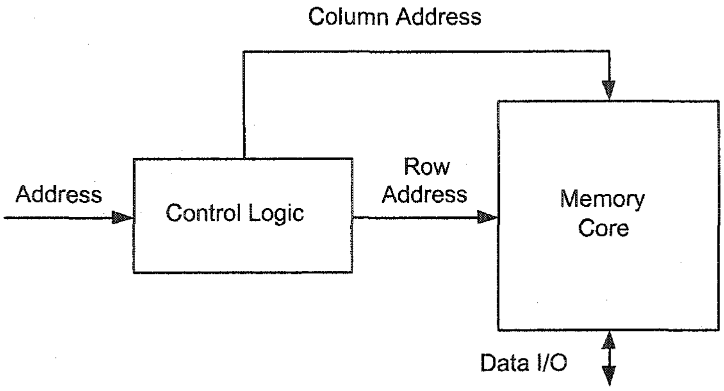


Figure 4.1: Simplified Block Diagram of the DRAM Chip

fied to get better performance, based on the simulation waveforms. Trial and error work as well as systematic worst-case condition simulation work typically needs to be done to get the best performance. After that, the layout is developed according to the schematic.

The MLDRAM layout work is divided into two parts: the memory core and the control logic. The layout of the memory cell is fully customized to satisfy the area and performance requirements. Design Rules Check (DRC), Electrical Rules Check (ERC) and Layout Versus Schematic check (LVS) can be run on the layout to ensure that it is safely manufacturable and implements the same function as the schematic design. Simulations on the extracted layout (that is, a simulation model with parasitic elements which is deduced from the layout) can be done to verify the function and performance of the circuit. The function of the control logic can be described using a hardware description language, like Verilog or VHDL, and then converted into a gate-level netlist using the synthesis tools.

After the floorplan of the chip is determined, the layout of the memory core and the control logic can be placed and routed over one or more iterations to get the optimized layout. The final optimized layout needs to be verified with DRC and LVS to ensure manufacturability and functionality.

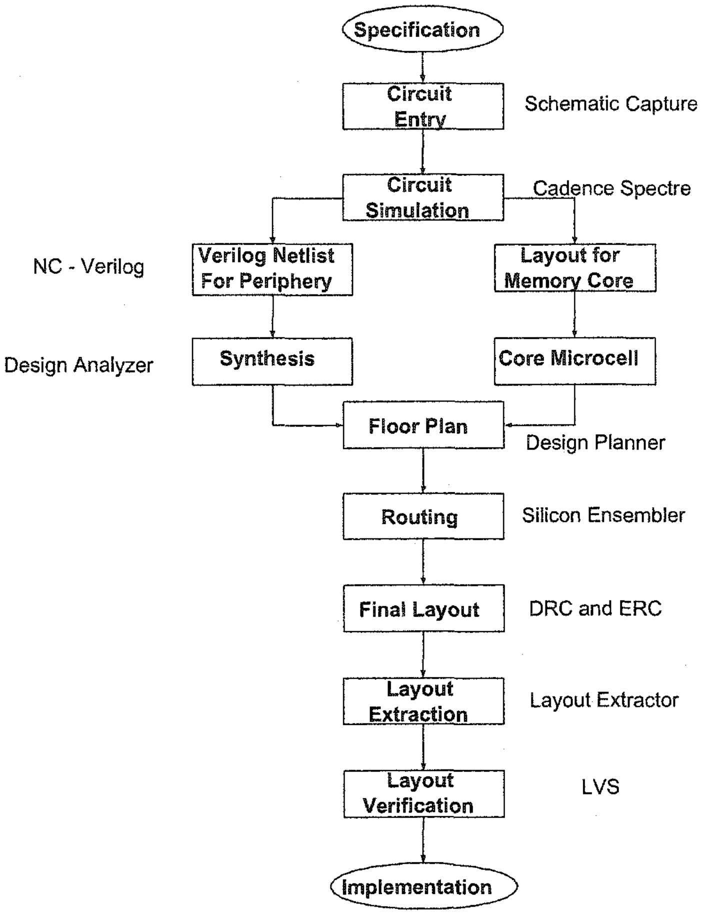


Figure 4.2: Memory Design Flow

The layout implementation of a memory chip is especially constrained by the process technology. In this thesis, the implementation uses Taiwan Semiconductor Manufacturing Corporation (TSMC)'s 0.18-$\mu$m CMOS mixed-signal process. The device models for a real DRAM process are not available for us, so we needed to use a logic process instead. Six metal layers are available in the process, but only the first three layers are used in the implementation of the new MLDRAM. This is consistent with recent DRAM practice, which only use three metal layers (presumably to keep the manufacturing cost low). In this chapter, the implementation of the memory core will be described in detail.

## 4.1 Implementation of the Memory Core

The memory core consists of the memory cell array and the pitch-matched SAs and wordline drivers. The basic element of the memory array is the 1T1C memory cell, which was laid out manually. Because the bitline is heavily capacitive, typically only 256 memory cells are connected with each bitline in most contemporary DRAMs [15]. This 256-cell-per-bitline configuration is therefore also used in our design. Other standard DRAM implementation techniques, such as twisted bitline and wordline strapping, are also considered in the design of the memory array.

### 4.1.1 Memory Cells

The memory cell uses MOSAID's embedded DRAM layout macros in 0.18-$\mu$m technology [25]. The cell access transistor is NMOS and the cell capacitor is implemented as the gate-substrate capacitor of another NMOS transistor, as shown in Figure 4.3. The polysilicon (poly) gate of the NMOS transistor is used as the storage node of the capacitor and the substrate connected with the source and drain nodes forms the common plate for the cell capacitor. The capacitance of a cell capacitor is determined by the gate oxide thickness and material (both are fixed in a given process) and the area of the cell capacitor layout, i.e. the area of the poly gate of the cell capacitor.

In TSMC's 0.18-$\mu$m CMOS technology, the NMOS gate-substrate capacitance per unit area is calculated as follows [22]:

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} = \frac{3.51345\, fF/\mu m}{4.08 \times 10^{-3} \mu m} = 8.6114\, fF/\mu m^2$$

where $\varepsilon_{ox}$ is the permittivity of the gate oxide and $t_{ox}$ is its thickness. The actual capacitance is a little bigger than this parallel plate capacitance value due to the additional fringe and coupling parasitic capacitances. In order to compare memory performance for different bitline to cell capacitance ratios, four kinds of memory

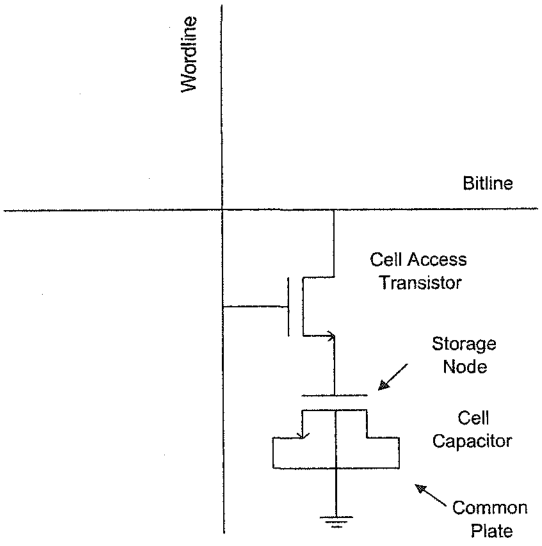


Figure 4.3: Schematic of the Memory Cell

cell with different cell capacitances ($35fF$, $45fF$, $55fF$ and $65fF$) were considered in the simulations. A top view of the memory cell layout is illustrated in Figure 4.4.

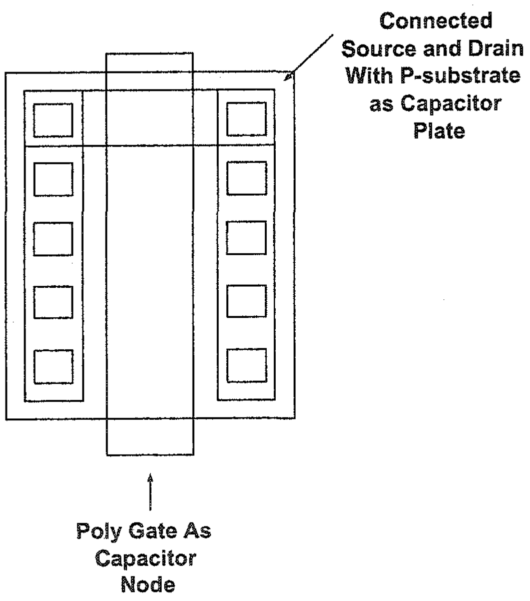

Connected Source and Drain With P-substrate as Capacitor Plate
Poly Gate As Capacitor Node

Figure 4.4: Top View of the Cell Capacitor Layout

The common plate and the p-type substrate are connected together to a back bias voltage of -1.0 V. The back biased substrate can ensure the cell capacitor being on all the time, even when the cell voltage is 0 V. So the cell capacitance is stabilized and maximized. Meanwhile, the substrate back bias increases the threshold voltage of the cell access transistors and thus decreases the sub-threshold leakage current of the cell access transistors. The back bias also decreases the junction leakage current from the storage node to the substrate via the source node of the access transistor. Thus the refresh period of the cells is increased.

### 4.1.2 Bitline and Wordline Pitches and the Feature Size

In a DRAM array, the parallel bitlines are located physically orthogonal to the parallel wordlines. The memory cells are located at half of the intersections of the bitlines and the wordlines. The physical area of the memory cell is determined by the bitline and wordline pitches. The bitline pitch is the sum of the bitline width and inter-bitline space, and the wordline pitch is the sum of the wordline width and inter-wordline space. Meanwhile, the width of the SA is constrained by the pitch of the bitline and the width of the wordline driver is constrained by the pitch of the wordline.

The minimum feature size, usually denoted by F in memories, is another important definition in DRAM design. It is the minimum realizable process dimension, which is typically equal to half of the bitline or wordline pitch. Alternatively, one could define F to be half of the average of the bitline and wordline pitches.

Within the folded bitline array, the area of the DRAM cell can be no smaller than $2F \times 4F$, as illustrated in Figure 4.5. The dashed box in the figure delineates the boundary of a memory cell. The width of the cell includes one-half bitline contact width, one wordline width, one capacitor width, one field poly width and one half poly space widths. The height of the cell contains two one-half field oxide widths and one active area width.

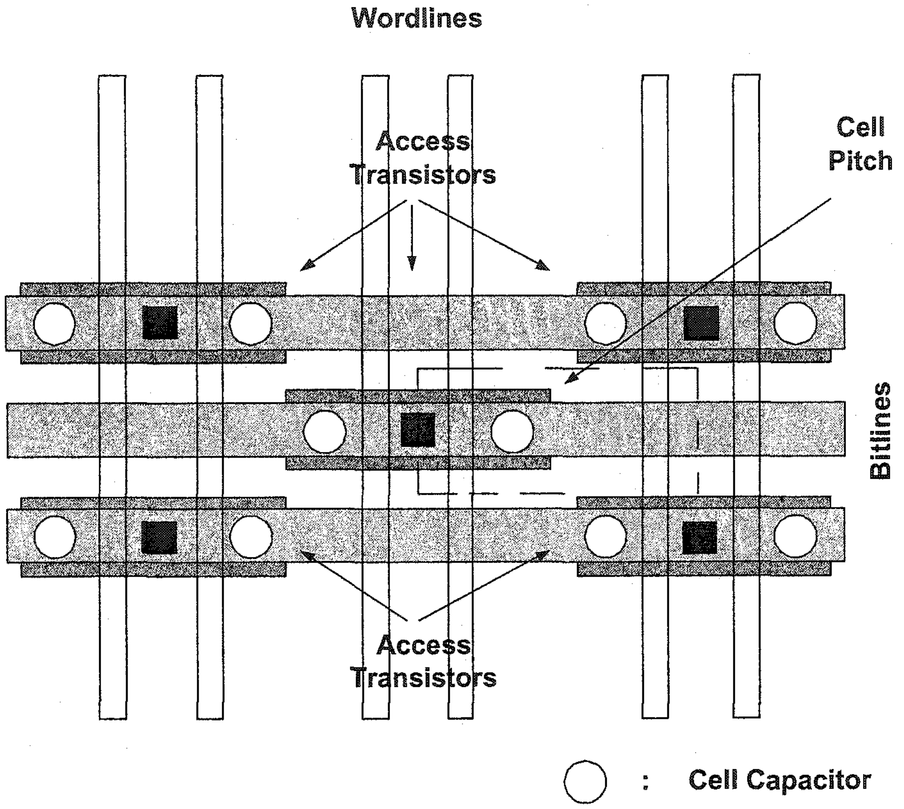


Figure 4.5: DRAM Cell Pitch

The minimum available cell area with folded bitlines ($8F^2$) is bigger than that of open bitlines (at least $6F^2$) [15]. The folded bitline arrangement thus gives us more area to implement the cell capacitor, whose value is critical to the performance of the DRAM. The disadvantage of folded bitlines is the reduced cell density, compared with open bitlines because only half of the intersections of the bitlines and wordlines are used for the memory cell, as we mentioned in Chapter 1.

In order to simulate the operating environment in a real DRAM array, the layout of the new design uses the minimum dimensions of all the major components in the memory array. Thus the bitline pitch is 0.560 $\mu$m and the wordline pitch is

0.910 $\mu$m. Since the memory cell (planar cells in a logic CMOS process) used in the simulation can not fit into such a small space, the cells for the inner column were implemented outside of the bitline pitch, as illustrated in Figure 4.6.

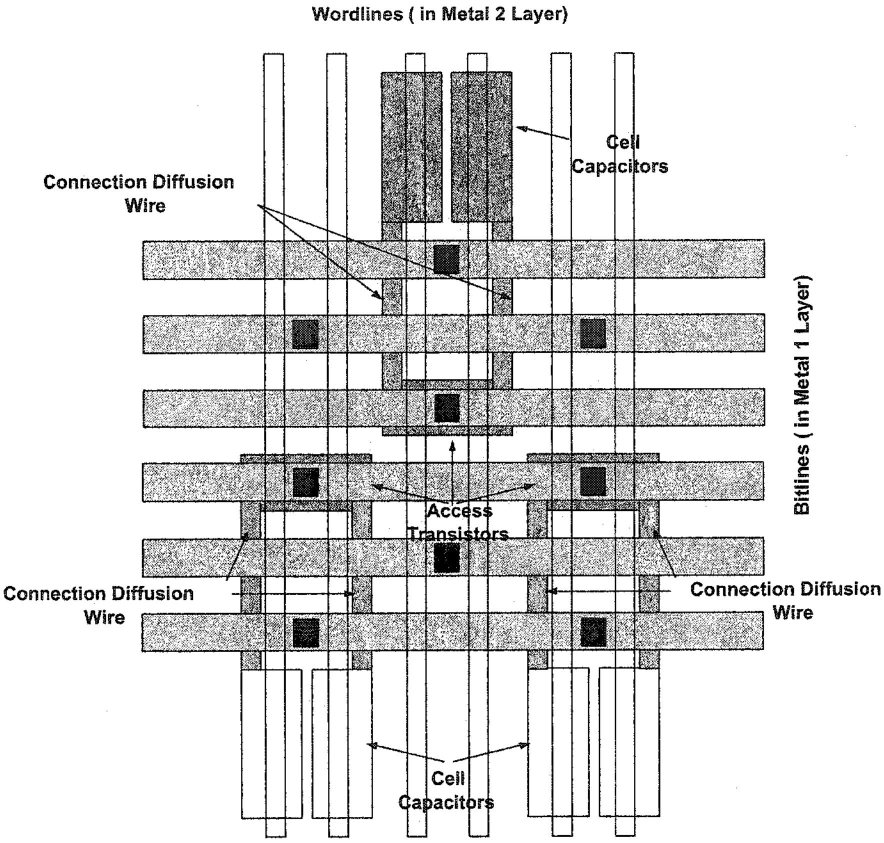

Wordlines ( in Metal 2 Layer)
Cell Capacitors
Connection Diffusion Wire
Bitlines ( in Metal 1 Layer)
Access Transistors
Connection Diffusion Wire
Connection Diffusion Wire
Cell Capacitors

Figure 4.6: Top View of the Memory Cell Placement

In this layout, the storage node of the cell capacitor is connected with the source of the cell access transistor through a diffusion wire. Although the resistivity of diffusion area is relatively large, the diffusion wire is very short and the resulting resistance will not be too big (about 300 Ω).

