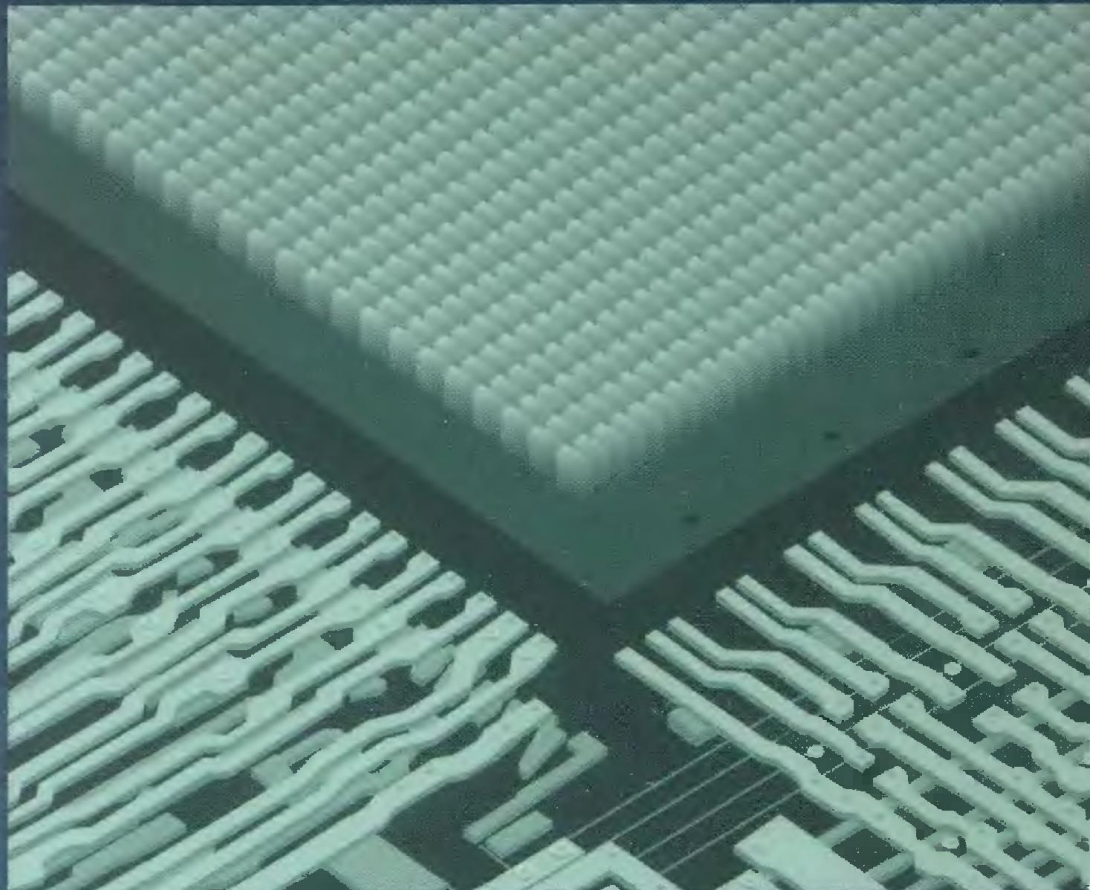


DRAM

Circuit Design

A TUTORIAL

Brent Keeth • R. Jacob Baker



**IEEE Press Series on
Microelectronic Systems**

Stuart K. Tewksbury and Joe E. Brewer,
Series Editors

BESANG EX2035
Micron v. BeSang, IPR2023-00900
Page (cover-1)

DRAM CIRCUIT DESIGN

IEEE Press
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331

IEEE Press Editorial Board
Robert J. Herrick, *Editor in Chief*

M. Akay	M. Eden	M. S. Newman
J. B. Anderson	M. E. El-Hawary	M. Padgett
P. M. Anderson	R. F. Hoyt	W. D. Reeve
J. E. Brewer	S. V. Kartalopoulos	G. Zobrist
	D. Kirk	

Kenneth Moore, *Director of IEEE Press*
Catherine Faduska, *Senior Acquisitions Editor*
John Griffin, *Acquisitions Editor*
Robert H. Bedford, *Assistant Acquisitions Editor*
Surendra Bhimani, *Production Editor*

IEEE Solid-State Circuits Society, *Sponsor*
SSC-S Liaison to IEEE Press, Stuart K. Tewksbury

Cover design: William T. Donnelly, *WT Design*
Technical edits and illustrations: Mary J. Miller, *Micron Technology, Inc.*

Technical Reviewers
Joseph P. Skudlarek, *Cypress Semiconductor Corporation, Beaverton, OR*
Roger Norwood, *Micron Technology, Inc., Richardson, TX*
Elizabeth J. Brauer, *Northern Arizona University, Flagstaff, AZ*

Books of Related Interest from the IEEE Press

Electronic and Photonic Circuits and Devices

Edited by Ronald W. Waynant and John K. Lowell

A Selected Reprint Volume

1999	Softcover	232 pp	IEEE Order No. PP5748	ISBN 0-7803-3496-5
------	-----------	--------	-----------------------	--------------------

EMC and the Printed Circuit Board: Design, Theory, and Layout Made Simple

Mark I. Montrose

1999	Hardcover	344 pp	IEEE Order No. PC5756	ISBN 0-7803-4703-X
------	-----------	--------	-----------------------	--------------------

High-Performance System Design: Circuits and Logic

Edited by Vojin G. Oklobdzija

A Selected Reprint Volume

1999	Hardcover	560 pp	IEEE Order No. PC 5765	ISBN 0-7803-4716-1
------	-----------	--------	------------------------	--------------------

Integrated Circuits for Wireless Communications

Edited by Asad A. Abidi, Paul R. Gray, Robert G. Meyer

A Selected Reprint Volume

1999	Hardcover	688 pp	IEEE Order No. PC5716	ISBN 0-7803-3459-0
------	-----------	--------	-----------------------	--------------------

CMOS: Circuit Design, Layout, and Simulation

R. Jacob Baker, Harry W. Li, David E. Boyce

1998	Hardcover	944 pp	IEEE Order No. PC5689	ISBN 0-7803-3416-7
------	-----------	--------	-----------------------	--------------------

DRAM CIRCUIT DESIGN

A Tutorial

Brent Keeth

*Micron Technology, Inc.
Boise, Idaho*

R. Jacob Baker

*Boise State University
Micron Technology, Inc.
Boise, Idaho*

IEEE Solid-State Circuits Society, *Sponsor*



**IEEE Press Series on
Microelectronic Systems**

Stuart K. Tewksbury and Joe E. Brewer, *Series Editors*



**IEEE
PRESS**

The Institute of Electrical and Electronics Engineers, Inc., New York

This book and other books may be purchased at a discount from the publisher when ordered in bulk quantities. Contact:

IEEE Press Marketing
Attn: Special Sales
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Fax: +1 732 981 9334

For more information about IEEE Press products, visit the
IEEE Online Catalog & Store: <http://www.ieee.org/ieeestore>.

©2001 by the Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, 17th Floor, New York, NY 10016-5997.

All rights reserved. No part of this book may be reproduced in any form, nor may it be stored in a retrieval system or transmitted in any form, without written permission from the publisher.

10 9 8 7 6 5 4 3 2 1

ISBN 0-7803-6014-1
IEEE Order No. PC5863

Library of Congress Cataloging-in-Publication Data

Keeth, Brent, 1960–

DRAM circuit design : a tutorial / Brent Keeth, R. Jacob Baker.

p. cm.

“IEEE Solid-State Circuits Society, sponsor.”

Includes bibliographical references and index.

ISBN 0-7803-6014-1

I. Semiconductor storage devices Design and construction. I. Baker, R. Jacob, 1964–

II. Title

TK7895.M4 K425 2000

621.39'732--dc21

00-059802

Contents

Preface	xi
Acknowledgments	xiii
List of Figures	xv
Chapter 1 An Introduction to DRAM	
1.1 DRAM Types and Operation	1
1.1.1 The 1k DRAM (First Generation)	1
1.1.2 The 4k–64 Meg DRAM (Second Generation)	7
1.1.3 Synchronous DRAMs (Third Generation)	16
1.2 DRAM Basics	22
1.2.1 Access and Sense Operations	26
1.2.2 Write Operation	30
1.2.3 Opening a Row (Summary)	31
1.2.4 Open/Folded DRAM Array Architectures	33
Chapter 2 The DRAM Array	
2.1 The Mbit Cell	35
2.2 The Sense Amp	46
2.2.1 Equilibration and Bias Circuits	46
2.2.2 Isolation Devices	48
2.2.3 Input/Output Transistors	49
2.2.4 Nsense- and Psense-Amplifiers	50
2.2.5 Rate of Activation	52
2.2.6 Configurations	52
2.2.7 Operation	55

2.3	Row Decoder Elements	57
2.3.1	Bootstrap Wordline Driver	58
2.3.2	NOR Driver	60
2.3.3	CMOS Driver	61
2.3.4	Address Decode Trees	62
2.3.5	Static Tree	62
2.3.6	P&E Tree	63
2.3.7	Predecoding	64
2.3.8	Pass Transistor Tree	65
2.4	Discussion	65

Chapter 3 Array Architectures

3.1	Array Architectures	69
3.1.1	Open Digitline Array Architecture	69
3.1.2	Folded Array Architecture	79
3.2	Design Examples: Advanced Bilevel DRAM Architecture	87
3.2.1	Array Architecture Objectives	88
3.2.2	Bilevel Digitline Construction	89
3.2.3	Bilevel Digitline Array Architecture	93
3.2.4	Architectural Comparison	98

Chapter 4 The Peripheral Circuitry

4.1	Column Decoder Elements	105
4.2	Column and Row Redundancy	108
4.2.1	Row Redundancy	110
4.2.2	Column Redundancy	113

Chapter 5 Global Circuitry and Considerations

5.1	Data Path Elements	117
5.1.1	Data Input Buffer	117
5.1.2	Data Write Muxes	121
5.1.3	Write Driver Circuit	122
5.1.4	Data Read Path	124
5.1.5	DC Sense Amplifier (DCSA)	125
5.1.6	Helper Flip-Flop (HFF)	127
5.1.7	Data Read Muxes	129
5.1.8	Output Buffer Circuit	129
5.1.9	Test Modes	131