In this way, the capacitance of the bitline should more closely approximate the value in a real DRAM array. Note that the capacitance between adjacent bitlines accounts for most of the bitline capacitance and the bitline capacitances are inversely proportional to the spacing between them.

### 4.1.3 Twisted Bitlines

In the memory array, the capacitance (parallel plate and fringe) between the bitlines is inversely proportional to the pitch of the bitlines. As the DRAM cell scales down in size, the bitline pitch is becoming smaller and the capacitance between adjacent bitlines is increasing. At the same time, the operating voltage is decreasing to avoid dielectric breakdown. The noise margin in MLDRAM, which is $\frac{1}{2*(N-1)}$ of the operating voltage, is also becoming smaller than before. So it is especially important to eliminate or control all sources of noise injection that might affect the weak data and reference signal voltage.

Bitline twisting is a layout technique for controlling the noise caused by bitline-to-bitline capacitance of the memory array within the folded bitline structure. It is shown in Figure 4.7.

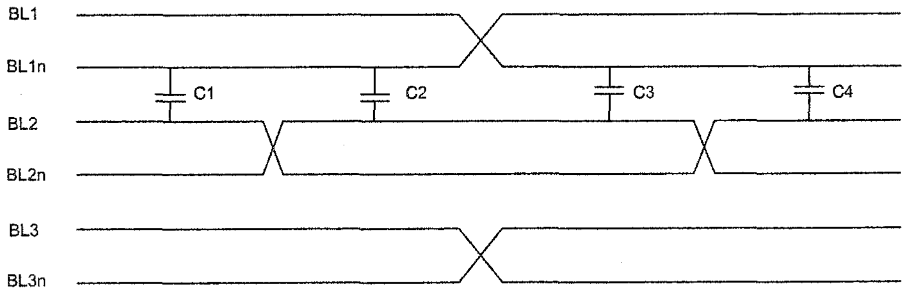


Figure 4.7: Bitline Twisting

Consider bitlines $BL_1$, $BL_{1n}$, $BL_2$ and $BL_{2n}$ as an example. The capacitance between bitlines $BL_{1n}$ and $BL_2$ is C1, the capacitance between bitlines $BL_{1n}$ and $BL_{2n}$ is C2, the capacitance between bitlines $BL_1$ and $BL_{2n}$ is C3 and the capacitance between bitlines $BL_1$ and $BL_2$ is C4. C1 and C2 balance out any coupling noise induced by $BL_{1n}$ into memory column 2. Similarly, C3 and C4 balance out any noise coupled from $BL_1$ into column 2. C1 and C4 balance out any noise coupled from $BL_2$ to column 1. Finally, C2 and C3 balance out any noise coupled from $BL_{2n}$ to column 1.

A possible layout for the bitline twist is illustrated in Figure 4.8. Vias between different metal layers are used in the implementation of bitline twist. The metals used for the bitline twist, such as tungsten, is more resistive than the bitline layer metal (typically aluminum in DRAM), so the twist connection will increase the resistance of the bitline. In order to keep the resistance increase as small as possible, two vias were used in parallel.

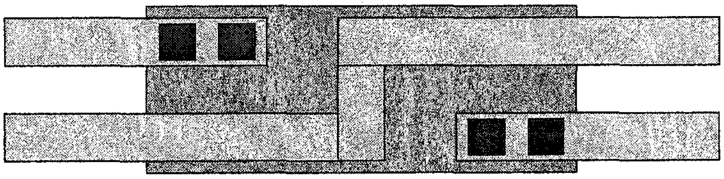


Figure 4.8: Top View of Bitline Twist Layout

### 4.1.4 Strapped Wordlines

The wordline in the memory array is highly capacitive due to the large number of memory cell access transistor gates connected to it. The resistance of the wordline is also large due to the relatively high resistivity of the polysilicon layer. Without special treatment, the rise time of a purely poly wordline is likely to be slow. The effective resistance of the wordline can be reduced greatly by strapping the poly wordline with a parallel metal wire. Three different wordline strapping arrangements are illustrated in the Figure 4.9. In our layout, the first scheme is selected due to its simple structure.

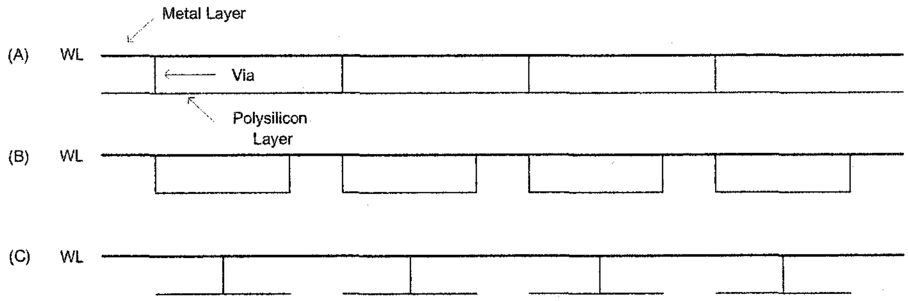


Figure 4.9: Wordline Strap

### 4.1.5 The Interconnection Block

An interconnection block is located at one end of each subbitline pair. The block includes three connection transistors. The first transistor is used to connect the subbitline to the bitline. The second is used to connect end-to-end adjacent subbitlines. The third one is used to precharge the subbitline. The schematic of the interconnection block is illustrated in Figure 4.10.

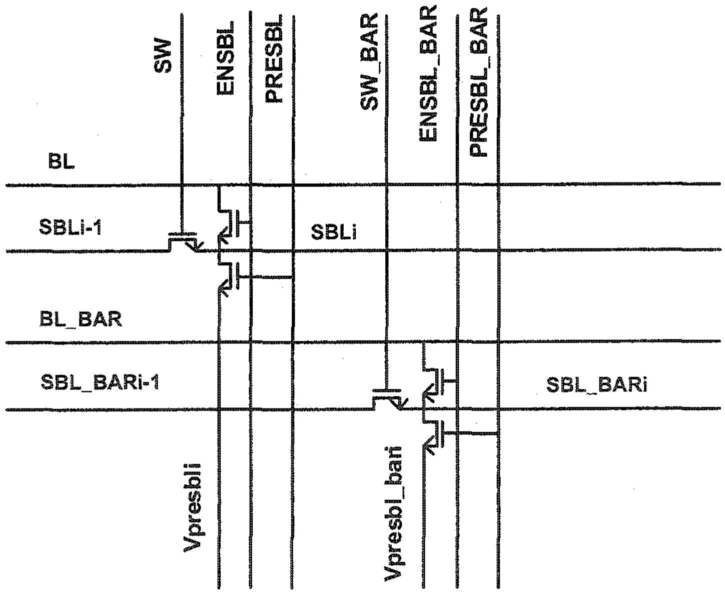


Figure 4.10: Schematic of the Interconnection Block

In order to save area, the interconnection blocks for one bitline pair are laid out together. The interconnection block must be compact because it is repeated for each subbitline pair. Any area overhead is amplified by the number of subbitline sections.

The layout of the interconnection block is illustrated in Figure 4.11. In this layout, the bitlines are placed in the metal3 layer, and the subbitlines are placed in the metal1 layer. In order to save area, the interconnection blocks for the same bitline pair are staggered with two bypasses in the metal2 layer. The precharge voltage is carried by a short wide poly wire, thus the resistance of the poly wire is kept small. Designed in this way, the layout of the interconnection block is highly compact.

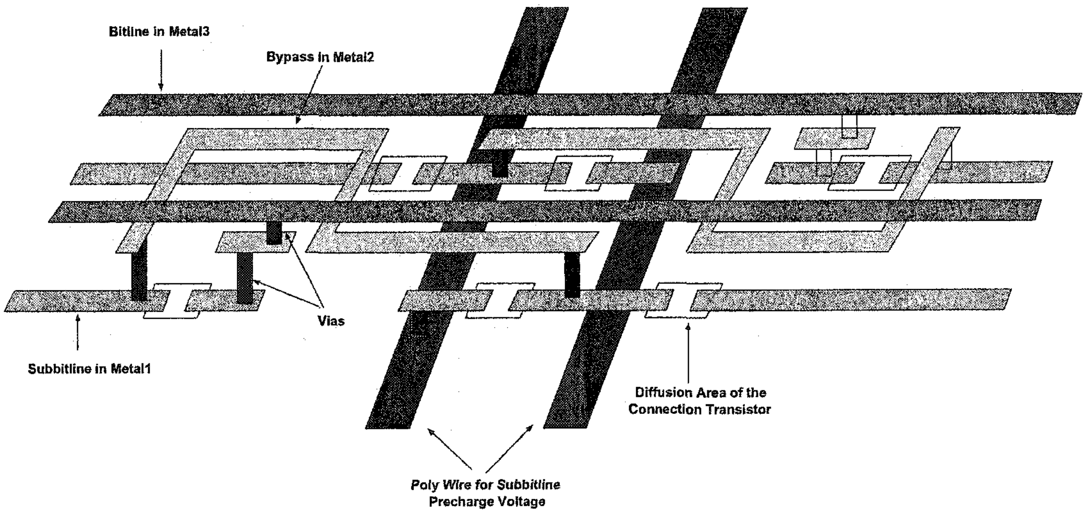
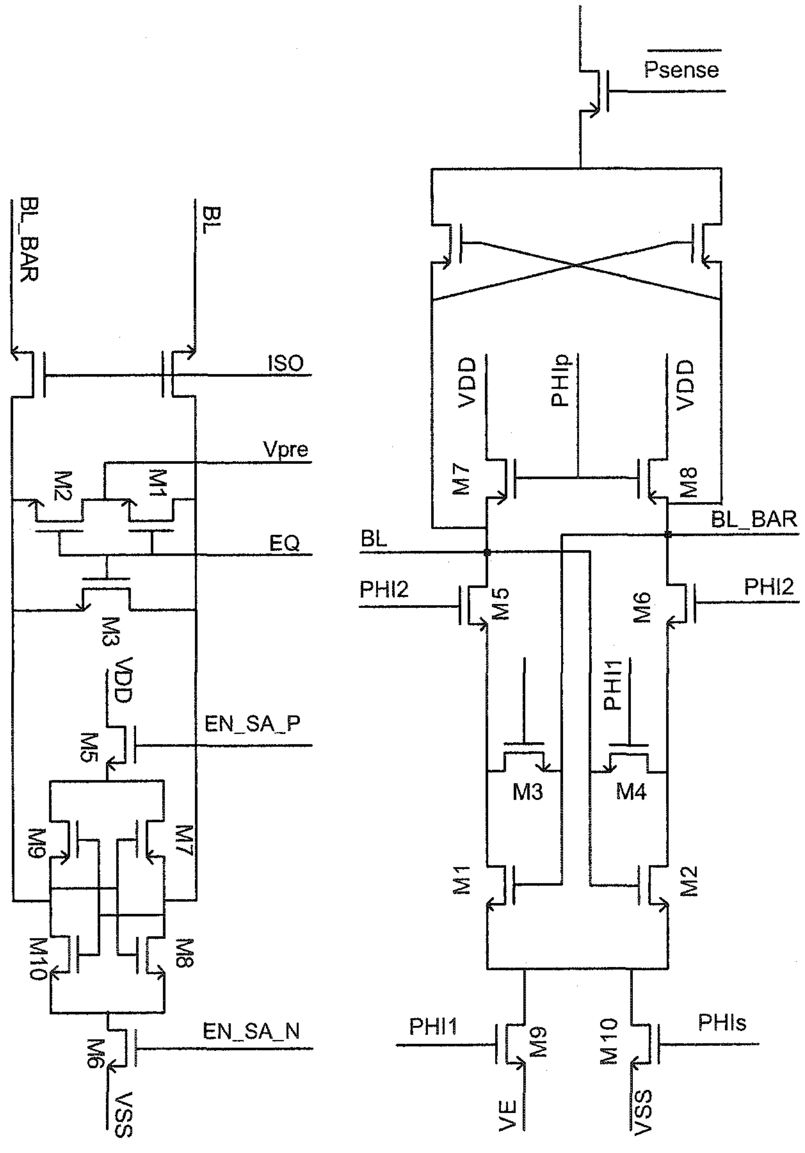

Bitline in Metal3
Bypass in Metal2
Vias
Subbitline in Metal1
Diffusion Area of the Connection Transistor
Poly Wire for Subbitline Precharge Voltage

Figure 4.11: 3-D View of the Interconnection Block Layout

### 4.1.6 Sense Amplifiers

Two different SA schemes were used in the simulations. One was the common cross-coupled inverter latch SA; the second one was a SA with input offset cancelation. They were both introduced in Chapter 1. The width of the SAs is minimized to fit into the bitline pitch. The layout of these SAs was completed by S. Tian [20]. The schematics of these two SAs are shown for comparison in Figure 4.12.



(a) The Common SA

(b) The SA With Input Noise Cancelation

Figure 4.12: Schematics of Two SAs

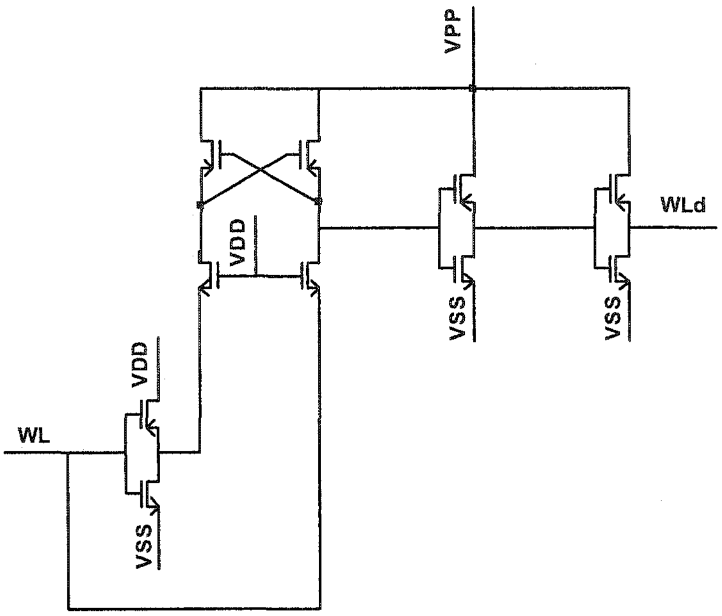


Figure 4.13: Schematic of the Wordline Driver

### 4.1.7 The Wordline Driver

Since the wordlines in DRAM are both highly capacitive and resistive, in order to decrease the RC delay of the wordline and to improve the read-write access time, low-impedance wordline drivers are employed and the poly wordlines are shunted with metal wordlines. In addition, the wordline drivers are required to drive the wordline signal to at least one threshold voltage above $V_{DD}$, so that a full $V_{DD}$ signal can be passed without attenuation from the bitline to the cell storage node. This technique is called wordline boosting [15].

There are many different kinds of wordline drivers. In our design, we used the proven wordline driver circuits from ATMOS Corporation, as shown in Figure 4.13. Wordline boosting is realized for these CMOS drivers by simply changing the power supply voltage to the boosted power supply voltage, denoted by $V_{pp}$.

# Chapter 5

# Simulation and Verification

In order to verify the correct functionality of the designed circuits, various simulations were run to detect design errors and to verify the correctness of the design fixes. The simulations were carried out with abstract models of the circuit, and then we determined the responses of the designed circuit to various input stimuli. Following typical custom IC design practice, simulations were carried out using two kinds of models: schematic and extracted layout. Since the physical layout must be acceptable for manufacturing, it must comply with the design rules set by the process engineers.

In this chapter, the basic concepts of circuit simulation are introduced at first. Then the simulations of the new MLDRAM design are described in detail. A capacitance model is developed for the cell and hierarchical bitline to predict the various capacitances, and hence the cell signals, in the new MLDRAM for different configurations. An optimized configuration of the proposed MLDRAM is developed based on the above simulation and analysis.

## 5.1 DRC and LVS

In IC design, the schematics of the circuits are first entered into a CAD system (i.e. captured) assuming a given technology. Schematic simulation models may include estimated models for parasitic properties, such as resistance and capacitance. Schematic simulations are carried out to verify that the schematic design meets the specifications for the circuit. Then the physical layout of the circuit is created using the various available physical layers. All the special requirements for the power and ground, as well as timing and noise cancelation, were considered in the layout design. Since the physical layout of the circuits must be manufacturable and hence must conform to layout design rules, DRC was run to ensure there were no violations of design rules for the process.

After the layout was shown to pass the DRC, it was "extracted" to get the topology of the corresponding electrical circuit. LVS was then run to verify that the circuit extracted from the layout matches the intended circuit schematic. This first version of the extracted layout does not consider the parasitic resistances, capacitances and inductances. In order to get a more accurate physical description of the layout, an extraction with "parasitics" is required for more accurate simulations. If the post-layout simulation with parasitics still meets the circuit specifications, then the design can be considered ready for manufacture. Otherwise, the design has to be modified and improved until all these specifications are satisfied or found to be unsatisfiable.

One problem that was encountered in previous MLDRAM work was the automatic extraction of the cell capacitor. In Yunan Xiang and Sue Ann Ung's simulations, the layout of this transistor has no source and drain nodes so the Cadence layout extractor could not extract the transistor from this position in the layout. Since the extracted layout could not be modified by Cadence, the simulation with extracted layout could not be run. The capacitance of the cell and the bitline could therefore only be simulated at the schematic level with the manually estimated and explicitly included models of the parasitic capacitances. Thus it was very difficult to predict the behavior of the designed circuit precisely.

The cell capacitance modeling problem was solved in the new layout. As mentioned in Chapter 4, the storage node of the cell capacitor was implemented with the gate capacitance of the transistor. In the new layout, the source and drain nodes were added and connected with the substrate. The capacitor could then be automatically extracted. The new layout is compared with the previous one and shown in Figure 5.1.

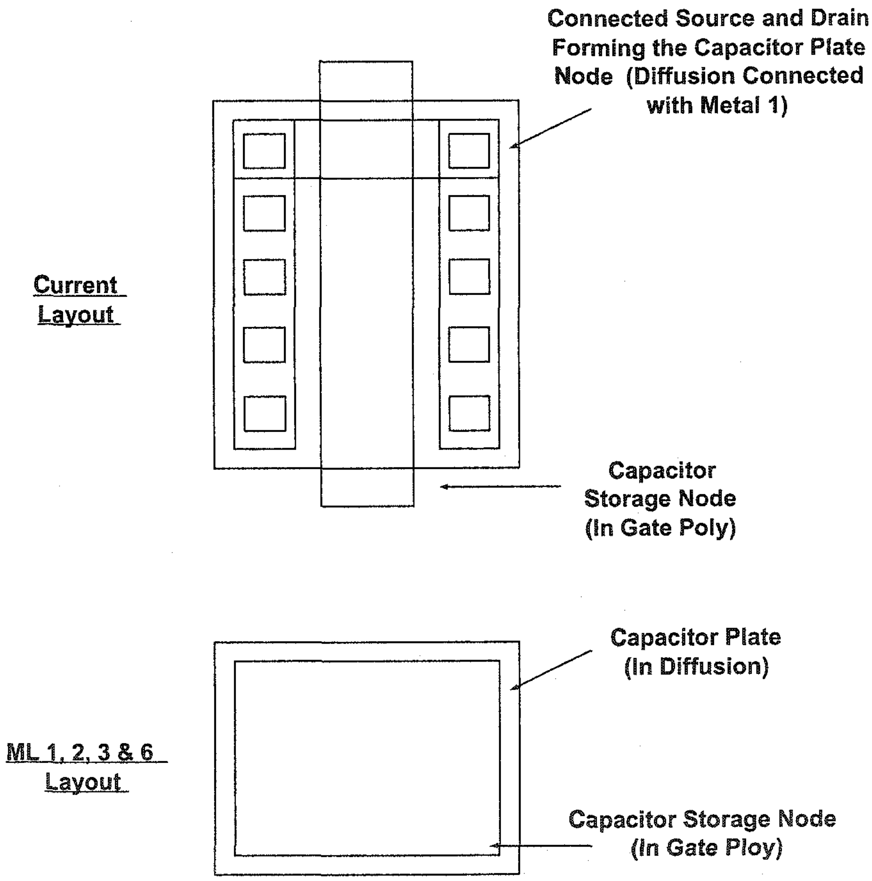

Connected Source and Drain Forming the Capacitor Plate Node (Diffusion Connected with Metal 1)
Current Layout
Capacitor Storage Node (In Gate Poly)
Capacitor Plate (In Diffusion)
ML 1, 2, 3 & 6 Layout
Capacitor Storage Node (In Gate Ploy)

Figure 5.1: Original and Improved Cell Capacitor Layouts

By inspecting the netlist file of the extracted layout, we could analyze the different components of the bitline parasitic capacitance. An accurate model of the design trade-offs could then be formed to get the optimized configuration of the new MLDRAM.

## 5.2 The SPICE Circuit Model

SPICE (Simulation Program with Integrated Circuit Emphasis) is an analog circuit simulator that is widely used for IC and printed circuit board design [16]. The original SPICE program was developed at the University of California at Berkeley in the early 1970's. SPICE uses analytical and empirical models for the different components used in circuit design. It can calculate the analog response of a designed circuit to any arbitrary analog input stimulus. So the behavior of a possible circuit design can be predicted in SPICE and verified before the design is submitted for manufacturing. An accurate model with reasonable calculation effort is very important for modern IC and circuit design due to the complexity of the circuits.

The Cadence IC CAD environment includes a SPICE-like circuit simulator called SPECTRE [24]. SPECTRE includes several built-in MOSFET models, including BSIM3v3 (Berkeley Short-Channel IGFET Model 3 version 3). BSIM3v3 has been widely adopted as an industry standard model for deep-submicron MOSFET transistors [16]. In the BSIM3v3 MOSFET transistor model, over 200 parameters are used to get an accurate and effective description of the MOSFET transistor. Because transistors with different gate dimensions differ greatly in their behaviors, different BSIM3v3 models are used for different ranges of the gate sizes, LMIN, LMAX, WMIN and WMAX. The parameters for a typical BSIM3v3 model of TSMC 0.18-$\mu m$ CMOS are given in Appendix A.

## 5.3 Simulation

The periphery of the test chip is composed of standard cells from the standard cell library provided by CMC, and no schematics were provided for these standard cells. Therefore, the simulation of the periphery could only use the layout-extracted circuits for the basic gates.

For the memory core, schematic simulations were run first. Because the bitline capacitance is determined by the process technology and the physical dimensions and spacing in the layout, it was estimated manually to be 270 fF and the capacitance of the subbitline was estimated to be 10 fF. The performance of the DRAM highly depends on the parasitic capacitance in the memory core. In order to simulate the real capacitance environment in the DRAM array, the layout was extracted with the parasitic capacitance. The same stimulus was applied to it and a more accurate simulation results were obtained.

The bitline and subbitline capacitances determine the signal voltages that arise during the sensing operation. We divided the parasitic capacitance of the bitline into several components and analyzed them with a mathematical model. As the result, an optimized configuration was proposed for the MLDRAM.

### 5.3.1 Schematic Simulation

In the new test layout of the proposed MLDRAM, we use three pairs of bitlines. The top and bottom bitline pairs are only used to simulate the real capacitance environment for the middle bitline pair. Due to the large area of the memory cell layout, we can not make memory cells for these two bitline pairs and thus no memory cells are connected with their associated subbitlines. The middle pair of bitlines are the only working bitlines and they have 16 subbitlines each. Each subbitline in turn connects to 16 memory cells, one reference cell and one dummy cell. All of these cells have the same cell capacitance. Dummy cells are used at the outside edge of the memory array so that the used (interior) cells all behave the same way. The memory cell uses the structure we proposed in Chapter 4.

The memory cells connected to each bitline were numbered as follows. If the number of the cell is converted into a nine-bit binary value, the first four bits select the subbitline to which the cell is connected, and the last five bits select the position of the cell on the subbitline. All the odd numbered cells are connected to the true bitline and all the even numbered cells are connected to the complementary bitline. The address numbering method is shown in Figure 5.2 for the example of the 65th cell on a bitline.

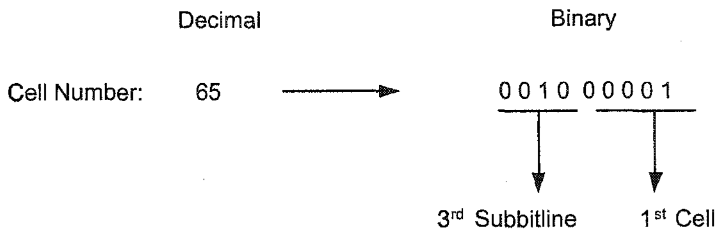


Figure 5.2: Address Numbering Convention

In this simulation, the data is written into the 65th memory cell, which is connected to subbitline 3. The second subbitline and its associated reference cell are used to generate the data signal along with the data cell and its associated subbitlines (two cells and two subbitlines). The reference generation uses the reference cells connected with the complementary subbitline 4 and 13 to store the two reference signals ($\frac{1}{4}V_{DD}$ and $\frac{3}{4}V_{DD}$). The reference cells are carefully selected on the complementary bitline and their associated subbitlines are also far from the subbitlines to which the data cells are connected. The simplified simulation schematic is shown in Figure 5.3. The schematic of one subbitline block is shown in Figure 5.4.

The simulation waveform for storing and then sensing '11' with the assumed capacitance values is shown in Figure 5.5.

The control signal for data signal generation and sensing is shown in Figure 5.6. In order to get '11', both subbitline 2 and 3 are precharged to $V_{DD}$. Also, the addressed cell and a dummy cell are connected to these two subbitlines to keep the

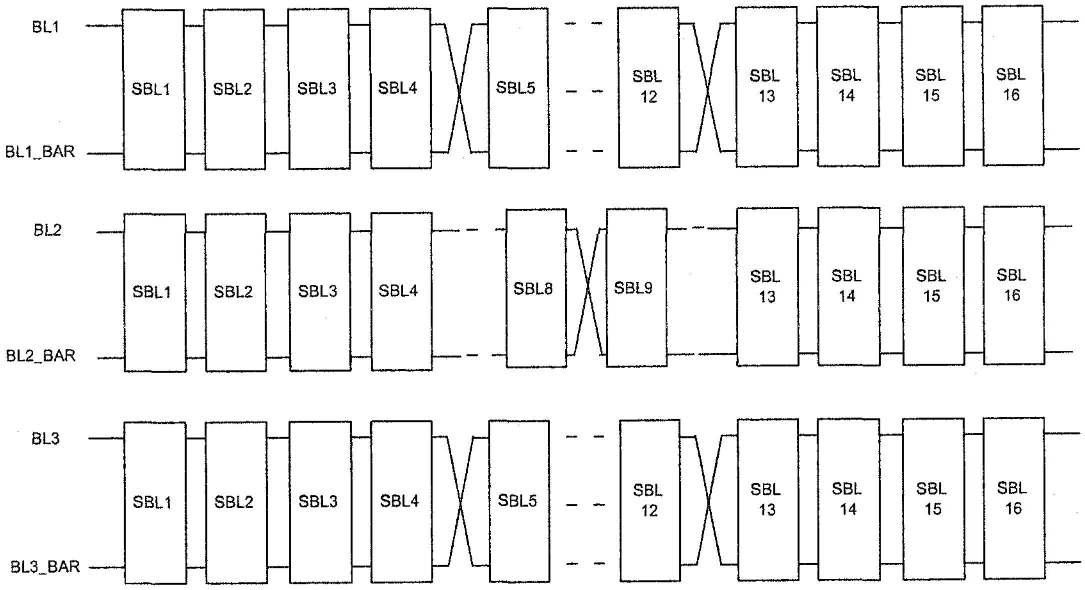


Figure 5.3: Simplified Simulation Schematic

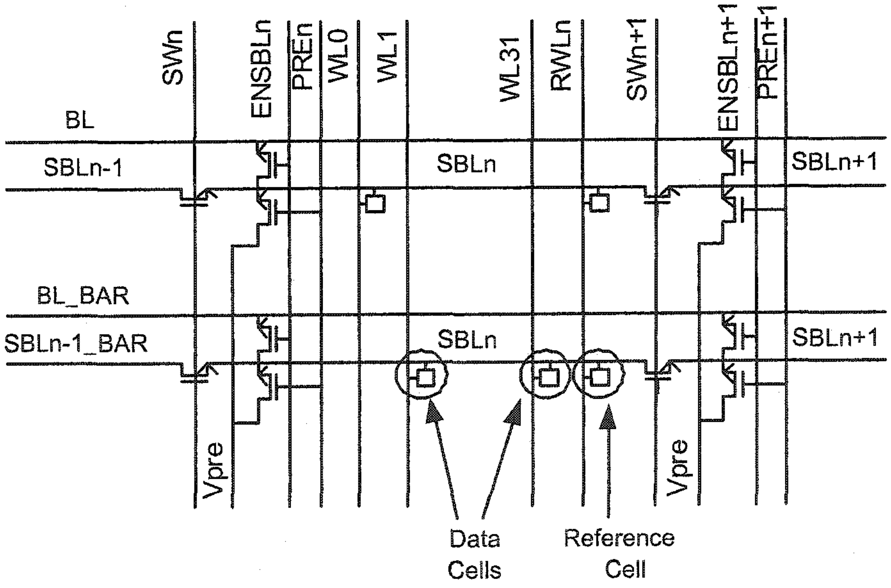


Figure 5.4: Subbitline Schematic

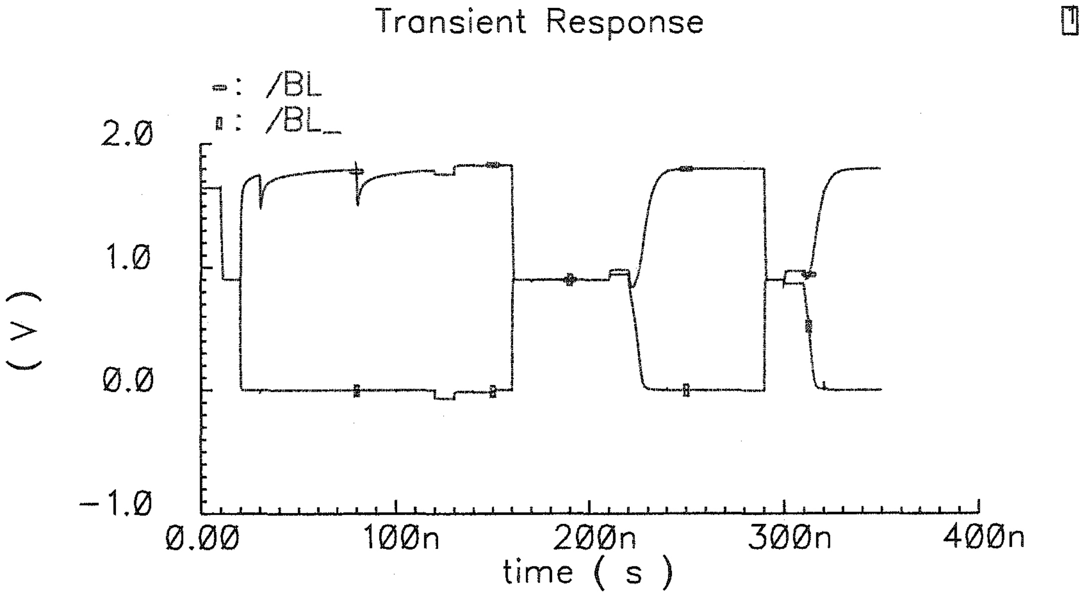


Figure 5.5: Schematic Simulation Waveform for '11'

total capacitance equal. Then the switch between subbitline 2 and 3 is closed and this leads to charge sharing between these two subbitlines. The resulting voltage $V_{DD}$ is then trapped into the 65th cell by de-asserting its wordline. After a period of storage and immediately prior to sensing, the true bitline and subbitline 2 and 3 are all precharged to $\frac{V_{DD}}{2}$, then the trapped data is released onto the two subbitlines to get two copies of the attenuated data signals. Subbitline 3, which carries the first copy of the data signal, is connected with the overlapping bitline first to sense the first bit.