5.2 Address Path Elements	132
5.2.1 Row Address Path	133
5.2.2 Row Address Buffer	133
5.2.3 CBR Counter	134
5.2.4 Predecode Logic	135
5.2.5 Refresh Rate	135
5.2.6 Array Buffers	137
5.2.7 Phase Drivers	137
5.2.8 Column Address Path	138
5.2.9 Address Transition Detection	139
5.3 Synchronization in DRAMs	142
5.3.1 The Phase Detector	144
5.3.2 The Basic Delay Element	144
5.3.3 Control of the Shift Register	145
5.3.4 Phase Detector Operation	146
5.3.5 Experimental Results	147
5.3.6 Discussion	149
Chapter 6 Voltage Converters	
6.1 Internal Voltage Regulators	155
6.1.1 Voltage Converters	155
6.1.2 Voltage References	156
6.1.3 Bandgap Reference	161
6.1.4 The Power Stage	162
6.2 Pumps and Generators	166
6.2.1 Pumps	167
6.2.2 DVC2 Generator	174
6.3 Discussion	174
Appendix	177
Glossary	189
Index	193
About the Authors	199

An Introduction to DRAM

Dynamic random access memory (DRAM) integrated circuits (ICs) have existed for more than twenty-five years. DRAMs evolved from the earliest 1-kilobit (Kb) generation to the recent 1-gigabit (Gb) generation through advances in both semiconductor process and circuit design technology. Tremendous advances in process technology have dramatically reduced feature size, permitting ever higher levels of integration. These increases in integration have been accompanied by major improvements in component yield to ensure that overall process solutions remain cost-effective and competitive. Technology improvements, however, are not limited to semiconductor processing. Many of the advances in process technology have been accompanied or enabled by advances in circuit design technology. In most cases, advances in one have enabled advances in the other. In this chapter, we introduce some fundamentals of the DRAM IC, assuming that the reader has a basic background in complementary metal-oxide semiconductor (CMOS) circuit design, layout, and simulation [1].

1.1 DRAM TYPES AND OPERATION

To gain insight into how modern DRAM chips are designed, it is useful to look into the evolution of DRAM. In this section, we offer an overview of DRAM types and modes of operation.

1.1.1 The 1k DRAM (First Generation)

We begin our discussion by looking at the 1,024-bit DRAM (1,024 x 1 bit). Functional diagrams and pin connections appear in Figure 1.1 and Figure 1.2, respectively. Note that there are 10 address inputs with pin labels R_1 – R_5 and C_1 – C_5 . Each address input is connected to an on-chip address input buffer. The input buffers that drive the *row* (R) and *column*

(C) decoders in the block diagram have two purposes: to provide a known *input capacitance* (C_{IN}) on the address input pins and to detect the input address signal at a known level so as to reduce timing errors. The level V_{TRIP} , an idealized trip point around which the input buffers slice the input signals, is important due to the finite transition times on the chip inputs (Figure 1.3). Ideally, to avoid distorting the duration of the logic zeros and ones, V_{TRIP} should be positioned at a known level relative to the maximum and minimum input signal amplitudes. In other words, the reference level should change with changes in temperature, process conditions, *input maximum amplitude* (V_{IH}), and *input minimum amplitude* (V_{IL}). Having said this, we note that the input buffers used in first-generation DRAMs were simply inverters.

Continuing our discussion of the block diagram shown in Figure 1.1, we see that five address inputs are connected through a decoder to the 1,024-bit memory array in both the row and column directions. The total number of addresses in each direction, resulting from decoding the 5-bit word, is 32. The single memory array is made up of 1,024 memory elements laid out in a square of 32 rows and 32 columns. Figure 1.4 illustrates the conceptual layout of this memory array. A memory element is located at the intersection of a row and a column.

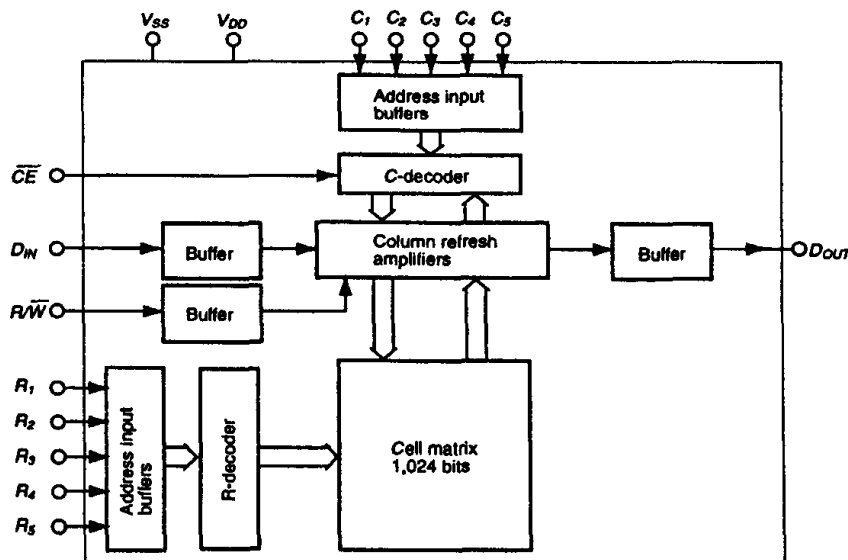


Figure 1.1 1,024-bit DRAM functional diagram.

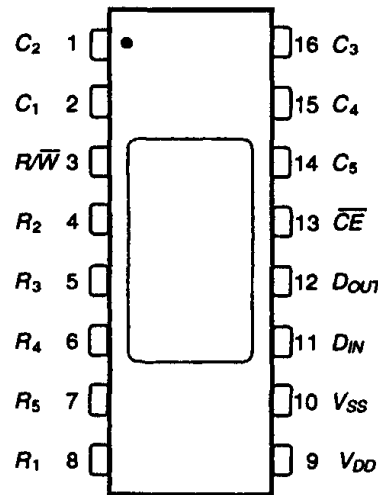


Figure 1.2 1,024-bit DRAM pin connections.

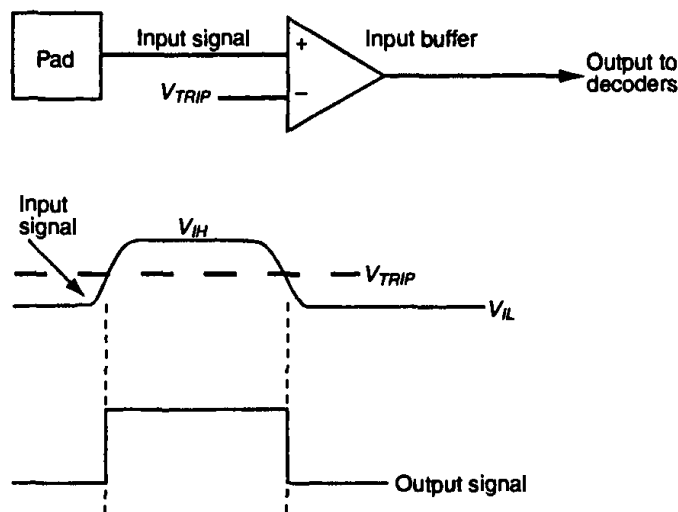


Figure 1.3 Ideal address input buffer.

By applying an address of all zeros to the 10 address input pins, the memory data located at the intersection of row0, RA0, and column 0, CA0, is accessed. (It is either written to or read out, depending on the state of the $R/\overline{W}$ input and assuming that the $\overline{CE}$ pin is LOW so that the chip is enabled.)

It is important to realize that a single bit of memory is accessed by using both a row and a column address. Modern DRAM chips reduce the number of external pins required for the memory address by using the same pins for

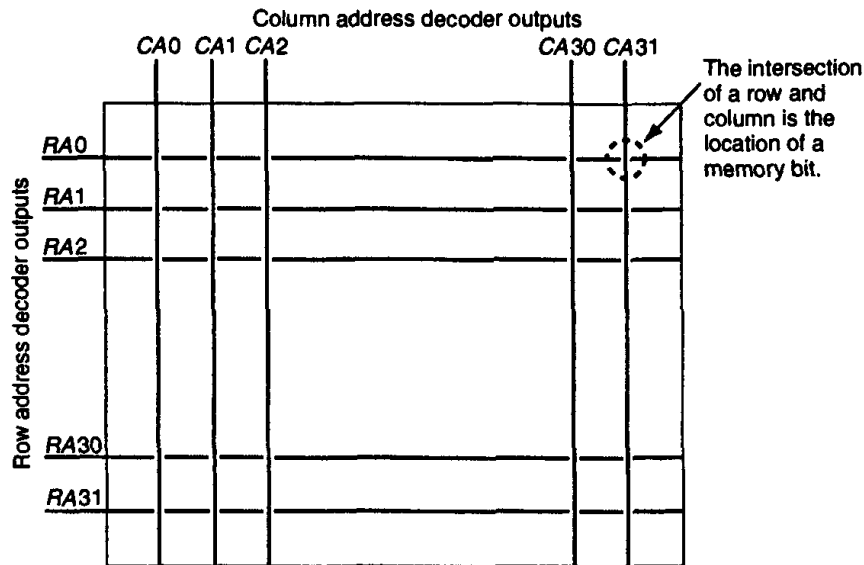


Figure 1.4 Layout of a 1,024-bit memory array.

both the row and column address inputs (*address multiplexing*). A clock signal *row address strobe* ($\overline{RAS}$) strobes in a row address and then, on the same set of address pins, a clock signal *column address strobe* ($\overline{CAS}$) strobes in a column address at a different time.

Also note how a first-generation memory array is organized as a logical square of memory elements. (At this point, we don't know what or how the memory elements are made. We just know that there is a circuit at the intersection of a row and column that stores a single bit of data.) In a modern DRAM chip, many smaller memory arrays are organized to achieve a larger memory size. For example, 1,024 smaller memory arrays, each composed of 256 kbits, may constitute a 256-Meg (256 million bits) DRAM.

1.1.1.1 Reading Data Out of the 1k DRAM. Data can be read out of the DRAM by first putting the chip in the Read mode by pulling the $R/\overline{W}$ pin HIGH and then placing the chip enable pin $\overline{CE}$ in the LOW state. Figure 1.5 illustrates the timing relationships between changes in the address inputs and data appearing on the D_{OUT} pin. Important timing specifications present in this figure are *Read cycle time* (t_{RC}) and *Access time* (t_{AC}). The term t_{RC} specifies how fast the memory can be read. If t_{RC} is 500 ns, then the DRAM can supply 1-bit words at a rate of 2 MHz. The term t_{AC} speci-

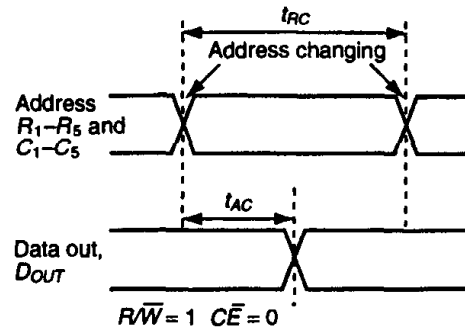


Figure 1.5 1k DRAM Read cycle.

fies the maximum length of time after the input address is changed before the output data (D_{OUT}) is valid.

1.1.1.2 Writing to the 1k DRAM. Writing data to the DRAM is accomplished by bringing the $R/\overline{W}$ input LOW with valid data present on the D_{IN} pin. Figure 1.6 shows the timing diagram for a Write cycle. The term *Write cycle time* (t_{WC}) is related to the maximum frequency at which we can write data into the DRAM. The term *Address to Write delay time* (t_{AW}) specifies the time between the address changing and the $R/\overline{W}$ input going LOW. Finally, *Write pulse width* (t_{WP}) specifies how long the input data must be present before the $R/\overline{W}$ input can go back HIGH in preparation for another Read or Write to the DRAM. When writing to the DRAM, we can think of the $R/\overline{W}$ input as a clock signal.

1.1.1.3 Refreshing the 1k DRAM. The dynamic nature of DRAM requires that the memory be refreshed periodically so as not to lose the contents of the memory cells. Later we will discuss the mechanisms that lead to the dynamic operation of the memory cell. At this point, we discuss how memory Refresh is accomplished for the 1k DRAM.

Refreshing a DRAM is accomplished internally: external data to the DRAM need not be applied. To refresh the DRAM, we periodically access the memory with every possible row address combination. A timing diagram for a Refresh cycle is shown in Figure 1.7. With the $\overline{CE}$ input pulled HIGH, the address is changed, while the $R/\overline{W}$ input is used as a strobe or clock signal. Internally, the data is read out and then written back into the same location at full voltage; thus, logic levels are restored (or refreshed).

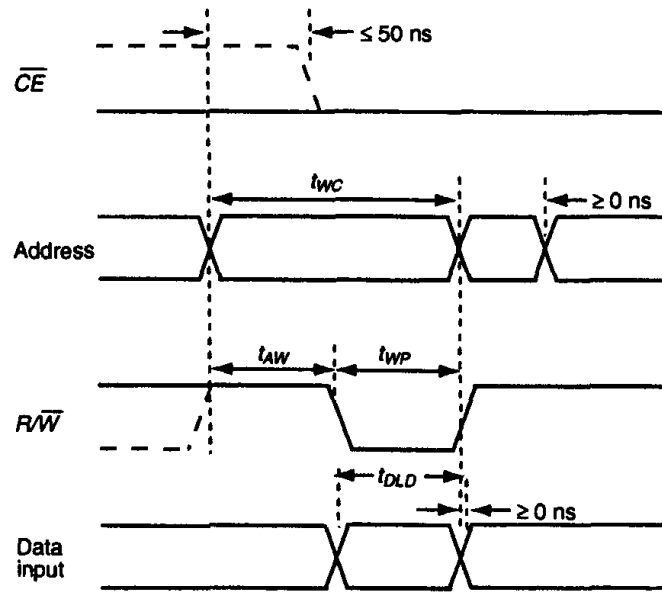


Figure 1.6 1k DRAM Write cycle.

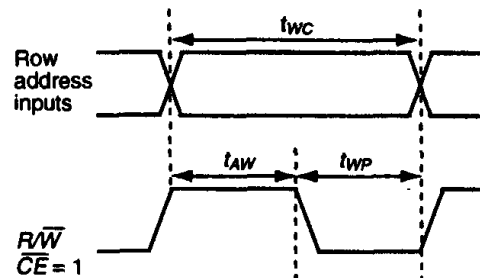


Figure 1.7 1k DRAM Refresh cycle.

1.1.1.4 A Note on the Power Supplies. The voltage levels used in the 1k DRAM are unusual by modern-day standards. In reviewing Figure 1.2, we see that the 1k DRAM chip uses two power supplies: V_{DD} and V_{SS} . To begin, V_{SS} is a greater voltage than V_{DD} : V_{SS} is nominally 5 V, while V_{DD} is -12 V. The value of V_{SS} was set by the need to interface to logic circuits that were implemented using transistor-transistor logic (TTL) logic. The 17-V difference between V_{DD} and V_{SS} was necessary to maintain a large signal-to-noise ratio in the DRAM array. We discuss these topics in greater detail later in the book. The V_{SS} power supply used in modern DRAM designs, at the time of this writing, is generally zero; the V_{DD} is in the neighborhood of 2.5 V.

1.1.1.5 The 3-Transistor DRAM Cell. One of the interesting circuits used in the 1k DRAM (and a few of the 4k and 16k DRAMs) is the 3-transistor DRAM memory cell shown in Figure 1.8. The column- and rowlines shown in the block diagram of Figure 1.1 are split into Write and Read line pairs. When the Write rowline is HIGH, M1 turns ON. At this point, the data present on the Write columnline is passed to the gate of M2, and the information voltage charges or discharges the input capacitance of M2. The next, and final, step in writing to the mbit cell is to turn OFF the Write rowline by driving it LOW. At this point, we should be able to see why the memory is called dynamic. The charge stored on the input capacitance of M2 will leak off over time.

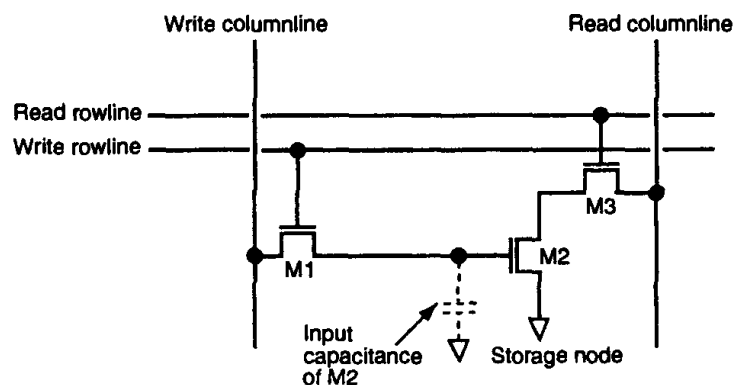


Figure 1.8 3-transistor DRAM cell.

If we want to read out the contents of the cell, we begin by first precharging the Read columnline to a known voltage and then driving the Read rowline HIGH. Driving the Read rowline HIGH turns M3 ON and allows M2 either to pull the Read columnline LOW or to not change the precharged voltage of the Read columnline. (If M2's gate is a logic LOW, then M2 will be OFF, having no effect on the state of the Read columnline.) The main drawback of using the 3-transistor DRAM cell, and the reason it is no longer used, is that it requires two pairs of column and rowlines and a large layout area. Modern 1-transistor, 1-capacitor DRAM cells use a single rowline, a single columnline, and considerably less area.

1.1.2 The 4k–64 Meg DRAM (Second Generation)

We distinguish second-generation DRAMs from first-generation DRAMs by the introduction of multiplexed address inputs, multiple memory

arrays, and the 1-transistor/1-capacitor memory cell. Furthermore, second-generation DRAMs offer more modes of operation for greater flexibility or higher speed operation. Examples are *page mode*, *nibble mode*, *static column mode*, *fast page mode* (FPM), and *extended data out* (EDO). Second-generation DRAMs range in size from 4k (4,096 x 1 bit, i.e., 4,096 address locations with 1-bit input/output word size) up to 64 Meg (67,108,864 bits) in memory sizes of 16 Meg x 4 organized as 16,777,216 address locations with 4-bit input/output word size, 8 Meg x 8, or 4 Meg x 16.

Two other major changes occurred in second-generation DRAMs: (1) the power supply transitioned to a single 5 V and (2) the technology advanced from NMOS to CMOS. The change to a single 5 V supply occurred at the 64kbit density. It simplified system design to a single power supply for the memory, processor, and any TTL logic used in the system. As a result, rowlines had to be driven to a voltage greater than 5 V to turn the NMOS access devices fully ON (more on this later), and the substrate held at a potential less than zero. For voltages outside the supply range, charge pumps are used (see Chapter 6). The move from NMOS to CMOS, at the 1Mb density level, occurred because of concerns over speed, power, and layout size. At the cost of process complexity, complementary devices improved the design.

1.1.2.1 Multiplexed Addressing. Figure 1.9 shows a 4k DRAM block diagram, while Figure 1.10 shows the pin connections for a 4k chip. Note that compared to the block diagram of the 1k DRAM shown in Figure 1.1, the number of address input pins has decreased from 10 to 6, even though the memory size has quadrupled. This is the result of using multiplexed addressing in which the same address input pins are used for both the row and column addresses. The row address strobe ($\overline{RAS}$) input clocks the address present on the DRAM address pins A_0 to A_5 into the row address latches on the falling edge. The column address strobe ($\overline{CAS}$) input clocks the input address into the column address latches on its falling edge.