The control signals for reference signal generation are shown in Figure 5.7. The subbitlines connected to the complementary bitline are precharged variously to $V_{DD}$ or $V_{SS}$, as we described in Chapter 3. After charge sharing, two resulting reference voltages, $\frac{V_{DD}}{4}$ and $\frac{3V_{DD}}{4}$, are trapped into the reference cells connected to subbitlines_bar 4 and 13. Then the complementary bitline and subbitline_bar 4 are precharged to $\frac{V_{DD}}{2}$. The first reference voltage $\frac{V_{DD}}{4}$ is released onto the subbitline_bar 4 and then the complementary bitline to produce the attenuated reference voltage.

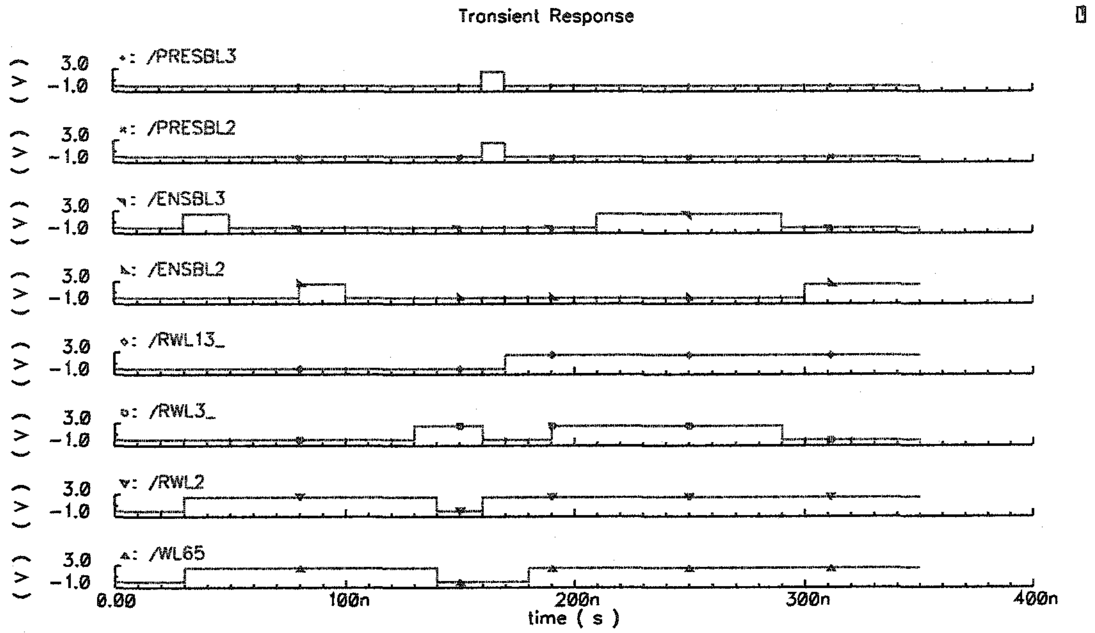


Figure 5.6: Control Signal for Data Signal Generation and Sensing

At the same time, the attenuated data voltage is produced by connecting the data cell holding the data signal with the true subbitline and bitline. After that, sensing begins with the data and first reference voltages. The resulting first bit is then latched by the SA in the buddy array. Then the sensing of the second bit continues using the second copy of the data cell signal and one of the second reference signal.

The superimposed simulation waveforms for '01' and '00' with the assumed capacitances are shown in Figure 5.8.

The functionality of the proposed MLDRAM was validated using the schematic simulation results by inspecting the appropriate events in the waveforms. Next, the schematic was implemented in layout and the more accurate simulations using the extracted simulation models were carried out.

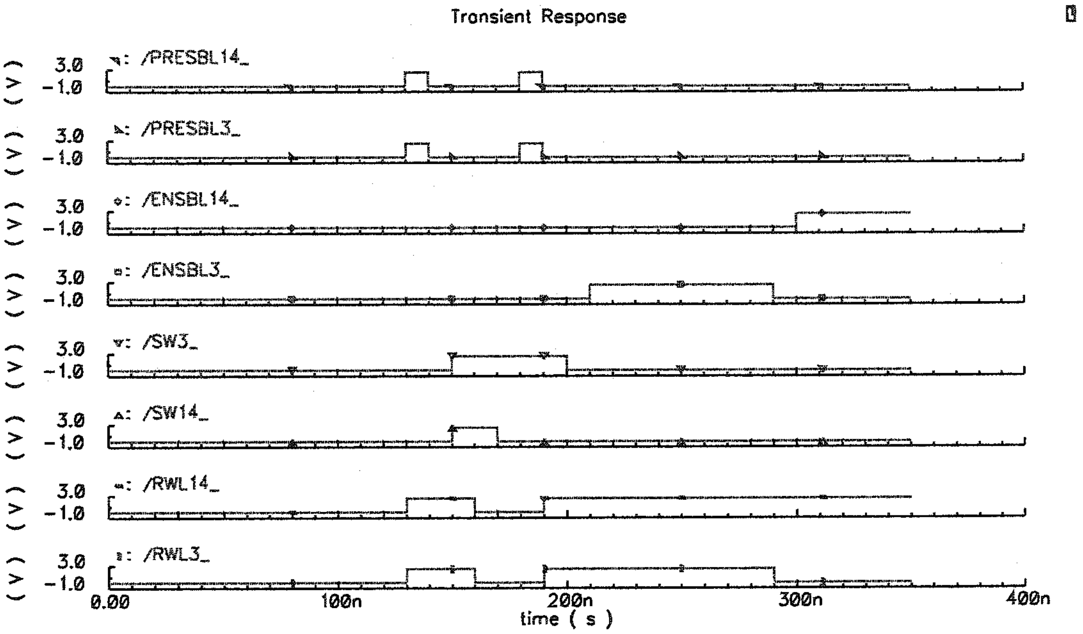


Figure 5.7: Control Signals for Reference Signal Generation

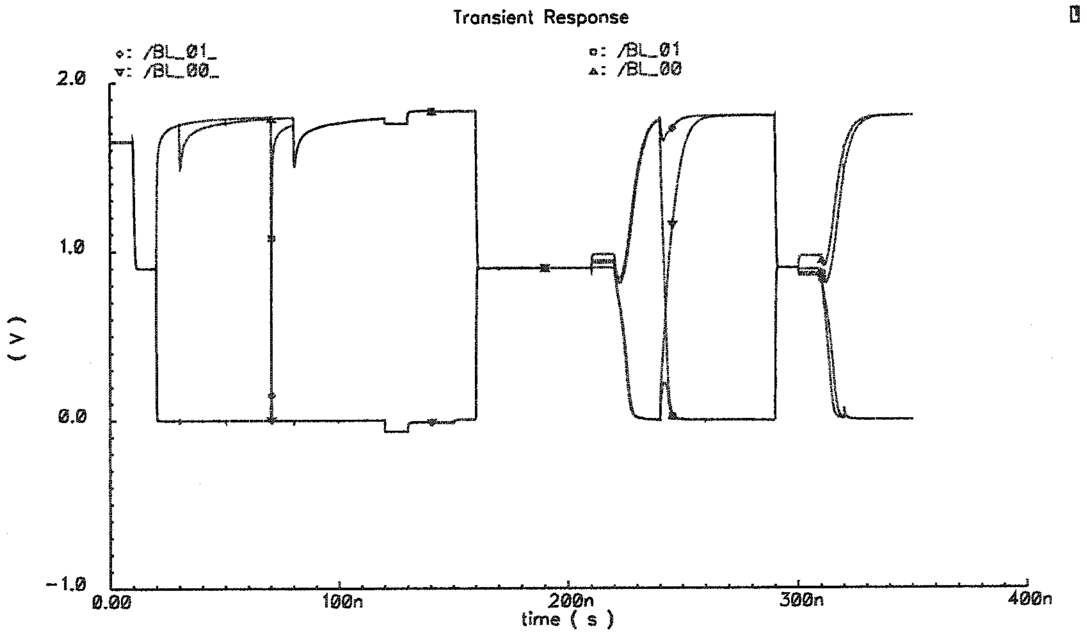


Figure 5.8: Schematic Simulation Waveforms for ‘00’ and ’01’

### 5.3.2 Simulation With Extracted Parasitic Capacitance

In order to run more accurate simulations of the proposed MLDRAM, we needed to design the layout according to the circuit schematic and then extract the layout with the parasitic capacitances. In Cadence, we can enable the software switch 'parasitic_caps' in the layout extractor and then run the extraction with parasitic capacitances.

The same input stimulus file that was used in the previous simulations could be used again. Three data values, '00', '01' and '11', were written into the memory cells, and then read out to verify the functionality of the extracted layout with parasitic capacitances. The superimposed simulation waveforms with the parasitic capacitance are shown in Figure 5.9. All three data values were written and sensed correctly.

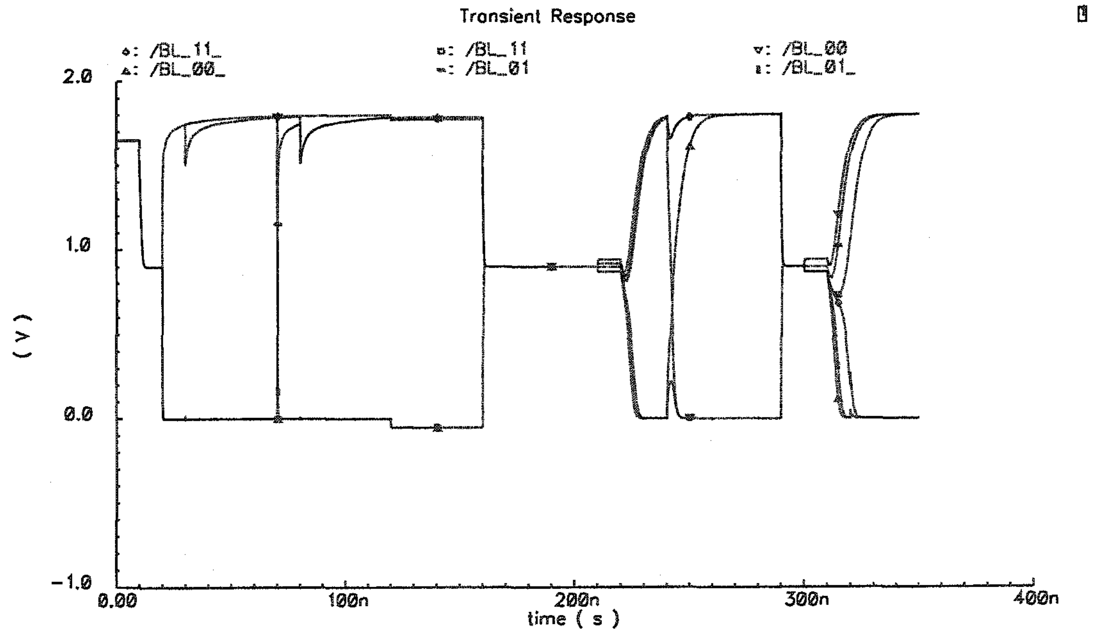


Figure 5.9: Superimposed Simulation Waveforms for '00', '01' and '11'

The resulting differential signals for sensing were slightly smaller that in the previous schematic simulation, which suggests that the manually predicted values of the bitline and subbitline capacitances were smaller than the probably more realistic extracted values.

## 5.4 Capacitance Model of the New Design

The ratio of the total effective bitline capacitance to the cell capacitance is very important to be able to reliably read the data stored in the memory cell despite the signal attenuation. In order to read a memory cell in the new MLDRAM, the cell is first connected to the subbitline, and then afterwards is connected to the bitline. The above mentioned capacitance ratio needs to include all the external circuits that are connected to the cell, and thus must include the subbitline, bitline and the interconnection transistors. If we denote the cell capacitance by $C_{cell}$ and the external capacitance by $C_{ext}$, a new symbol ρ can be defined for the capacitance ratio as follows.

$$\rho = \frac{C_{ext}}{C_{cell}} = \frac{\sum(external\ capacitances\ connected\ to\ cell)}{C_{cell}}$$

Data sensing inevitably involves charge sharing, which is a charge redistribution between the memory cell capacitance and the connected external capacitance. If we denote the original voltage of the cell before accessing by $V_{data}$ and the precharge voltage of the external capacitor by $V_{pre}$, then after charge sharing, the voltage of the external capacitor will have a same voltage $V'_{data}$ as the voltage of the memory cell. By conservation of charge before and after charge sharing we have

$$V'_{data} * (C_{cell} + C_{ext}) = V_{data} * C_{cell} + V_{pre} * C_{ext}$$

Therefore we have

$$V'_{data} = \frac{V_{data} * C_{cell} + V_{pre} * C_{ext}}{C_{cell} + C_{ext}}$$

The reference voltage is generated in a similar way, except that the reference voltage is stored initially in a reference cell. Recall that the reference cell has the same capacitance as the data cell. We can thus assume the same cell capacitance, external capacitance and precharge voltage. The original reference cell voltage is $V_{ref}$. After charge sharing, the voltage of the external circuits have a voltage $V_{ref}^{'}$ as follows:

$$V_{ref}^{'} * (C_{cell} + C_{ext}) = V_{ref} * C_{cell} + V_{pre} * C_{ext}$$

$$V_{ref}^{'} = \frac{V_{ref} * C_{cell} + V_{pre} * C_{ext}}{C_{cell} + C_{ext}}$$

The difference between $V_{data}^{'}$ and $V_{ref}^{'}$ is $V_{diff}$ as follows:

$$V_{diff} = V_{data}^{'} - V_{ref}^{'} = \frac{V_{data} * C_{cell} + V_{pre} * C_{ext}}{C_{cell} + C_{ext}} - \frac{V_{ref} * C_{cell} + V_{pre} * C_{ext}}{C_{cell} + C_{ext}}$$

$$= \frac{(V_{data} - V_{ref}) * C_{cell}}{C_{cell} + C_{ext}} = \frac{V_{data} - V_{ref}}{1 + \frac{C_{ext}}{C_{cell}}} = \frac{V_{data} - V_{ref}}{1 + \rho}$$

The noise margin $(V_{cell} - V_{ref})$ is fixed for a given MLDRAM. As we mentioned in Chapter 2, the noise margin will be reduced when the number of voltage levels is increased in the MLDRAM. If the number of voltage level is N, then the noise margins of a N-level MLDRAM are only $\frac{1}{N-1}$ of those in the conventional DRAM. The final voltage difference $V_{diff}$ is inversely proportional to the ratio $\rho$ of the capacitance of the external circuits to the capacitance of the memory cell.

In order to sense the data voltage reliably, we need the differential bitline signal to be big enough after the attenuation caused by charge sharing, which means we need to place an upper bound to the ratio $\rho$. In a real DRAM chip, it is difficult to make a memory cell with a large capacitance in the very small $8F^2$ area. DRAM manufacturers have done this using trench or stacked capacitors with high dielectric (eg. silicon nitride, ONO and possibly tantalum pentoxide in the future). We can not in the immediate future rely on big increases in the capacitance, so the only choice is to try to keep the external capacitance as small as possible, while keeping the cell capacitance as big as we can, assuming available cells.