Figure 1.11 shows the timing relationships between $\overline{RAS}$, $\overline{CAS}$, and the address inputs. Note that t_{RC} is still (as indicated in the last section) the random cycle time for the DRAM, indicating the maximum rate we can write to or read from a DRAM. Note too how the row (or column) address must be present on the address inputs when $\overline{RAS}$ (or $\overline{CAS}$) goes LOW. The parameters t_{RAS} and t_{CAS} indicate how long $\overline{RAS}$ or $\overline{CAS}$ must remain LOW after clocking in a column or row address. The parameters t_{ASR} , t_{RAH} , t_{ASC} , and t_{CAH} indicate the setup and hold times for the row and column addresses, respectively.

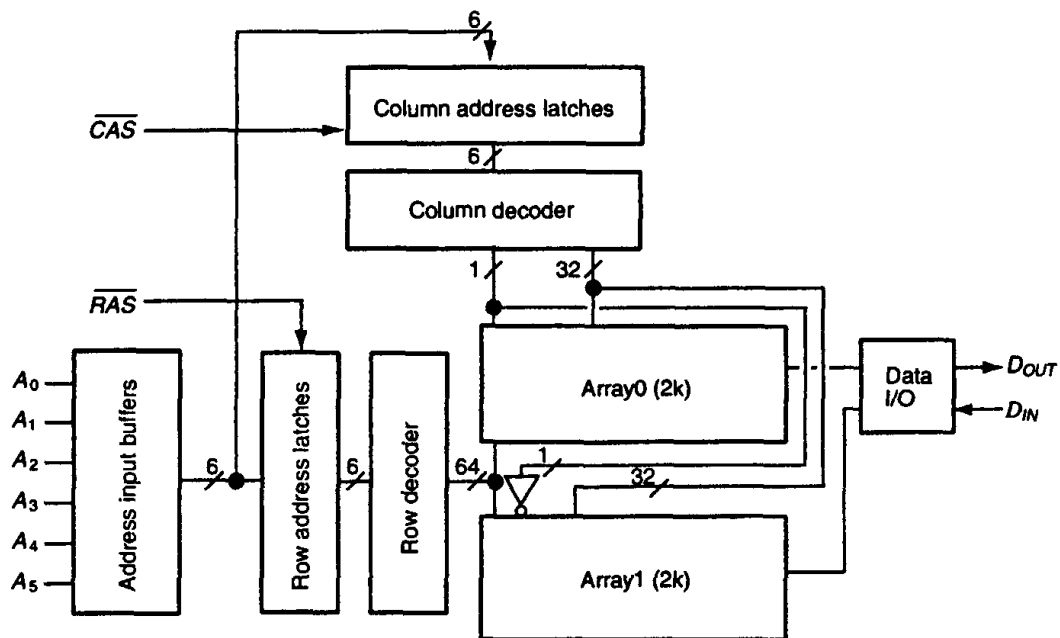


Figure 1.9 Block diagram of a 4k DRAM.

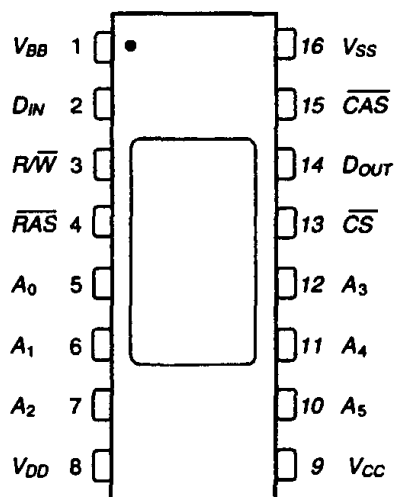


Figure 1.10 4,096-bit DRAM pin connections.

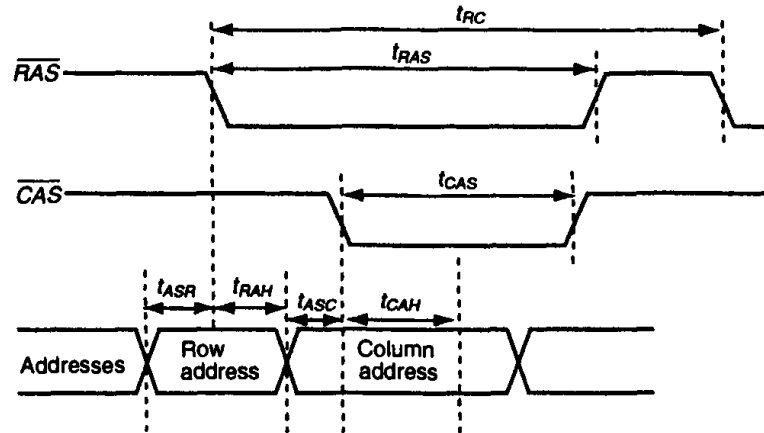


Figure 1.11 Address timing.

1.1.2.2 Multiple Memory Arrays. As mentioned earlier, second-generation DRAMs began to use multiple or segmented memory arrays. The main reason for splitting up the memory into more than one array at the cost of a larger layout area can be understood by considering the parasitics present in the dynamic memory circuit element. To understand the origins of these parasitics, consider the modern DRAM memory cell comprising one MOSFET and one capacitor, as shown in Figure 1.12.

In the next section, we cover the operation of this cell in detail. Here we introduce the operation of the cell. Data is written to the cell by driving the rowline (a.k.a., wordline) HIGH, turning ON the MOSFET, and allowing the columnline (a.k.a., digitline or bitline) to charge or discharge the storage capacitor. After looking at this circuit for a moment, we can make the following observations:

1. The wordline (rowline) may be fabricated using polysilicon (poly). This allows the MOSFET to be formed by crossing the poly wordline over an n+ active area.
2. To write a full V_{CC} logic voltage (where V_{CC} is the maximum positive power supply voltage) to the storage capacitor, the rowline must be driven to a voltage greater than $V_{CC} +$ the n-channel MOSFET threshold voltage (with body effect). This voltage, $> V_{CC} + V_{TH}$, is often labeled V_{CC} pumped (V_{CCP}).
3. The bitline (columnline) may be made using metal or polysilicon. The main concern, as we'll show in a moment, is to reduce the parasitic capacitance associated with the bitline.

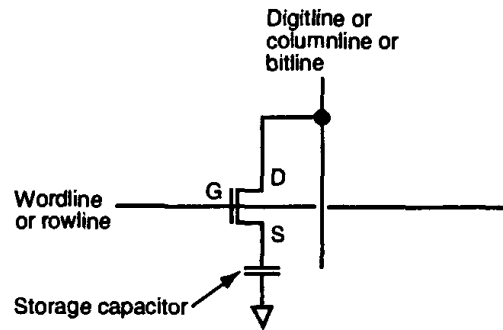


Figure 1.12 1-transistor, 1-capacitor (1T1C) memory cell.

Consider the row of N dynamic memory elements shown in Figure 1.13. Typically, in a modern DRAM, N is 512, which is also the number of bitlines. When a row address is strobed into the DRAM, via the address input pins using the falling edge of $\overline{RAS}$, the address is decoded to drive a wordline (rowline) to V_{CCP} . *This turns ON an entire row in a DRAM memory array.* Turning ON an entire row in a DRAM memory array allows the information stored on the capacitors to be sensed (for a Read) via the bitlines or allows the charging or discharging, via the bitlines, of the storage capacitors (for a Write). Opening a row of data by driving a wordline HIGH is a *very important* concept for understanding the modes of DRAM operation. For Refresh, we only need to supply row addresses during a Refresh operation. For page Reads—when a row is open—a large amount of data, which is set by the number of columns in the DRAM array, can be accessed by simply changing the column address.

We're now in a position to answer the question: "Why are we limited to increasing the number of columnlines (or bitlines) used in a memory array?" or "Why do we need to break up the memory into smaller memory arrays?" The answer to these questions comes from the realization that the more bitlines we use in an array, the longer the delay through the wordline (see Figure 1.13).

If we drive the wordline on the left side of Figure 1.13 HIGH, the signal will take a finite time to reach the end of the wordline (the wordline on the right side of Figure 1.13). This is due to the distributed resistance/capacitance structure formed by the resistance of the polysilicon wordline and the capacitance of the MOSFET gates. The delay limits the speed of DRAM operation. To be precise, it limits how quickly a row can be opened and closed. To reduce this RC time, a polycide wordline is formed by adding a silicide, for example, a mixture of a refractory metal such as tungsten with

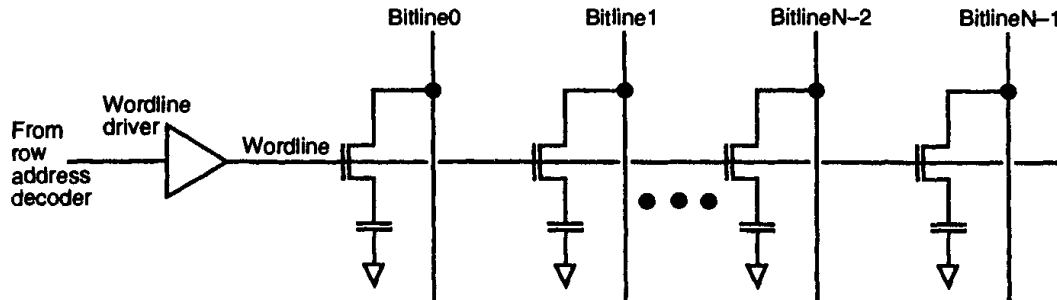


Figure 1.13 Row of N dynamic memory elements.

polysilicon, on top of polysilicon. Using a polycide wordline will have the effect of reducing the wordline resistance. Also, additional drivers can be placed at different locations along the wordline, or the wordline can be *stitched* at various locations with metal.

The limitations on the additional number of wordlines can be understood by realizing that by adding more wordlines to the array, more parasitic capacitance is added to the bitlines. This parasitic capacitance becomes important when sensing the value of data charge stored in the memory element. We'll discuss this in more detail in the next section.

1.1.2.3 Memory Array Size. A comment is in order about memory array size and how addressing can be used for setting word and page size. (We'll explain what this means in a moment.) If we review the block diagram of the 4k DRAM shown in Figure 1.9, we see that two 2k-DRAM memory arrays are used. Each 2k memory is composed of 64 wordlines and 32 bitlines for 2,048 memory elements/address locations per array. In the block diagram, notice that a single bit, coming from the column decoder, can be used to select data, via the bitlines, from Array0 or Array1.

From our discussion earlier, we can open a row in Array0 while at the same time opening a row in Array1 by simply applying a row address to the input address pins and driving $\overline{RAS}$ LOW. Once the rows are open, it is a simple matter of changing the column address to select different data associated with the same open row from either array. If our word size is 1 bit, we could define a page as being 64 bits in length (32 bits from each array). We could also define our page size as 32 bits with a 2-bit word for input/output. We would then say that the DRAM is a 4k DRAM organized as 2k x 2. Of course, in the 4k DRAM, in which the number of bits is small, the concepts of page reads or size aren't too useful. We present them here simply to illustrate the concepts. Let's consider a more practical and modern configuration.

Suppose we have a 64-Meg DRAM organized as 16 Meg x 4 (4 bits input/output) using 4k row address locations and 4k column address locations (12 bits or pins are needed for each 4k of addressing). If our (sub) memory array size is 256kbits, then we have a total of 256 memory arrays on our DRAM chip. We'll assume that there are 512 wordlines and 512 bitlines (digitlines), so that the memory array is logically square. (However, physically, as we shall see, the array is not square.) Internal to the chip, in the address decoders, we can divide the row and column addresses into two parts: the lower 9 bits for addressing the wordlines/bitlines in a 256k memory array and the upper 3 bits for addressing one of the 64 "group-of-four" memory arrays (6 bits total coming from the upper 3 bits of the row and column addresses).

Our 4-bit word comes from the group-of-four memory arrays (one bit from each memory array). We can define a page of data in the DRAM by realizing that when we open a row in each of the four memory arrays, we are accessing 2k of data (512 bits/array x 4 arrays). By simply changing the column address without changing the row address and thus opening another group-of-four wordlines, we can access the 2k "page" of data. With a little imagination, we can see different possibilities for the addressing. For example, we could open 8 group-of-four memory arrays with a row address and thus increase the page size to 16k, or we could use more than one bit at a time from an array to increase word size.

1.1.2.4 Refreshing the DRAM. Refreshing the DRAM is accomplished by sequentially opening each row in the DRAM. (We'll discuss how the DRAM cell is refreshed in greater detail later in the book.) If we use the 64-Meg example in the last section, we need to supply 4k row addresses to the DRAM by changing the external address inputs from 000000000000 to 111111111111 while clocking the addresses into the DRAM using the falling edge of $\overline{RAS}$. In some DRAMs, an internal row address counter is present to make the DRAM easier to refresh. The general specification for 64-Meg DRAM Refresh is that all rows must be refreshed at least every 64 ms, which is an average of 15.7 μ s per row. This means, that if the Read cycle time t_{RC} is 100 ns (see Figure 1.11), it will take $4,096 \cdot 100$ ns or 410 μ s to refresh a DRAM with 4k of row addresses. The percentage of time the DRAM is unavailable due to Refresh can be calculated as 410 μ s/64 ms or 0.64% of the time. Note that the Refresh can be a burst, taking 410 μ s as just described, or distributed, where a row is refreshed every 15.7 μ s.

1.1.2.5 Modes of Operation. From the last section, we know that we can open a row in one or more DRAM arrays concurrently, allowing a page

of data to be written to or read from the DRAM. In this section, we look at the different modes of operation possible for accessing this data via the column address decoder. Our goal in this section is not to present all possible modes of DRAM operation but rather to discuss the modes that have been used in second-generation DRAMs. These modes are page mode, nibble mode, static column mode, fast page mode, and extended data out.

Figure 1.14 shows the timing diagram for a page mode Read, Write, and Read-Modify-Write. We can understand this timing diagram by first noticing that when $\overline{RAS}$ goes LOW, we clock in a row address, decode the row address, and then drive a wordline in one or more memory arrays to V_{CCP} . The result is an open row(s) of data sitting on the digitlines (columnlines). *Only one row can be opened in any single array at a time.* Prior to opening a row, the bitlines are precharged to a known voltage. (Precharging to $V_{CC}/2$ is typically performed using internal circuitry.) Also notice at this time that data out, D_{OUT} , is in a Hi-Z state; that is, the DRAM is not driving the bus line connected to the D_{OUT} pin.

The next significant timing event occurs when $\overline{CAS}$ goes LOW and the column address is clocked into the DRAM (Figure 1.14). At this time, the column address is decoded, and, assuming that the data from the open row is sitting on the digitlines, it is steered using the column address decoder to D_{OUT} . We may have an open row of 512 bits, but we are steering only one bit to D_{OUT} . Notice that when $\overline{CAS}$ goes HIGH, D_{OUT} goes back to the Hi-Z state.

By strobing in another column address with the same open row, we can select another bit of data (again via the column address decoder) to steer to the D_{OUT} pin. In this case, however, we have changed the DRAM to the Write mode (Figure 1.14). This allows us to write, with the same row open via the D_{IN} pin in Figure 1.10, to any column address on the open row. *Later, second-generation DRAMs used the same pins for both data input and output to reduce pin count. These bidirectional pins are labeled DQ.*

The final set of timing signals in Figure 1.14 (the right side) read data out of the DRAM with $R/\overline{W}$ HIGH, change $R/\overline{W}$ to a LOW, and then write to the same location. Again, when $\overline{CAS}$ goes HIGH, D_{OUT} goes back to the Hi-Z state.

The remaining modes of operation are simple modifications of page mode. As seen in Figure 1.15, FPM allows the column address to change while $\overline{CAS}$ is LOW. The speed of the DRAM improves by reducing the delay between $\overline{CAS}$ going LOW and valid data present, or accessed, on D_{OUT} (t_{CAC}). EDO is simply an FPM DRAM that doesn't force D_{OUT} to a Hi-Z state when $\overline{CAS}$ goes HIGH. The data out of the DRAM is thus available

for a longer period of time, allowing for faster operation. In general, opening the row is the operation that takes the longest amount of time. Once a row is open, the data sitting on the columnlines can be steered to D_{OUT} at a fast rate. Interestingly, using column access modes has been the primary method of boosting DRAM performance over the years.

The other popular modes of operation in second-generation DRAMs were the static column and nibble modes. Static column mode DRAMs used flow-through latches in the column address path. When a column address was changed externally, with $\overline{CAS}$ LOW, the column address fed directly to the column address decoder. (The address wasn't clocked on the falling edge of $\overline{CAS}$.) This increased the speed of the DRAM by preventing the outputs from going into the Hi-Z state with changes in the column address.

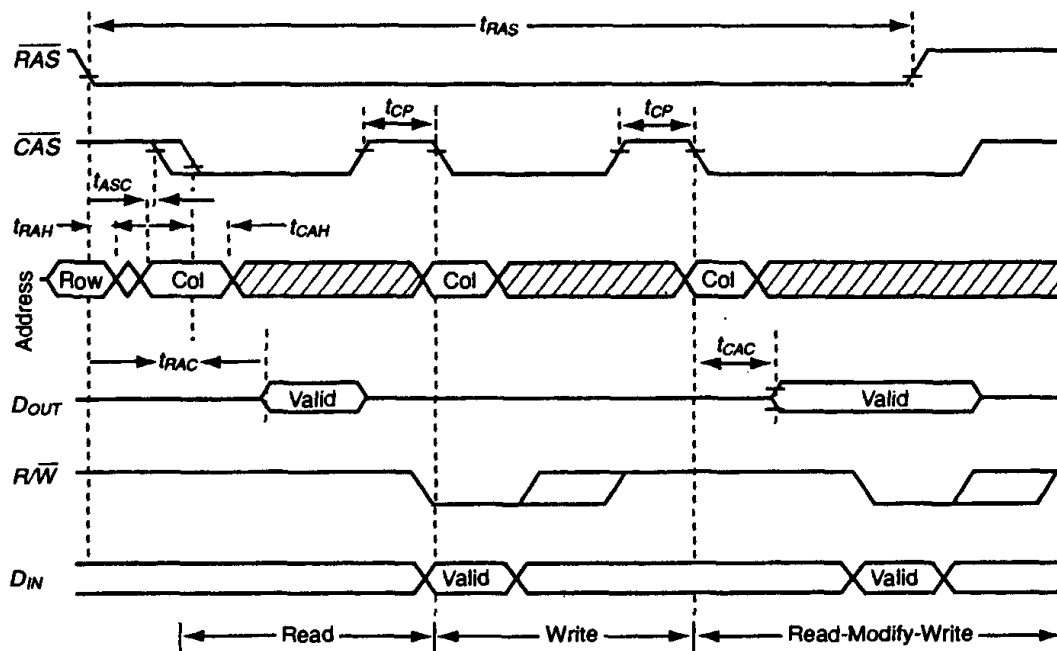


Figure 1.14 Page mode.

Nibble mode DRAMs used an internal presettable address counter so that by strobing $\overline{CAS}$, the column address would change internally. Figure 1.16 illustrates the timing operation for a nibble mode DRAM. The first time $\overline{CAS}$ transitions LOW (first being defined as the first transition after $\overline{RAS}$ goes LOW), the column address is loaded into the counter. If $\overline{RAS}$ is held LOW and $\overline{CAS}$ is toggled, the internal address counter is incremented, and the sequential data appears on the output of the DRAM. The term *nibble mode* comes from limiting the number of $\overline{CAS}$ cycles to four (a nibble).