### 5.4.1 Analysis of the Various Components of the External Capacitance

In the proposed hierarchical bitline model, the addressed memory cell is first connected to the short subbitline and then connected to the bitline through the subbitline enable transistor. The charge sharing thus occurs in two steps. The first step is between the memory cell and the subbitline, and the second step is between the bitline and the connected subbitline and memory cell. Assume that the voltage of the memory cell is $V_{DD}$ prior to being accessed, and the voltage of the subbitline and bitline are both precharged to $\frac{V_{DD}}{2}$. For serial sensing, we need two copies of the data after the access of the addressed memory cell (to compare with the two different references), and so a reference cell on the adjacent subbitline is required, which is also precharged to $\frac{V_{DD}}{2}$.

Assume that the cell capacitance is 35 fF and that all the cell access and subbitline switch transistors have the same gate dimension $W/L = 0.42\,\mu m/0.18\,\mu m$. With the capacitance parameters given by TSMC [22], the capacitance components of the transistor can be calculated as follows.

(1) Gate capacitance: $C_g = 0.65$ fF

(2) Overlap capacitance: $C_{gdo} = C_{gso}$=0.154 fF

(3) Junction capacitance for the interconnection transistor:

$C_{ji} = C_{jbi} + C_{jswi} = 0.447$ fF

(4) Junction capacitance for the cell access transistor pair:

$C_{ja} = C_{jba} + C_{jswa} = 0.509$ fF

The interconnection and cell access transistors have different structures and capacitances. As mentioned in Chapter 1, two adjacent cell access transistors share the same drain terminal. This can be illustrated in Figure 5.10.

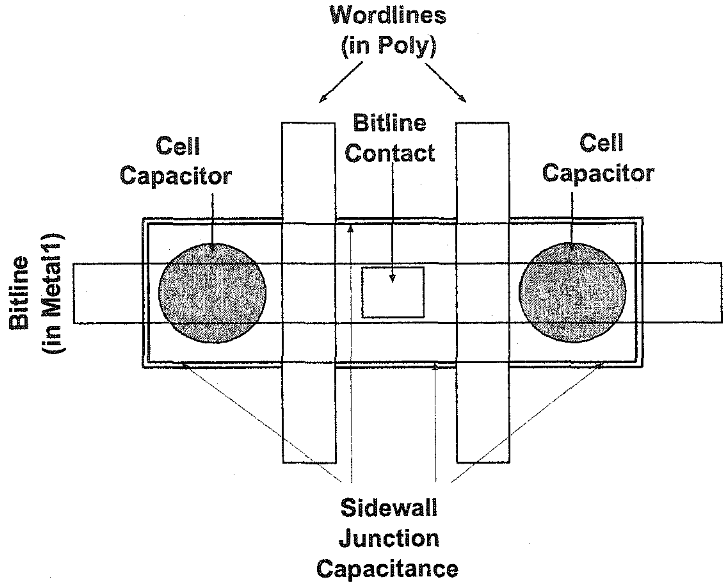


Figure 5.10: Junction Capacitance in the Shared Drain Transistor Pair

In the following parasitic capacitance analysis, we take the above 16-subbitline MLDRAM layout model as an example. It has 16 subbitlines with each bitline, and 16 data cells with each subbitline for a total of 256 cells per bitline. The netlist file of the extracted layout for a complementary bitline is shown as an example in Appendix B. We will measure the parasitic capacitance in this model, then figure out the average capacitance for one unit length (i.e. one micron). Other models for different configuration can then be constructed and analyzed.

#### 5.4.1.1 Components of the External Parasitic Capacitance

1. The parasitic capacitance $C_{BL2\_BAR}$ for the complementary bitline BL2_BAR is 231.78 *f*F.

(1) The capacitance between BL2_BAR and other bitlines and the complementary bitlines is 187.4 fF, which is shown in Figure 5.11. The capacitance C1 between BL2_BAR and BL2 is 65.4 fF; the capacitance C2 between BL2_BAR and BL1_BAR is 21.8 fF; the capacitance C3 between BL2_BAR and BL1 is 38.8 fF; the capacitance C4 between BL2_BAR and BL3 is 39.6 fF; the capacitance C5 be-

tween BL2_BAR and BL3_BAR is 21.8 fF. The length of the bitline is 704.17$\mu$m, so the average capacitance per micron is 0.266 $f$F.

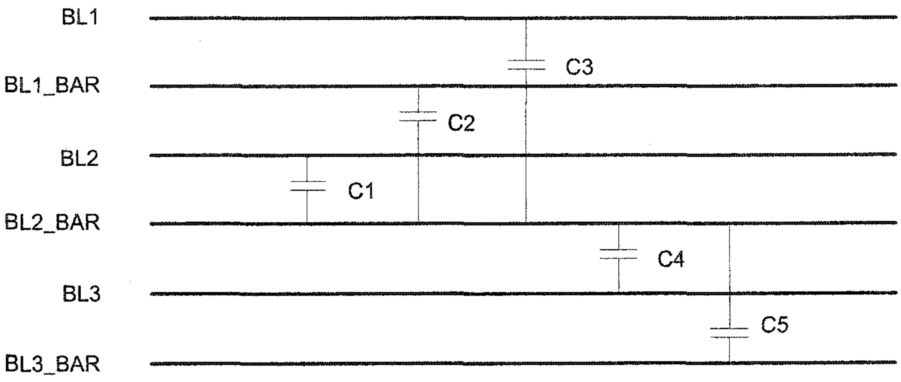


Figure 5.11: Parasitic Capacitance between the Bitlines

(2) The capacitance between BL2_BAR and each of the 16 subbitlines below BL2_BAR is 13.312 fF. The length of the subbitline is 35.92 $\mu$m, so the average capacitance per micron is 0.023 fF. Each subbitline has nine cell pairs connected with it (eight data cell pairs and one reference and dummy cell pair), so the average length per cell pair is 3.99 $\mu$m. The length of the interconnection block is 10.20 $\mu$m.
(3) The capacitance between the complementary bitline BL2_BAR and the metal2 bypass below BL2_BAR is 6.128 fF. This bypass is used to make a compact layout of the interconnection block, which is explained in Chapter 4.
(4) The capacitance between BL2_BAR and metal1 ground GND is 6.13 fF.
(5) The capacitance between BL2_BAR and the 16 metal2 wordlines is 18.81 fF. The poly wordlines are strapped with wires in metal2. Since we have 16 subbitline blocks and each subbitline block (two subbitlines) has 32 wordlines, we have 576 wordlines in total. So the average capacitance for each wordline going in the bitline direction is 0.0356 fF.

2. The parasitic capacitance for one subbitline is 11.482 fF. This value contains the following components:
The capacitance between the subbitline and the bitline is 1.389 fF. The capacitance

between two adjacent subbitlines is 8.745 fF. The capacitance between the subbitline and the conductor that carries ground GND is 1.143 fF. The capacitance between the subbitline and the $V_{PRE}$ conductor is 0.205 fF because the $V_{PRE}$ poly conductor is wider than the other control signal bars in the interconnection block (Recall the layout constraints for $V_{PRE}$ in the assumed process). The length of the subbitline is 35.92 $\mu$m, so the average capacitance of the subbitline per micron is 0.320 fF.

#### 5.4.1.2 Components due to Transistor Capacitance

Except for the parasitic capacitance to nearby interconnect conductors, the bitline and subbitline capacitances are also increased by junction and gate capacitances, which are contributed by the transistors connected to the bitline (or subbitline). Each transistor, which has the drain node connected with the bitline (or subbitline), adds its drain junction capacitance to the whole capacitance of the bitline and subbitline. For the capacitor in the interconnection block, this junction capacitance is 0.447 fF, while the junction capacitance in the memory block is 0.509 fF because two adjacent memory cells share one drain node at the bitline contact to save area. The drain junction capacitance is added to the whole capacitance whenever the transistor is turned on.

The gate capacitance is different, and it works only when the transistor is turned on. The channel is connected with the source and drain nodes and the gate capacitance exists between the channel and the gate. The gate capacitance is 0.65 fF for our 0.42/0.18 $\mu$m transistor.

### 5.4.2 Capacitance Model for the MLDRAM

A Matlab program (shown in Appendix C) was developed to calculate the total capacitance of the bitline. Under the assumption that each bitline has 16 subbitlines, and each subbitline has 16 data cells, 1 reference cell and 1 dummy cell, the total capacitance of the bitline is 243.83 fF, while the total capacitance of each subbitline is 21.938 fF. This leads to a differential signal of 24.743 mV after charge sharing.

Now assume that each bitline has 8 subbitlines, and each subbitline has 32 data cells, 1 reference cell and 1 dummy cell. The total capacitance of the bitline is 204.12 fF while the total capacitance of the subbitline is 38.498 fF. This leads the differential signal to be 28.684 mV.

Now consider a third configuration where each bitline is connected to 32 subbitlines, and each subbitline has 8 data cells, 1 reference cell and 1 dummy cell. The total capacitance with such a bitline is 323.26 fF, while the total capacitance of the subbitline is 13.658 fF. This leads the differential signal to be 20.223 mV.

The above simulation result are compared in Table 5.1.

| | SBL Capacitance (fF) | BL Capacitance (fF) | Resulting Signal Difference (mV) |
|---|---|---|---|
| 16 SBLs / BL and 16 Data Cells / SBL | 243.83 | 21.938 | 24.743 |
| 8 SBLs / BL and 32 Data Cells / SBL | 204.12 | 38.498 | 28.684 |
| 32 SBLs / BL and 8 Data Cells / SBL | 323.26 | 13.658 | 20.223 |

Table 5.1: Simulation Results for Different Models

The simulation results for ten different cell capacitances are shown below in Figure 5.12. Note that the number of data cells per bitline column is 256 for all scenarios in the figure. As expected, the signal voltage increases with the cell capacitance.

The similar simulation has also be carried on for 128-cell-per-bitline structure.

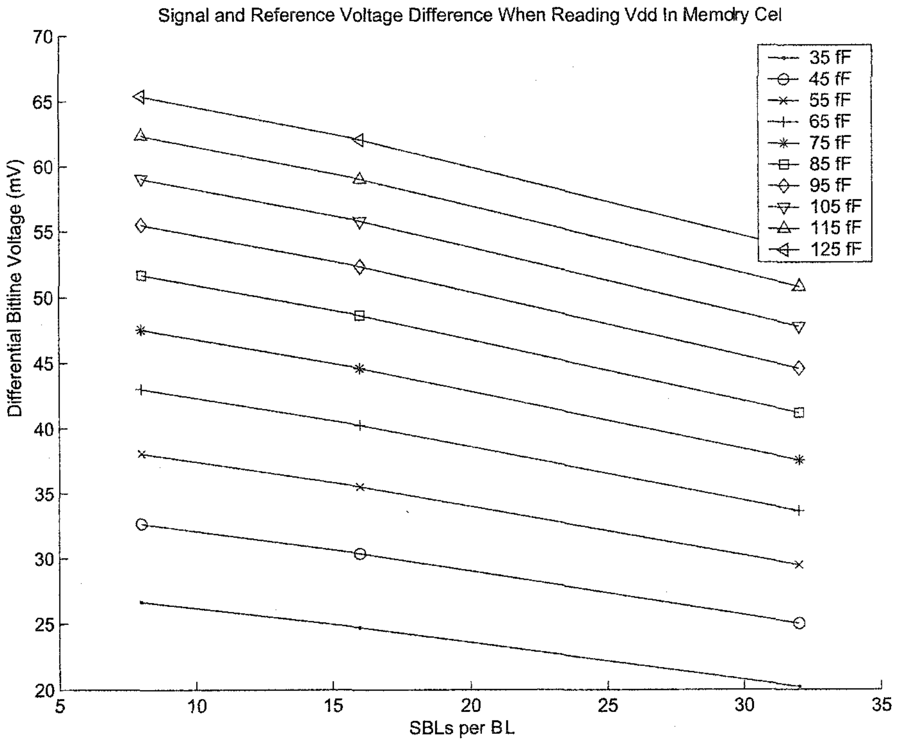


Figure 5.12: Differential Bitline Signal for Various Configurations (256-cell-per-bitline)

The results are plotted in the following Figure 5.13. The resulting differential signals are much bigger (around 15 mV) than those of the 256-cell-per-bitline structure because of the shorter bitlines.

New dielectrics with higher K value have been explored to increase the capacitance of the cell storage capacitor within limited area. For example, if $Ta_2O_5$ dielectric is used, the cell capacitance will be roughly 300 fF. Another simulation using a larger range of $C_{cell}$ was carried out and the results are plotted in the following Figure 5.14.

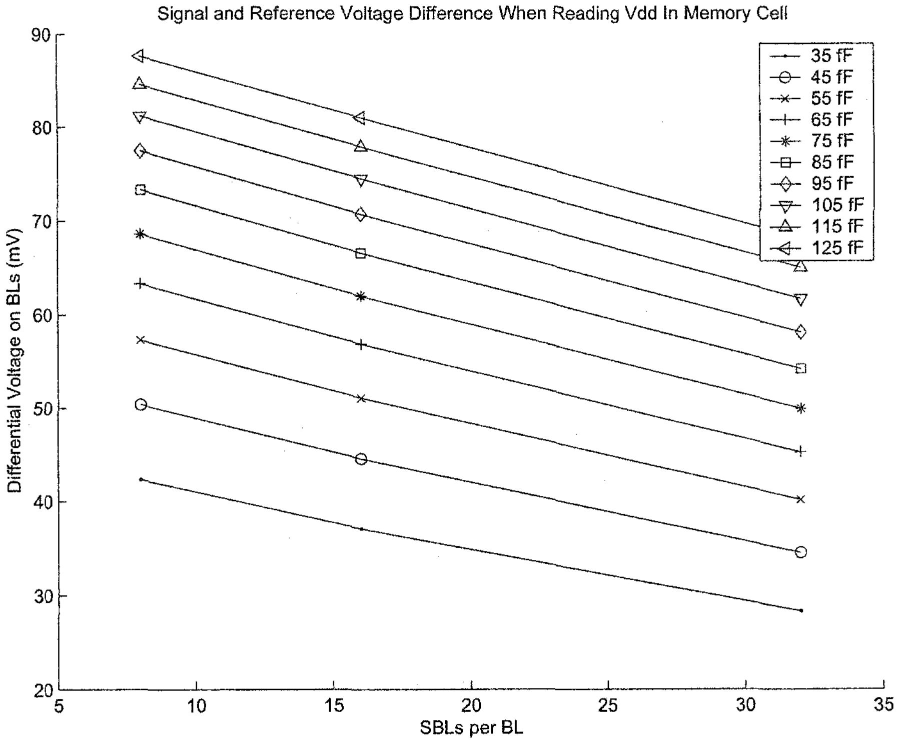


Figure 5.13: Differential Bitline Signal for Various Configurations (128-cell-per-bitline)

## 5.5 Simulation With an Optimized Configuration

To confirm the predictions of the proposed model, a new layout was made, which only has 8 subbitlines per bitline, and each subbitline has 32 data cells, 1 reference cell and 1 dummy cell. The simulation waveform of '11' with the parasitic capacitance is shown in Figure 5.15.