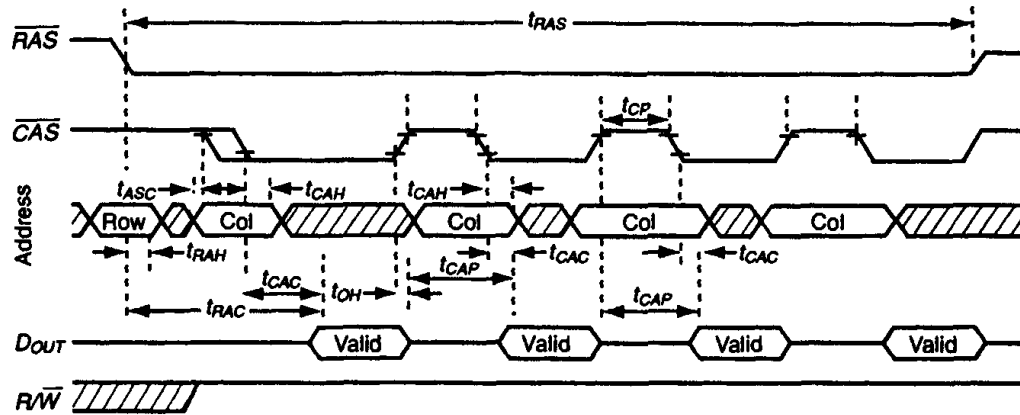


Figure 1.15 Fast page mode.

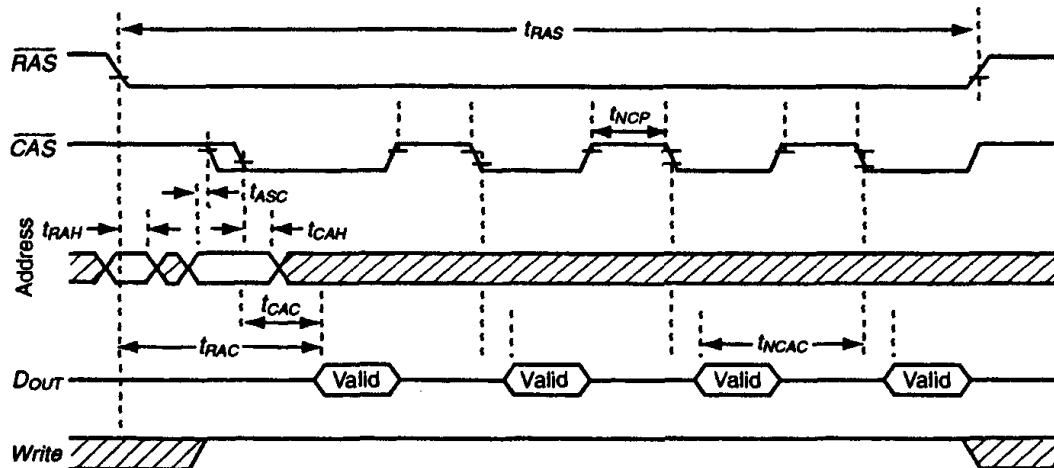


Figure 1.16 Nibble mode.

1.1.3 Synchronous DRAMs (Third Generation)

Synchronous DRAMs (SDRAMs) are made by adding a synchronous interface between the basic core DRAM operation/circuitry of second-generation DRAMs and the control coming from off-chip to make the DRAM operation faster. All commands and operations to and from the DRAM are executed on the rising edge of a master or command clock signal that is common to all SDRAMs and labeled *CLK*. See Figure 1.17 for the pin connections of a 64Mb SDRAM with 16-bit input/output (I/O).

At the time of this writing, SDRAMs operate with a maximum *CLK* frequency in the range of 100–143 MHz. This means that if a 64Mb SDRAM

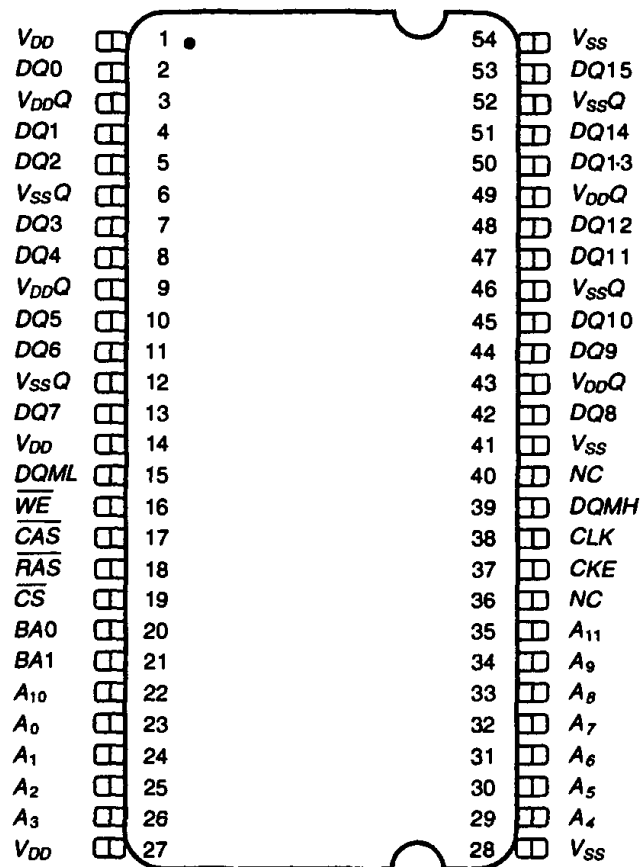


Figure 1.17 Pin connections of a 64Mb SDRAM with 16-bit I/O.

is organized as a x16 part (that is, the input/output word size is 16 bits), the maximum rate at which the words can be written to the part is 200–286 MB/s.

Another variation of the SDRAM is the *double-data-rate SDRAM* (DDR SDRAM, or simply DDR DRAM). The DDR parts register commands and operations on the rising edge of the clock signal while allowing data to be transferred on both the rising and falling edges. A differential input clock signal is used in the DDR DRAM with the labeling of, not surprisingly, *CLK* and *CLK*. In addition, the DDR DRAM provides an output data strobe, labeled DQS, synchronized with the output data and the input *CLK*. DQS is used at the controller to strobe in data from a DRAM. The big benefit of using a DDR part is that the data transfer rate can be twice the clock frequency because data can be transferred on both the rising and falling edges of *CLK*. This means that when using a 133 MHz clock, the data written to and read from the DRAM can be transferred at 266M words/s. Using the numbers from the previous paragraph, this means that a 64Mb DDR

SDRAM with an input/output word size of 16 bits will transfer data to and from the memory controller at 400–572 MB/s.

Figure 1.18 shows the block diagram of a 64Mb SDRAM with 16-bit I/O. Note that although CLK is now used for transferring data, we still have the second-generation control signals $\overline{CS}$, $\overline{WE}$, $\overline{CAS}$, and $\overline{RAS}$ present on the part. (CKE is a clock enable signal which, unless otherwise indicated, is assumed HIGH.) Let's discuss how these control signals are used in an SDRAM by recalling that in a second-generation DRAM, a Write was executed by first driving $\overline{WE}$ and $\overline{CS}$ LOW. Next a row was opened by applying a row address to the part and then driving $\overline{RAS}$ LOW. (The row address is latched on the falling edge of $\overline{RAS}$.) Finally, a column address was applied and latched on the falling edge of $\overline{CAS}$. A short time later, the data applied to the part would be written to the accessed memory location.

For the SDRAM Write, we change the syntax of the descriptions of what's happening in the part. However, the fundamental operation of the DRAM circuitry is the same as that of the second-generation DRAMs. We can list these syntax changes as follows:

1. The memory is segmented into banks. For the 64Mb memory of Figure 1.17 and Figure 1.18, each bank has a size of 16Mbs (organized as 4,096 row addresses [12 bits] x 256 column addresses [8 bits] x 16 bits [16 DQ I/O pins]). As discussed earlier, this is nothing more than a simple logic design of the address decoder (although in most practical situations, the banks are also laid out so that they are physically in the same area). The bank selected is determined by the addresses $BA0$ and $BA1$.
2. In second-generation DRAMs, we said, "We open a row," as discussed earlier. In SDRAM, we now say, "We activate a row in a bank." We do this by issuing an active command to the part. Issuing an active command is accomplished on the rising edge of CLK with a row/bank address applied to the part with $\overline{CS}$ and $\overline{RAS}$ LOW, while $\overline{CAS}$ and $\overline{WE}$ are held HIGH.
3. In second-generation DRAMs, we said, "We write to a location given by a column address," by driving $\overline{CAS}$ LOW with the column address

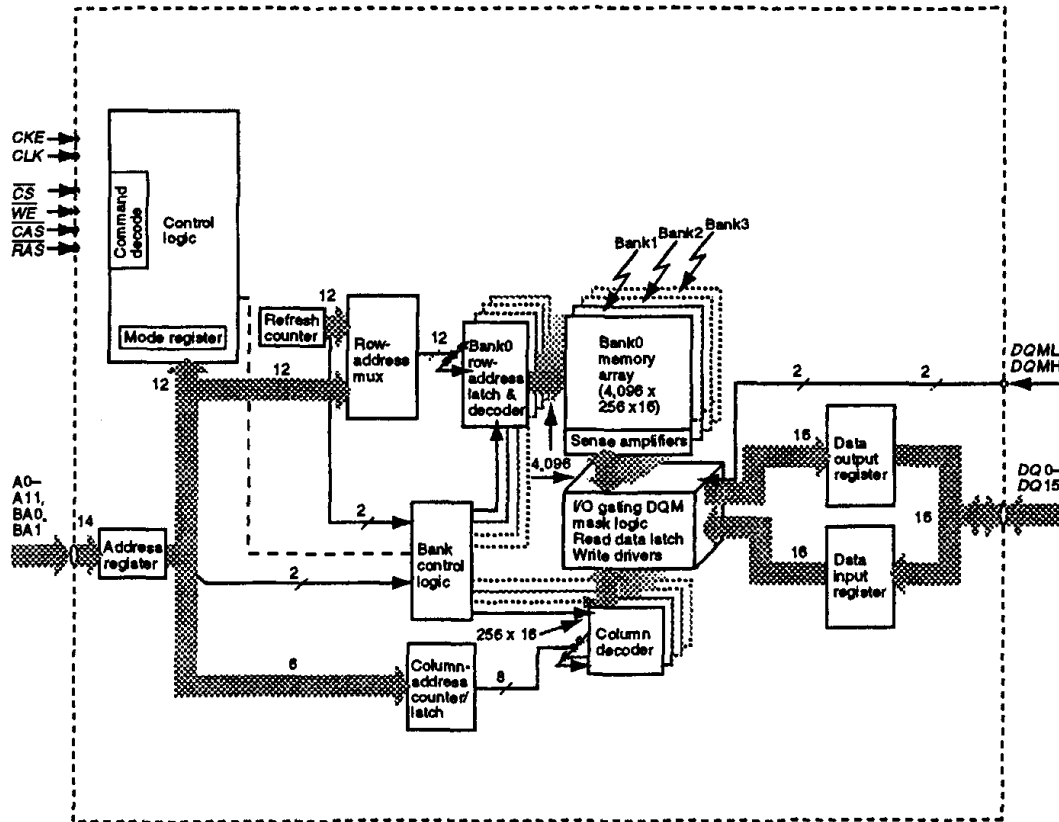


Figure 1.18 Block diagram of a 64Mb SDRAM with 16-bit I/O.