The new control signals are slightly different from those used for the 16-subbitline MLDRAM. Since we only have eight subbitlines for each bitline, the subbitline pairs used previously for data and reference voltage generation will not be used here, only one subbitline is used instead because at least 8 subitlines are required to generate the two different reference voltage by charge sharing. The control signals for data signal generation and sensing is shown in Figure 5.16. The control signals

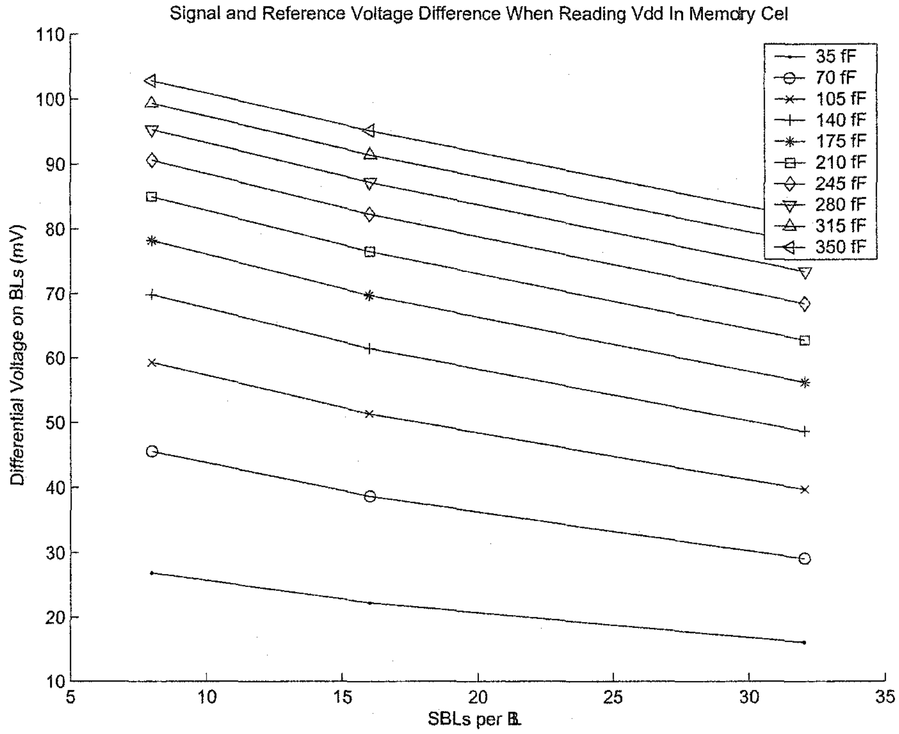


Figure 5.14: Differential Bitline Signals for a Large Range of Cell Sizes

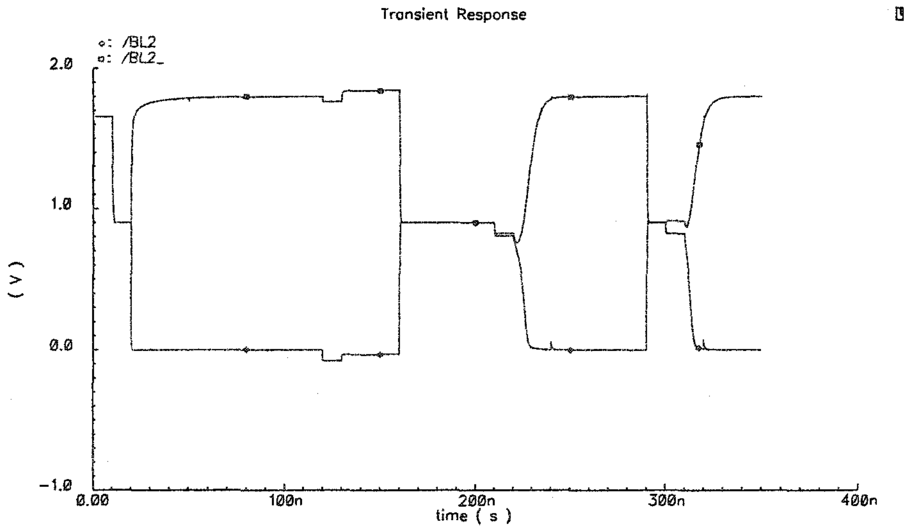


Figure 5.15: Schematic Simulation Waveform for '11'

for reference signal generation is shown in Figure 5.17.

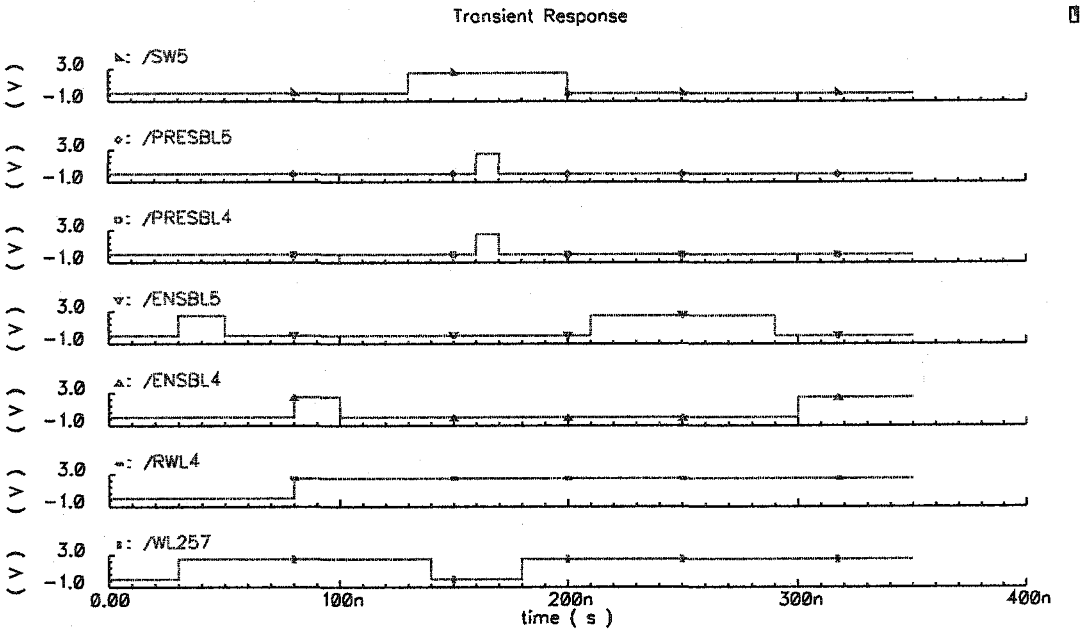


Figure 5.16: Control Signal for Data Signal Generation and Sensing

Compared with the 16-subbitline-per-bitline structure, the differential bitline signal in the new scheme is increased by 5.122 mV in the new structure, which makes the new design slightly more robust against the noise and cell charge leakage. This result is slightly different (1.181 mV higher) from the prediction of the Matlab program because of the effects of other noise sources, such as the noise injection from the wordline to the bitline.

## 5.6 Discussion

The proposed serial MLDRAM has several advantages compared with Birk's MLDRAM with parallel sensing. First, the serial sensing reduces the area overhead of SAs for parallel sensing because only one SA is used for each bitline pair while two SAs are required for parallel sensing.

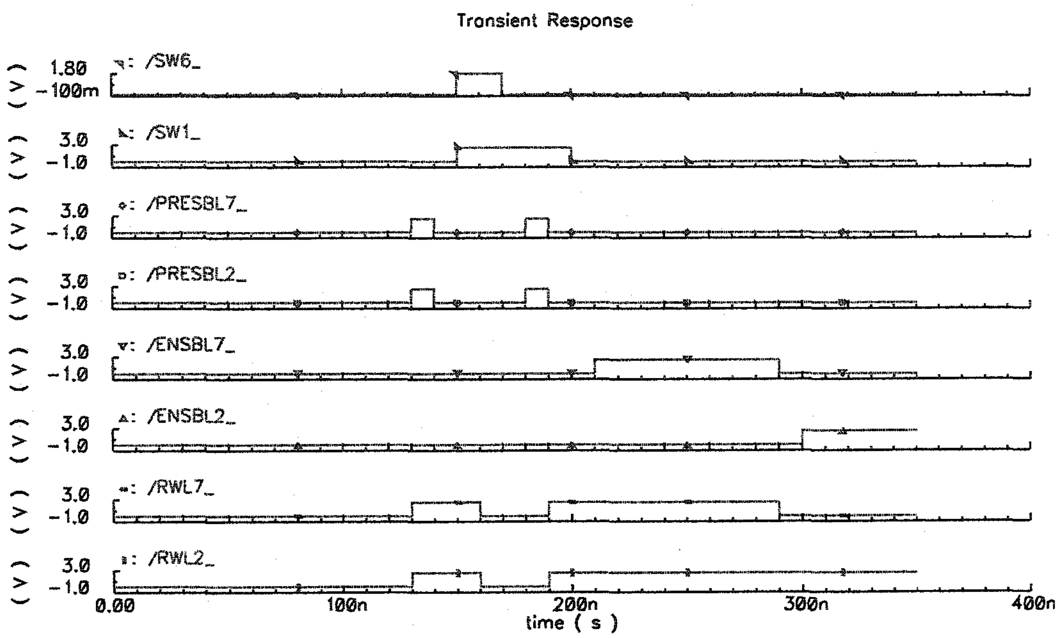


Figure 5.17: Control Signal for Reference Signal Generation

Second, use of the hierarchical, multi-divided bitlines can reduce the capacitance of the external circuits during sensing and minimize cell signal attenuation.

Third, only one SA is used for the bitline pair, so the SAs can be placed at both sides of the memory array and staggered to fit into two bitline pitches, which is commonly used in the conventional DRAM. Both Gillingham's and Birk's ML-DRAMs required the SA to fit within only one bitline pitch.

The storage density is increased in the proposed MLDRAM. Since one memory cell can store one and a half bits, the storage density of MLDRAM can be increased by 50 percent at maximum (for 3-level operation mode). However, some area overhead is added because of the extra circuits for various data and reference signal generations, such as the transistors used in the interconnection block. Thus the final storage density gain will be smaller than the gain produced by the multilevel signaling. There is a more detailed discussion of these tradeoff-offs in the next chapter.

The layout of the proposed MLDRAM was prepared according to a working schematic. The bitline and cell were carefully packed with the minimum spacing. The layout was extracted with parasitic capacitances and a post-layout simulation was carried out with the extracted layout. The simulation results predicted the behavior of the new MLDRAM with more precision.

We analyzed the capacitance components in the proposed MLDRAM. A rough estimate of the capacitance was used to predict the properties of various alternative circuit configurations without requiring the actual layouts.

# Chapter 6

# Conclusions

MLDRAM attempts to increase the storage density of DRAM by using more than two data signal levels in the storage cells. However, the noise margins in ML-DRAMS are decreased by inserting more voltage levels in between the two supply voltages, $V_{DD}$ and $V_{SS}$. They are further decreased due to the charge sharing steps in required MLDRAM read operation, as we mentioned in Chapter 5. The reduction is proportional to the ratio of the capacitance of the external circuits (eg. bitline, any connected subbitlines) to the memory cell being read. In this thesis, a new serial MLDRAM scheme was studied with the aim of reducing the capacitance ratio of the hierarchical and multi-divided bitlines, and thus increasing the noise margins within MLDRAM while preserving the density advantages of MLDRAM. The new MLDRAM design was laid out and simulated in TSMC's 0.18-$\mu$m mixed-signal CMOS technology. Capacitance models of the bitline and subbitline were analyzed and the models were used to predict the capacitance and bitline signal strengths of different configurations. An optimized design was implemented as the result of the project.

## 6.1 Design and Simulation

The memory cell used in our design was inspired by MOSAID's embedded DRAM design scaled down to 0.18-$\mu$m logic technology. A real DRAM process was unavailable to us. Similar embedded DRAM cell designs were used in the previous test chips and proved to be reliable. Automatically extracting the cell capacitor parameters using the CAD tools was a challenge in previous MLDRAM projects in the VLSI laboratory. The transistor used as the cell capacitor has no source and drain nodes in the previous layouts, so the layout extractor could not handle this transistor. The resulting simulations could only be carried out at the schematic level with a hand-calculated and inserted cell capacitance.

The above problems were successfully solved by modifying the memory cell layout. The source and drain nodes of the cell capacitor were added to the layout and connected to the substrate. The new layout could then be extracted automatically to get the transistor which was used as the cell capacitor. A further extraction with parasitic capacitance can be used to simulate the operation of the designed circuit with more precision.

Since the planar cell capacitor is too big to fit into the bitline pitch, each cell in a central memory array column was relocated outside a three-column array and connected with the access transistor through a short diffusion wire. Diffusion was necessary because of layout constraints, but the added resistance should be small for the short wire. The bitlines are still implemented with the minimum spacing to simulate the real operational environment in the DRAM array. The bitline capacitance in our simulation should therefore be a reasonable approximation to that in a real MLDRAM chip using a true DRAM process. Certainly the bitline coupling capacitance will be more realistic than those in previous MLDRAMs designed in the VLSI laboratory.

Two configurations of the proposed MLDRAM were implemented and compared in this project under the fixed constraint of 256 cells per bitline (512 cells per column). The 8-subbitlines-per-bitline hierarchical structure has a smaller bitline capacitance compared with the 16-subbitlines-per-bitline structure and produces a stronger signal after the memory cell access, which makes it more robust to noise and cell charge leakage. Based on an analysis of the parasitic capacitance of the hierarchical bitline, a capacitance model was developed and used to predict the relevant capacitances of the proposed MLDRAM.

The capacitance between the long metal bitline must be taken into consideration carefully when we calculate the capacitance of the entire bitline. Since the final signal voltage depends on the capacitance ratio of the connected portions of the hierarchical bitline (i.e. the effective bitline) and the memory cell, and the bitline capacitance accounts for a major part of the whole external capacitance, methods for deceasing the capacitance of the bitline are very important to the future success of MLDRAM design.

## 6.2 Increased Storage Density with the New Serial MLDRAM

In the new serial MLDRAM, the storage density was increased by storing more than one bit in each memory cell. However, extra area needs to be used to implement the control signal generation, multilevel data and reference signal generation, and the interconnection switch transistors. Thus the final storage density gain will be smaller than the gain produced by the multilevel signaling. The overall storage density increase is calculated and listed in Table 6.1 for the proposed MLDRAM for 3-level as well as 4-level and 6-level operation.

| MLDRAM Scheme | Maximum Gain | Interconnection Overhead | Dummy and Reference Cell Overhead | Final Gain |
|---|---|---|---|---|
| 3-level | 50 % | 16.9 % | 5.8 % | 27.3 % |
| 4-level | 100 % | 16.9 % | 5.8 % | 77.3 % |
| 6-level | 150 % | 16.9 % | 5.8 % | 127.3 % |

Table 6.1: Area Comparison with DRAM for Different Signalling Modes

In the above table, we have assumed that each column (one bitline pair) has 512 data cells (8 subitlines per bitline and 32 data cells per subbitline). The wordline pitch F is taken to be 0.910 $\mu$m. The widths and numbers of different cell array components are listed in Table 6.2. Compared with conventional DRAM, the area overhead in the serial MLDRAM includes the interconnection blocks and the dummy and reference cells. The results in Table 6.1 suggest that the density gain for 3-level mode may be too small to be attractive in production. A 27.3 % density gain will likely be considered too small to offset the risks of a new technology. But it might be useful to build and characterize 3-level test chips for research purposes.

| Serial MLDRAM Component | Dummy Cell and Reference Cell | Subbitline | Interconnection Block | SA |
|---|---|---|---|---|
| Width of Components | 0.91 micron (1F) | 60.1 micron (66 F) | 9.9 micron (10.9 F) | 35.1 micron (38.6 F) |
| Number of Components | 32 | 8 | 8 | 1 |

Table 6.2: Widths and Numbers of Different Components in the Serial MLDRAM

We also compare the storage density situation between the proposed 4-level serial MLDRAM and Birk's 4-level MLDRAM (also assuming 512 data cells for each column). Because the test chip of Birk's design did not use a comparable compact cell array layout, we will assume our own bitline and wordline pitches for Birk's design. We also make the rough assumption that Birk's switch matrix has the same area as the interconnection block in the serial MLDRAM. In Birk's MLDRAM, two more SAs are required for the sensing operation (one is required for each of the three subbitlines in a column). The SAs in Birk's MLDRAM must also be implemented in a single bitline pitch, compared with the double bitline pitch staggered SA arrangement in the serial MLDRAM. We assume that the SA can be realized in the narrower pitch but with the length of the SA being doubled thus preserving the same area. Based on the above assumptions, we can estimate the widths and numbers of different components in Birk's MLDRAM in Table 6.3.

| Birk's MLDRAM Component | Dummy Cell and Reference Cell | Subbitline | Switch Matrix | SA |
|---|---|---|---|---|
| Width of Components | 0.91 micron (1F) | 157.1 micron (172.6 F) | 9.9 micron (10.9 F) | 35.1 micron (77.2 F) |
| Number of Components | 12 | 3 | 3 | 3 |

Table 6.3: Widths and Numbers of Different Components in Birk's MLDRAM

The area overhead in Birk's MLDRAM includes the switch matrixes, the dummy and reference cells and the two additional SAs. The area comparison between 4-level implementations of Serial MLDRAM and Birk's MLDRAM is made in Table 6.4. The new serial MLDRAM has a slight density advantage, but the disadvantage of the required multiple sensing steps.

| MLDRAM Scheme | Maximum Gain | Interconnection Overhead | Dummy and Reference Cell Overhead | SA Overhead | Final Gain |
|---|---|---|---|---|---|
| New 4-level MLDRAM | 100 % | 16.9 % | 5.8 % | 0 % | 77.3 % |
| Birk's 4-level MLDRAM | 100 % | 6.4 % | 2.0 % | 30.0 % | 61.6 % |

Table 6.4: Area Comparison between the Serial MLDRAM and Birk's MLDRAM

From the above table, we can see that in the new serial MLDRAM, most of the area overhead was added by the interconnection blocks. In Birk's MLDRAM, the area for the additional SAs is rather big and they would be difficult to implement in the single bitline pitch. Also, Birk's design requires non-powers-of-two numbers of arrays (eg. 3 arrays for 4 level mode), which may be inconvenient. The serial MLDRAM will support column repair more easily since adjacent columns do not share signals, as they must do in Birk's scheme.

## 6.3 Future Work

Based on the work and discussion in this thesis, we have many suggestions for related future work, as described in the following subsections.

### 6.3.1 Use two transistors in the interconnection block

One interconnection block is required by each subbitline to permit connections with the bitline, the adjacent subbitline and the precharge voltage. For each subbitline pair, 6 transistors are used in the interconnection block and they must be fit into the pitch of the subbitline pair. The layout area of this section was compressed to its minimum size in our design. However, the length of the interconnection block is still 9.9 $\mu$m. This is not short compared with the length of the subbitline memory block, 60.1 $\mu$m. The length of the interconnection block thus accounts for a relatively large (16.5%) part of the bitline. Since the capacitance of the bitline is proportional to its length, it is important to reduce the length of the interconnection block to maximize the noise margins during sensing.

Among the three transistors required in the interconnection block by each subbitline, one transistor is used for the connection to the precharge voltage. We could omit this transistor, and precharge the subbitline through the bitline. The disadvantage of this possibility is that we would no longer be able to precharge all of the subbitlines at the same time because of the two different required precharge voltages. This would make operation of the MLDRAM even slower than before since the subbitlines would need to be precharged in at least two steps (eg. to $V_{SS}$ and $V_{DD}$). The extra slowdown may be acceptable, however, if the memory is used in block-access mode, where the block latency is not critical.

### 6.3.2 Place bitlines in more layers

The capacitance between bitlines is inversely proportional to the distance between adjacent bitlines. In previous test chips, the bitlines were all placed in the same layer, so the bitlines must fit in the bitline pitch. With the further scaling of the minimum feature sizes, this bitline pitch is becoming smaller, which makes the capacitance between bitlines even bigger than before. If we could use more layers to implement the bitlines, the effective distance between the adjacent bitlines could be increased and the bitline capacitance could be reduced. The disadvantage is the increased cost with the extra one or more metal layers.

### 6.3.3 Use real DRAM technology

In this thesis, a planar embedded DRAM design was used for the memory cell. The layout dimensions of this cell are much bigger than a production DRAM cell. In order to simulate the real operation of DRAM, it would be far preferable to use a real DRAM technology to simulate the memory array.

In a real DRAM process, fewer metal layers, usually no more than three layers, are used in order to keep the cost low with fewer masks and processing steps. This limitation was respected in our MLDRAM layouts.

In the real DRAM process, high threshold voltage transistors are used as the cell access transistors to minimize cell leakage and maximize the refreshment period. We used logic transistors in our simulations, which will be slightly faster than DRAM transistors.

Multiple poly layers are used for the implementation of stacked cell capacitors in DRAM. With these poly layers and also high K value dielectrics, stacked cell capacitors with high capacitance can be implemented in a very small area. The DRAM poly layers would be required to get small storage cells, but this did not prevent us from using planar cell capacitors for one column only. The two adjacent dummy columns on either side of the central column with cells ensured that the parasitic capacitances would be reasonably realistic.

## 6.4 Some Future Challenges for MLDRAM

### 6.4.1 Noises with the Scaling of Supply Voltages

With the scaling of the technology, the supply voltage will go down at a similar rate to the device dimensions to avoid the dielectric breakdown of the transistor gate insulator [13]. MLDRAM can use the same advances as DRAM and thus scale down as fast as DRAM.

The noise margins in an MLDRAM are always some fixed fraction smaller than those in a conventional DRAM. Although we can use different methods to improve the performance of MLDRAM, the signal voltage is becoming weaker in absolute voltage with further scaling of the supply voltage. Other methods, such as the error control coding, could be used to improve the reliability of MLDRAM [17].

Many (if not most) noise sources will scale down with the supply voltage, such as the bitline capacitive coupling noise and the wordline injection noise to the bitline, etc. They are all proportional with the supply voltage. Thus if the noise problem in MLDRAM is solved in one process generation, the same techniques should continue to work with scaled processes.

Other noise sources, such as the thermal noise and shot noise, will not scale down with the supply voltage. The thermal noise is proportional with the ambient temperature and has no direct relation with the supply voltage. Thermal noise imposes a fundamental limit to voltage scaling, and the limit will be reached sooner for MLDRAMs, than DRAMs.

### 6.4.2 Speed Issue

The new serial MLDRAM is slower than the conventional DRAM because of the multi-step serial sensing operation. In our simulation, the writing operation took 140 ns and the sensing operation took 180 ns. However, no special effort was made to minimize the sensing time.

Although multi-step serial sensing is slower in the new MLDRAM, the sensing result of the entire row of accessed cells will be held by the SAs as a page of bits. The reading of each bit in such an opened page is fast, almost as fast as that in conventional DRAM (the unary-binary conversion logic will introduce a small delay). MLDRAM should still be suitable for file memory applications, which does not require a high random-access speed as long as the average speed for accessing all the bits in a block is sufficiently fast.

### 6.4.3 Silicon Prototype

A test chip in silicon should be built and characterized for the proposed MLDRAM. Various sizes of cell capacitors and lengths of bitlines should be included to explore the limits of the scheme. A SA with input offset cancelation could be implemented in this test chip to verify its robustness to noise. The layout of the new chip design should be made as compact as possible to better evaluate the area overhead of the MLDRAM scheme. Meanwhile, more than three voltage levels can be used to further increase the storage density. As in the previous test chips ML5 and ML6, the silicon prototype should provide several different numbers of signalling levels.

# Bibliography

[1] A. Chan. Design and implementation of a multilevel DRAM. Master's thesis, University of Alberta, 2000.

[2] Dan Clein and Greg Shimokoru. *CMOS IC Layout - Concept, Methodologies, and Tools*. Newnes, 1999.

[3] J.H. Tapia B.F. Cockburn and D.G. Elliott. A multilevel DRAM with hierarchical bitlines and serial sensing. *2003 IEEE International Workshop on Memory Technology, Design and Testing*, pages 14 – 19, July 2003.

[4] G. Birk D. G. Elliott and B. F. Cockburn. A comparative simulation study of four multilevel DRAMs. *1999 IEEE International Workshop on Memory Technology, Design and Testing*, pages 102–109, Aug. 1999.

[5] G. Bronner et al. A fully planarized 0.25-$\mu$m CMOS technology for 256 mbit DRAM and beyond. *VLSI Technology, 1995. Digest of Technical Papers. 1995 Symposium on*, pages 15–16, 6-8 June 1995.

[6] Greg Atwood et al. Intel StrataFlashTM memory technology overview. *Intel Technology Journal Q4-7*, 4, 1997.

[7] J.H. Ahn et al. Micro villus patterning (MVP) technology for 256 Mb DRAM stack cell. *VLSI Technology, 1992. Digest of Technical Papers. 1992 Symposium on*, pages 12–13, 2-4 June 1992.

[8] M. Horiguchi et al. An experimental large-capacity semiconductor file memory using 16-levels/cell storage. *Solid-State Circuits, IEEE Journal of*, 23(1):27 – 33, Feb. 1988.

[9] T. Furuyama et al. An experimental 2-bit/cell storage DRAM for macrocell or memory-on-logic application. *IEEE JSSC*, 24(2):388–393, April 1989.

[10] P. Gillingham. A sense and restore technique for multilevel DRAM. *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, 43(7):483–486, July 1996.

[11] Richard Gordon Gartnet Inc. Final memory market share rankings: Worldwide, 2003 (executive summary), 2004.

[12] K. Itoh. *VLSI Memory Chip Design*. Springer, 2001.

[13] A. Chandrakasan J. M. Rabaey and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2003.

[14] JR J.P. Eckert. A survey of digital computer memory systems. *IEEE Annals of the History of Computing*, 20(4):693–708, Dec 1998.

[15] Brent Keeth and R. Jacob Baker. *DRAM Circuit Design, A Tutorial*. IEEE Press, 2001.

[16] K. S. Kundert. *The Designer's Guide to SPICE and SPECTRE*. Kluwer Academic Publishers, Norwell, Massachusetts, 1995.

[17] Hui-Ling Lou and C. E. Sundberg. Coded modulation to increase storage capacity of multilevel memories. *Global Telecommunications Conference, 1998*, 6:8–12, Nov. 1998.

[18] B. Prince. *Semiconductor Memories: A Handbook of Design Manufacturing & Applications*. John Wiley & Sons, 1996.

[19] Shunichi Suzuki and Masaki Hirata. Threshold difference compensated sense amplifier. *Solid-State Circuits, IEEE Journal of*, SC-14(6):1066 – 1070, Dec. 1979.

[20] S. Tian. Implementation of sense amplifiers with input offset cancellation. Master's thesis, University of Alberta, August 2004.

[21] Shozo Saito Tohru Furuyama and Syuso Fujii. A new sense amplifier technique for VLSI Dynamic RAM's. *IEDM 81*, 1981.

[22] TSMC. Tsmc 0.18-$\mu$m mixed signal 1p6m salicide 1.8v/3.3v spice models, 2000.

[23] Sue Ann Ung. Design and evaluation of a variable-capacity multilevel DRAM test chip. Master's thesis, University of Alberta, Oct. 2003.

[24] Cadence website. http://www.cadence.com.

[25] MOSAID website. http://www.mosaid.com.

[26] Y. Xiang. Design, implementation and testing of a multilevel DRAM with adjustable cell capacity. Master's thesis, University of Alberta, 2002.

[27] B. F. Cockburn Y. Xiang and D. G. Elliott. Design of a multilevel DRAM with adjustable cell capacity. *Cdn. J. Electrical and Computer Eng.*, 26(2):55–59, April 2001.

[28] Nobuo Nakamura Yohji Watanabe and Shigeyoshi Watanabe. Offset compensating bitline sensing scheme for high density DRAM's. *Solid-State Circuits, IEEE Journal of*, 29(1):1066 – 1070, Jan. 1994.

# Appendix A

```
***************************

1.8V MOS DEVICES MODEL

**************************

   model nch bsim3v3

   1: type=n minr=1e-60 lmin=1.2e-06 - dxl
lmax=2.1e-05 wmin=1.01e-05 - dxw wmax=0.000101 tnom=25 version=3.2
tox=toxn toxm=toxn xj=1.6e-07 nch=3.9e+17 lln=-1 lwn=1 wln=1 wwn=1
lint=1e-08 ll=0 lw=0 lwl=0 wint=1e-08 wl=0 ww=0 wwl=0 mobmod=1
binunit=2 xl= - 2e-08 + dxl xw=0 + dxw dwg=0 dwb=0 ldif=9e-08
hdif=hdifn rsh=6.8 rd=0 rs=0 vth0=0.4452004 + dvthn
lvth0=4.083943e-08 wvth0=-1.48489e-07 pvth0=1.993185e-13
k1=0.5099412 lk1=3.668304e-08 wk1=2.766093e-08 pk1=-2.19267e-13
k2=0.01695608 lk2=-1.755245e-08 wk2=-1.034945e-08
pk2=6.480148e-14 k3=0 dvt0=0 dvt1=0 dvt2=0 dvt0w=0 dvt1w=0
dvt2w=0 nlx=0 w0=0 k3b=0 vsat=90659.09 lvsat=-0.006564545
ua=-7.469327e-10 lua=2.025139e-16 wua=1.830494e-15
pua=-6.726249e-21 ub=2.910666e-18 lub=-1.955871e-25
wub=-6.764042e-24 pub=9.411708e-30 uc=1.641302e-10
luc=-2.017381e-17 wuc=-5.974077e-16 puc=5.963958e-22 rdsw=170
```

```
prwb=0 prwg=0 wr=1 u0=0.04590562 lu0=3.578498e-09
wu0=-4.055627e-08 pu0=-2.913943e-15 a0=0.2119483
la0=-5.177149e-08 wa0=1.948019e-06 pa0=-3.003349e-12
keta=-0.005987795 lketa=-2.748571e-08 wketa=1.97602e-07
pketa=-1.586947e-13 a1=0 a2=0.99 ags=0.02 b0=0 b1=0
voff=-0.1498675 lvoff=4.906377e-09 wvoff=1.042303e-07
pvoff=-1.550514e-13 nfactor=1 cit=-0.0001267591
lcit=2.665205e-10 cdsc=0 cdscb=0 cdscd=0 eta0=5e-05 etab=-5e-05
dsub=0 pclm=0.7736364 lpclm=2.625818e-07 pdiblc1=1e-06
pdiblc2=0.0003968181 lpdiblc2=3.019691e-09 pdiblcb=0.01 drout=0
pscbe1=1.736364e+08 lpscbe1=262.5818 pscbe2=1e-06 pvag=0
delta=0.01 alpha1=0.448150714 beta0=11.59263 kt1=-0.2281038
lkt1=1.487402e-08 wkt1=4.09886e-08 pkt1=-1.499301e-13
kt2=-0.02603289 lkt2=8.023716e-10 wkt2=-4.269e-08
pkt2=-8.087905e-15 at=20000 ute=-1.606637 lute=4.379861e-08
wute=2.685053e-07 pute=-4.4149e-13 ua1=1.224e-09
ub1=-1.459973e-18 lub1=5.961568e-25 wub1=3.423235e-24
pub1=-6.00926e-30 uc1=-5.990395e-11 luc1=1.037438e-16
wuc1=7.225532e-16 puc1=-1.045737e-21 kt1l=0 prt=0 cj=cjn
pb=0.6882682 mj=0.3595262 cjsw=cjswn pbsw=0.6882682
mjsw=0.2003879 cjswg=cjswgn pbswg=0.6882682 mjswg=0.43879
cgdo=cgon cgso=cgon cta=0.001040287 ctp=0.000645489
pta=0.001554306 ptp=0.001554306 js=8.38e-06 jsw=1.6e-11 n=1
xti=3 capmod=3 nqsmod=0 xpart=1 cf=0 tlev=1 tlevc=1
alpha0=0 dlc=3e-9 llc=-0.039
```

# Appendix B

***************************

Netlist of Parasitic Capacitances

**************************

$+6259\ \ (VSS!\ \ BL2\_)\ \ capacitorc = 4.8218e-15\ \ m=1$

$+6218\ \ (BL2\_\ \ 861)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6217\ \ (BL2\_\ \ 819)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6216\ \ (BL2\_\ \ 777)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6215\ \ (BL2\_\ \ 735)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6214\ \ (BL2\_\ \ 690)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6213\ \ (BL2\_\ \ 648)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6212\ \ (BL2\_\ \ 606)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6211\ \ (BL2\_\ \ 564)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6210\ \ (BL2\_\ \ 522)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6209\ \ (BL2\_\ \ 480)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6208\ \ (BL2\_\ \ 438)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6207\ \ (BL2\_\ \ 396)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6206\ \ (BL2\_\ \ 315)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6205\ \ (BL2\_\ \ 273)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6204\ \ (BL2\_\ \ 231)\ \ capacitorc = 1.11198e-16\ \ m=1$