applied to the part and then applying data to the part. In an SDRAM, we write to the part by issuing the Write command to the part. Issuing a Write command is accomplished on the rising edge of *CLK* with a column/bank address applied to the part: $\overline{CS}$, $\overline{CAS}$, and $\overline{WE}$ are held LOW, and $\overline{RAS}$ is held HIGH.

Table 1.1 shows the commands used in an SDRAM. In addition, this table shows how inputs/outputs (*DQs*) can be masked using the *DQ* mask (*DQM*) inputs. This feature is useful when the DRAM is used in graphics applications.

Table 1.1 SDRAM commands. (Notes: 1)

Name	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	DQM	ADDR	DQs	Notes
Command inhibit (NOP)	H	X	X	X	X	X	X	—
No operation (NOP)	L	H	H	H	X	X	X	—
Active (select bank and activate row)	L	L	H	H	X	Bank/row	X	3
Read (select bank and column, and start Read burst)	L	H	L	H	L/H ⁸	Bank/col	X	4
Write (select bank and column, and start Write burst)	L	H	L	L	L/H ⁸	Bank/col	Valid	4
Burst terminate	L	H	H	L	X	X	Active	—
PRECHARGE (deactive row in bank or banks)	L	L	H	L	X	Code	X	5
Auto-Refresh or Self-Refresh (enter Self-Refresh mode)	L	L	L	H	X	X	X	6, 7
Load mode register	L	L	L	L	X	Op-code	X	2
Write Enable/output Enable	—	—	—	—	L	—	Active	8
Write inhibit/output Hi-Z	—	—	—	—	H	—	Hi-Z	8

Notes

1. CKE is HIGH for all commands shown except for Self-Refresh.
2. A0–A11 define the op-code written to the mode register.
3. A0–A11 provide row address, and BA0, BA1 determine which bank is made active.
4. A0–A9 (x4), A0–A8 (x8), or A0–A7 (x16) provide column address; A10 HIGH enables the auto PRECHARGE feature (nonpersistent), while A10 LOW disables the auto PRECHARGE feature; BA0, BA1 determine which bank is being read from or written to.

5. A10 LOW: $BA0$, $BA1$ determine the bank being precharged. A10 HIGH: all banks precharged and $BA0$, $BA1$ are “don’t care.”
6. This command is Auto-Refresh if CKE is HIGH and Self-Refresh if CKE is LOW.
7. Internal Refresh counter controls row addressing; all inputs and I/Os are “don’t care” except for CKE .
8. Activates or deactivates the DQs during Writes (zero-clock delay) and Reads (two-clock delay).

SDRAMs often employ pipelining in the address and data paths to increase operating speed. Pipelining is an effective tool in SDRAM design because it helps disconnect operating frequency and access latency. Without pipelining, a DRAM can only process one access instruction at a time. Essentially, the address is held valid internally until data is fetched from the array and presented to the output buffers. This single instruction mode of operation ties operating frequency and access time (or latency) together. However, with pipelining, additional access instructions can be fed into the SDRAM before prior access instructions have completed, which permits access instructions to be entered at a higher rate than would otherwise be allowed. Hence, pipelining increases operating speed.

Pipeline stages in the data path can also be helpful when synchronizing output data to the system clock. $\overline{CAS}$ latency refers to a parameter used by the SDRAM to synchronize the output data from a Read request with a particular edge of the system clock. A typical Read for an SDRAM with $\overline{CAS}$ latency set to three is shown in Figure 1.19. SDRAMs must be capable of reliably functioning over a range of operating frequencies while maintaining a specified $\overline{CAS}$ latency. This is often accomplished by configuring the pipeline stage to register the output data to a specific clock edge, as determined by the $\overline{CAS}$ latency parameter.

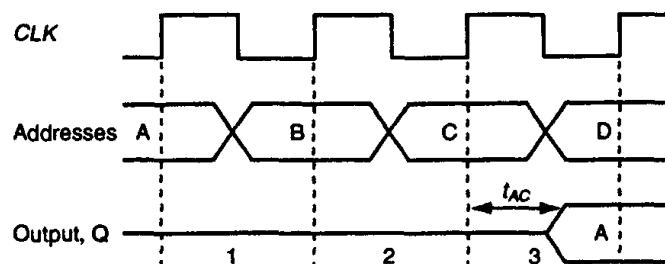


Figure 1.19 SDRAM with a latency of three.

At this point, we should understand the basics of SDRAM operation, but we may be asking, “Why are SDRAMs potentially faster than second-gener-

ation DRAMs such as EDO or FPM?” The answer to this question comes from the realization that it’s possible to activate a row in one bank and then, while the row is opening, perform an operation in some other bank (such as reading or writing). In addition, one of the banks can be in a *PRECHARGE* mode (the bitlines are driven to $V_{CC}/2$) while accessing one of the other banks and, thus, in effect hiding *PRECHARGE* and allowing data to be continuously written to or read from the SDRAM. (Of course, this depends on which application and memory address locations are used.) We use a mode register, as shown in Figure 1.20, to put the SDRAM into specific modes of operation for programmable operation, including pipelining and burst Reads/Writes of data [2].

1.2 DRAM BASICS

A modern DRAM memory cell or memory bit (mbit), as shown in Figure 1.21, is formed with one transistor and one capacitor, accordingly referred to as a 1T1C cell. The mbit is capable of holding binary information in the form of stored charge on the capacitor. The mbit transistor operates as a switch interposed between the mbit capacitor and the digitline. Assume that the capacitor’s common node is biased at $V_{CC}/2$, which we will later show as a reasonable assumption. Storing a logic one in the cell requires a capacitor with a voltage of $+V_{CC}/2$ across it. Therefore, the charge stored in the mbit capacitor is

$$Q = \frac{V_{CC}}{2} \cdot C,$$

where C is the capacitance value in farads. Conversely, storing a logic zero in the cell requires a capacitor with a voltage of $-V_{CC}/2$ across it. Note that the stored charge on the mbit capacitor for a logic zero is

$$Q = \frac{-V_{CC}}{2} \cdot C.$$

The charge is negative with respect to the $V_{CC}/2$ common node voltage in this state. Various leakage paths cause the stored capacitor charge to slowly deplete. To return the stored charge and thereby maintain the stored data state, the cell must be refreshed. The required refreshing operation is what makes DRAM memory dynamic rather than static.

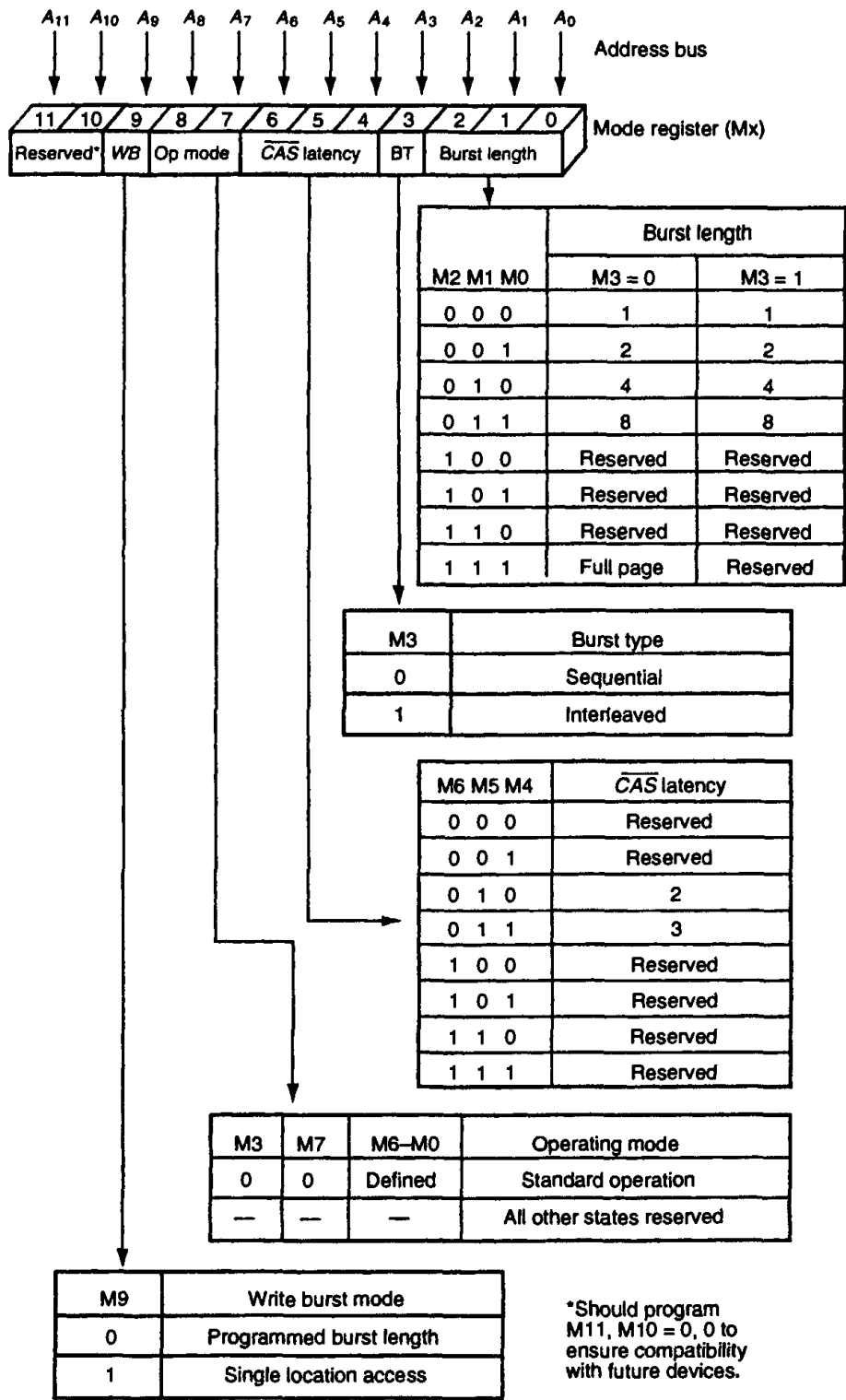


Figure 1.20 Mode register.