$+6131\ \ (BL2\_\ \ 861)\ \ capacitorc = 8.32462e-16\ \ m=1$

$+6130\ \ (BL2\_\ \ 819)\ \ capacitorc = 8.32462e-16\ \ m=1$

$+6129\ (BL2\_\ 777)\ capacitorc = 8.32462e-16\ m = 1$

$+6128\ (BL2\_\ 735)\ capacitorc = 8.32462e-16\ m = 1$

$+6127\ (BL2\_\ 690)\ capacitorc = 8.20495e-16\ m = 1$

$+6126\ (BL2\_\ 648)\ capacitorc = 8.22204e-16\ m = 1$

$+6125\ (BL2\_\ 606)\ capacitorc = 8.22204e-16\ m = 1$

$+6124\ (BL2\_\ 564)\ capacitorc = 8.22204e-16\ m = 1$

$+6123\ (BL2\_\ 522)\ capacitorc = 8.22204e-16\ m = 1$

$+6122\ (BL2\_\ 480)\ capacitorc = 8.22204e-16\ m = 1$

$+6121\ (BL2\_\ 438)\ capacitorc = 8.22204e-16\ m = 1$

$+6120\ (BL2\_\ 396)\ capacitorc = 8.22204e-16\ m = 1$

$+6119\ (BL2\_\ 357)\ capacitorc = 7.74791e-16\ m = 1$

$+6118\ (BL2\_\ 315)\ capacitorc = 8.32462e-16\ m = 1$

$+6117\ (BL2\_\ 273)\ capacitorc = 8.32462e-16\ m = 1$

$+6116\ (BL2\_\ 231)\ capacitorc = 8.32462e-16\ m = 1$

$+5950\ (BL2\_\ 901)\ capacitorc = 1.72966e-16\ m = 1$

$+5948\ (BL2\_\ 858)\ capacitorc = 1.72966e-16\ m = 1$

$+5946\ (BL2\_\ 816)\ capacitorc = 1.72966e-16\ m = 1$

$+5944\ (BL2\_\ 774)\ capacitorc = 1.72966e-16\ m = 1$

$+5942\ (BL2\_\ 732)\ capacitorc = 1.72966e-16\ m = 1$

$+5940\ (BL2\_\ 690)\ capacitorc = 1.58248e-16\ m = 1$

$+5939\ (BL2\_\ 651)\ capacitorc = 1.72966e-16\ m = 1$

$+5938\ (BL2\_\ 648)\ capacitorc = 1.58248e-16\ m = 1$

$+5937\ (BL2\_\ 609)\ capacitorc = 1.72966e-16\ m = 1$

$+5936\ (BL2\_\ 606)\ capacitorc = 1.58248e-16\ m = 1$

$+5935\ (BL2\_\ 567)\ capacitorc = 1.72966e-16\ m = 1$

$+5934\ (BL2\_\ 564)\ capacitorc = 1.58248e-16\ m = 1$

$+5933\ (BL2\_\ 525)\ capacitorc = 1.72966e-16\ m = 1$

$+5932\ (BL2\_\ 522)\ capacitorc = 1.58248e-16\ m = 1$

$+5931\ (BL2\_\ 483)\ capacitorc = 1.72966e-16\ m = 1$

$+5930\ (BL2\_\ 480)\ capacitorc = 1.58248e-16\ m = 1$

$+5929\ (BL2\_\ 441)\ capacitorc = 1.72966e-16\ \ m=1$

$+5928\ (BL2\_\ 438)\ capacitorc = 1.58248e-16\ \ m=1$

$+5927\ (BL2\_\ 399)\ capacitorc = 1.72966e-16\ \ m=1$

$+5926\ (BL2\_\ 396)\ capacitorc = 1.58248e-16\ \ m=1$

$+5925\ (BL2\_\ 357)\ capacitorc = 1.72966e-16\ \ m=1$

$+5924\ (BL2\_\ 354)\ capacitorc = 1.72966e-16\ \ m=1$

$+5922\ (BL2\_\ 312)\ capacitorc = 1.72966e-16\ \ m=1$

$+5920\ (BL2\_\ 270)\ capacitorc = 1.72966e-16\ \ m=1$

$+5918\ (BL2\_\ 225)\ capacitorc = 1.72966e-16\ \ m=1$

$+5916\ (BL2\_\ DWL\_R)\ capacitorc = 5.68906e-16\ \ m=1$

$+5915\ (BL2\_\ DWL\_L)\ capacitorc = 5.68906e-16\ \ m=1$

$+5914\ (BL2\_\ BL2)\ capacitorc = 4.23036e-16\ \ m=1$

$+5913\ (BL2\_\ WL32)\ capacitorc = 5.68906e-16\ \ m=1$

$+5908\ (WL31\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5903\ (WL30\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5898\ (WL29\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5893\ (WL28\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5888\ (WL27\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5883\ (WL26\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5878\ (WL25\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5873\ (WL24\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5868\ (WL9\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5863\ (WL23\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5858\ (WL8\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5853\ (WL22\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5798\ (WL7\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5792\ (WL21\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5786\ (WL6\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5780\ (WL20\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5774\ (WL5\ BL2\_)\ capacitorc = 5.68906e-16\ \ m=1$

$+5768\ (WL4\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5762\ (WL3\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5756\ (WL2\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5750\ (WL19\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5744\ (WL18\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5738\ (WL17\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5732\ (WL16\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5726\ (WL15\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5720\ (WL14\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5714\ (WL13\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5708\ (WL12\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5702\ (WL11\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5696\ (WL10\ BL2\_)\ capacitorc = 5.68906e-16\ m=1$

$+5655\ (VSS!\ BL2\_)\ capacitorc = 4.45721e-16\ m=1$

$+5490\ (BL2\_\ 690)\ capacitorc = 2.12784e-16\ m=1$

$+4715\ (VSS!\ BL2\_)\ capacitorc = 8.53758e-16\ m=1$

$+3243\ (BL2\_\ BL2)\ capacitorc = 6.54357e-14\ m=1$

$+3242\ (BL2\_\ BL1)\ capacitorc = 3.87938e-14\ m=1$

$+3236\ (BL1\_\ BL2\_)\ capacitorc = 2.18206e-14\ m=1$

$+2994\ (BL2\_\ 861)\ capacitorc = 1.26359e-16\ m=1$

$+2993\ (BL2\_\ 819)\ capacitorc = 1.26359e-16\ m=1$

$+2992\ (BL2\_\ 777)\ capacitorc = 1.26359e-16\ m=1$

$+2991\ (BL2\_\ 735)\ capacitorc = 1.26359e-16\ m=1$

$+2990\ (BL2\_\ 693)\ capacitorc = 2.00102e-16\ m=1$

$+2989\ (BL2\_\ 690)\ capacitorc = 1.26359e-16\ m=1$

$+2988\ (BL2\_\ 648)\ capacitorc = 1.26359e-16\ m=1$

$+2987\ (BL2\_\ 606)\ capacitorc = 1.26359e-16\ m=1$

$+2986\ (BL2\_\ 564)\ capacitorc = 1.26359e-16\ m=1$

$+2985\ (BL2\_\ 522)\ capacitorc = 1.26359e-16\ m=1$

$+2984\ (BL2\_\ 480)\ capacitorc = 1.26359e-16\ m=1$

$+2983\ (BL2\_\ 438)\ capacitorc = 1.26359e-16\ m = 1$

$+2982\ (BL2\_\ 396)\ capacitorc = 1.26359e-16\ m = 1$

$+2981\ (BL2\_\ 315)\ capacitorc = 1.26359e-16\ m = 1$

$+2980\ (BL2\_\ 273)\ capacitorc = 1.26359e-16\ m = 1$

$+2979\ (BL2\_\ 231)\ capacitorc = 1.26359e-16\ m = 1$

$+2978\ (BL2\_\ 218)\ capacitorc = 1.26359e-16\ m = 1$

$+2856\ (BL1\_\ BL2\_)\ capacitorc = 8.13471e-16\ m = 1$

$+2250\ (BL1\_\ BL2\_)\ capacitorc = 5.02194e-16\ m = 1$

$+5923\ (BL2\_\ 315)\ capacitorc = 1.46071e-16\ m = 1$

$+5921\ (BL2\_\ 273)\ capacitorc = 1.46071e-16\ m = 1$

$+5919\ (BL2\_\ 231)\ capacitorc = 1.46071e-16\ m = 1$

$+5917\ (BL2\_\ 218)\ capacitorc = 1.46071e-16\ m = 1$

$+3244\ (BL2\_\ BL3)\ capacitorc = 3.96041e-14\ m = 1$

$+3241\ (BL2\_\ BL3\_)\ capacitorc = 2.18206e-14\ m = 1$

$+2977\ (BL2\_\ BL3\_)\ capacitorc = 8.13471e-16\ m = 1$

$+2267\ (BL2\_\ BL3\_)\ capacitorc = 5.02194e-16\ m = 1$

$+5941\ (BL2\_\ 693)\ capacitorc = 1.46071e-16\ m = 1$

$+5491\ (BL2\_\ 693)\ capacitorc = 2.12784e-16\ m = 1$

$+5949\ (BL2\_\ 861)\ capacitorc = 1.46071e-16\ m = 1$

$+5947\ (BL2\_\ 819)\ capacitorc = 1.46071e-16\ m = 1$

$+5945\ (BL2\_\ 777)\ capacitorc = 1.46071e-16\ m = 1$

$+5943\ (BL2\_\ 735)\ capacitorc = 1.46071e-16\ m = 1$

# Appendix C

****************************************

Matlab Program to Calculate Parasitic Capacitance

****************************************

```
clear all;
close all;
Pattern = ['.-';'o-';'x-';'+-';'*-';'s-';'d-';'v-';'x-';'<-'];
% Structure parameter
Cm = [35,45,55,65,75,85,95,105,115,125];
N_SBL_per_BL = 16;
N_data_cell_pair_per_SBL = 8;
N_RD_cell_pair_per_SBL = 1;
N_cell_pair_per_SBL = N_data_cell_pair_per_SBL + N_RD_cell_pair_per_SBL;
% Dimension parameters
W_mem_block = 33.29;
W_interconn = 10.20;
W_BL = N_SBL_per_BL * (W_mem_block + W_interconn);
W_cell_pair = W_mem_block/N_cell_pair_per_SBL;
% Transitor capacitance
% Gate capacitance
Cg = 0.65;
% Gate over drain capacitance
```

```
Cgdo = 0.154;
%Junction capacitance for the interconnection transistor
Cji = 0.447;
%Junction capacitance for the access transistor pair
Cja = 0.509;
% Transistor capacitance per SBL
% Only count the cap from cell access transistor
Cnp_SBL_all = N_cell_pair_per_SBL*Cja+2*N_cell_pair_per_SBL*Cgdo+Cg;
%Add the interconnection transistor'scaps in
Cnp_SBL_all = Cnp_SBL_all + 3*(Cji+Cgdo)+Cg;
%Parasitic capacitance per SBL
Cp_SBL_BL = 1.389;
Cp_SBL_SBL = 8.745;
Cp_SBL_GND = 1.143;
Cp_SBL_VPRE = 0.205;
Cp_SBL_all = Cp_SBL_BL + Cp_SBL_SBL + Cp_SBL_GND + Cp_SBL_VPRE;
Cp_SBL_BL_per_um = Cp_SBL_BL/W_mem_block;
Cp_SBL_SBL_per_um = Cp_SBL_SBL/W_mem_block;
Cp_SBL_GND_per_um = Cp_SBL_GND/W_mem_block;
% Total cap per SBL
C_SBL_all = Cp_SBL_all + Cnp_SBL_all;
% Transistor capacitance perBL
Cnp_BL_all = N_SBL_per_BL*(Cji+Cgdo)+Cg;
%parasitic capacitance per BL
Cp_BL_BL = 187.5;
Cp_BL_per_SBL = 0.832;
Cp_BL_per_interconn = 0.383;
Cp_BL_GND = 6.121;
Cp_BL_BL_per_um = Cp_BL_BL/W_BL;
Cp_BL_SBL_per_um = Cp_BL_per_SBL/W_mem_block;
```

```
Cp_BL_per_WL = 0.0356;
Cp_BL_all = Cp_BL_BL + Cp_BL_per_SBL * N_SBL_per_BL + Cp_BL_per_interconn *
N_SBL_per_BL + Cp_BL_GND + Cp_BL_per_WL * N_SBL_per_BL * N_cell_pair_per_SBL *
2 * 2;
%total cap per BL
C_BL_all = Cp_BL_all + Cnp_BL_all;
%4models with number of SBL as 4/8/16/32
%Different cell capacitance
forj = 1 : 10
%Different SBL per BL 4/8/16/32
fori = 2 : 4
N_SBL_per_BL = 2^(i + 1);
N_data_cell_pair_per_SBL = 256/N_SBL_per_BL/2;
N_RD_cell_pair_per_SBL = 1;
N_cell_pair_per_SBL = N_data_cell_pair_per_SBL + N_RD_cell_pair_per_SBL;
W_mem_block = W_cell_pair * N_cell_pair_per_SBL;
W_BL = N_SBL_per_BL * (W_mem_block + W_interconn);
%Transistor capacitance per SBL
%Only count the cap from cell access transistor
Cnp_SBL_all = N_cell_pair_per_SBL * Cja + 2 * N_cell_pair_per_SBL * Cgdo + Cg;
%Add in the interconnection transistor's caps
Cnp_SBL_all = Cnp_SBL_all + 3 * (Cji + Cgdo) + Cg;
%Parasitic capacitance per SBL
Cp_SBL_BL = Cp_SBL_BL_per_um * W_mem_block;
Cp_SBL_SBL = Cp_SBL_SBL_per_um * W_mem_block;
Cp_SBL_GND = Cp_SBL_GND_per_um * W_mem_block;
Cp_SBL_VPRE = 0.205;
Cp_SBL_all = Cp_SBL_BL + Cp_SBL_SBL + Cp_SBL_GND + Cp_SBL_VPRE;
%Total cap per SBL
C_SBL_all = Cp_SBL_all + Cnp_SBL_all;
```

```
% Transistor capacitance per BL
Cnp_BL_all = N_SBL_per_BL * (Cji + Cgdo) + Cg;
% Parasitic capacitance per BL
Cp_BL_BL = Cp_BL_BL_per_um * W_BL;
Cp_BL_per_SBL = Cp_BL_SBL_per_um * W_mem_block;
Cp_BL_all = Cp_BL_BL + Cp_BL_per_SBL * N_SBL_per_BL
+Cp_BL_per_interconn * N_SBL_per_BL + Cp_BL_GND + Cp_BL_per_WL
*N_SBL_per_BL * N_cell_pair_per_SBL * 2 * 2;
%TotalcapperBL
C_BL_all = Cp_BL_all + Cnp_BL_all;
C_SBL_list(j,i) = C_SBL_all;
C_BL_list(j,i) = C_BL_all;
% Charge sharing
%Assume the stored data is '1', VDD = 1.8V, and the precharge voltage is VDD/2 =
0.9V.
% Reference voltage is3/4 * VDD = 1.35V
V_diff = (1.8 - 3/4 * 1.8)/(1 + (Cm(j) + 2 * C_SBL_all + Cm(j) + 2 * C_BL_all)/Cm(j)) *
1000;
V_diff_list(j,i) = V_diff;
end;
end;
% Plot section figure(1);
hold on;
subplot(1,3,1);
stem([4,8,16,32],C_SBL_list(j,:));
title('Capacitance of the SBL');
xlabel('SBLs per BL');
ylabel('Capacitance (fF)');
subplot(1,3,2);
stem([4,8,16,32],C_BL_list(j,:));
```

```
title('Capacitance of the BL');
xlabel('SBLs per BL');
ylabel('Capacitance (fF)');
subplot(1,3,3);
plot([8,16,32],V_diff_list(1,2:4),pattern(1,:))
title('Signal and Reference Voltage Difference WhenReading Vdd In Memory Cell');
xlabel('SBLs per BL');
ylabel('Voltage (mV)');
legend('35uF','45uF','55uF','65uF','75uF','85uF','95uF','105uF','115uF','125uF');
%End Of Program
```