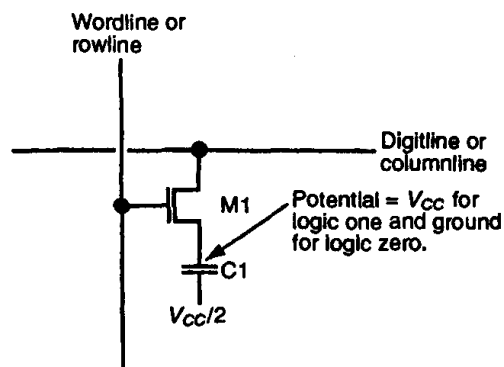


Figure 1.21 IT1C DRAM memory cell.
(Note the rotation of the rowline and columnline.)

The digitline referred to earlier consists of a conductive line connected to a multitude of mbit transistors. The conductive line is generally constructed from either metal or silicide/polycide polysilicon. Because of the quantity of mbits connected to the digitline and its physical length and proximity to other features, the digitline is highly capacitive. For instance, a typical value for digitline capacitance on a 0.35 μm process might be 300fF. Digitline capacitance is an important parameter because it dictates many other aspects of the design. We discuss this further in Section 2.1. For now, we continue describing basic DRAM operation.

The mbit transistor gate terminal is connected to a wordline (rowline). The wordline, which is connected to a multitude of mbits, is actually formed of the same polysilicon as that of the transistor gate. The wordline is physically orthogonal to the digitline. A memory array is formed by tiling a selected quantity of mbits together such that mbits along a given digitline do not share a common wordline and mbits along a common wordline do not share a common digitline. Examples of this are shown in Figures 1.22 and 1.23. In these layouts, mbits are paired to share a common contact to the digitline, which reduces the array size by eliminating duplication.

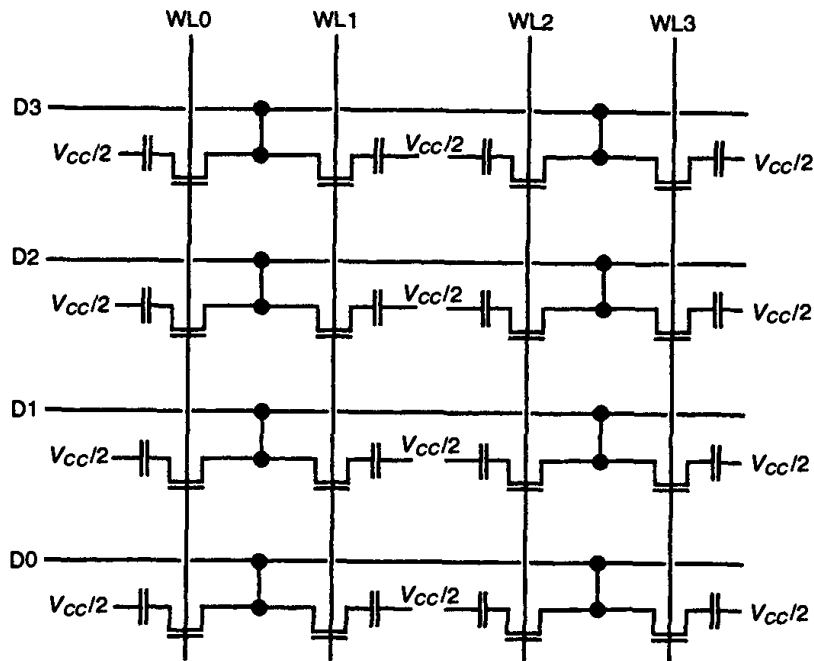


Figure 1.22 Open digitline memory array schematic.

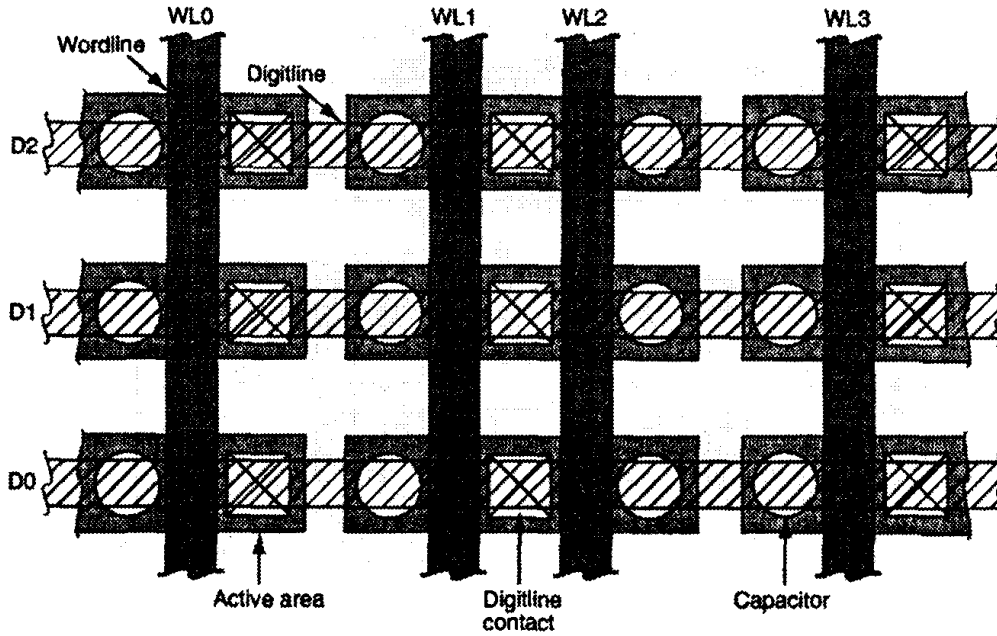


Figure 1.23 Open digitline memory array layout.

1.2.1 Access and Sense Operations

Next, we examine the access and sense operations. We begin by assuming that the cells connected to D1, in Figure 1.24, have logic one levels ($+V_{CC}/2$) stored on them and that the cells connected to D0 have logic zero levels ($-V_{CC}/2$) stored on them. Next, we form a digitline pair by considering two digitlines from adjacent arrays. The digitline pairs, labeled D0/D0* and D1/D1*, are initially equilibrated to $V_{CC}/2$ V. All wordlines are initially at 0 V, ensuring that the mbit transistors are OFF. Prior to a wordline firing, the digitlines are electrically disconnected from the $V_{CC}/2$ bias voltage and allowed to float. They remain at the $V_{CC}/2$ *PRECHARGE* voltage due to their capacitance.

To read mbit1, wordline WL0 changes to a voltage that is at least one transistor V_{TH} above V_{CC} . This voltage level is referred to as V_{CCP} or V_{PP} . To ensure that a full logic one value can be written back into the mbit capacitor, V_{CCP} must remain greater than one V_{TH} above V_{CC} . The mbit capacitor begins to discharge onto the digitline at two different voltage levels depending on the logic level stored in the cell. For a logic one, the capacitor begins to discharge when the wordline voltage exceeds the digitline *PRECHARGE* voltage by V_{TH} . For a logic zero, the capacitor begins to discharge when the wordline voltage exceeds V_{TH} . Because of the finite rise time of the wordline voltage, this difference in turn-on voltage translates into a significant delay when reading ones, as seen in Figure 1.25.

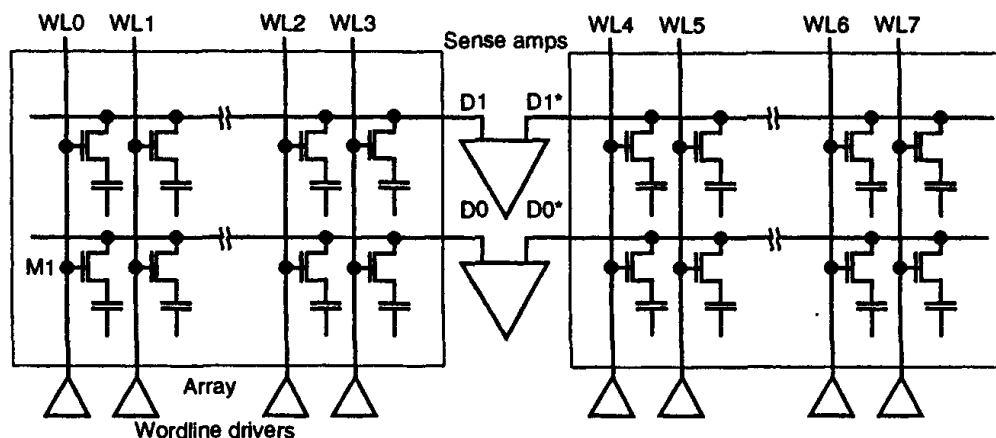


Figure 1.24 Simple array schematic (an open DRAM array).

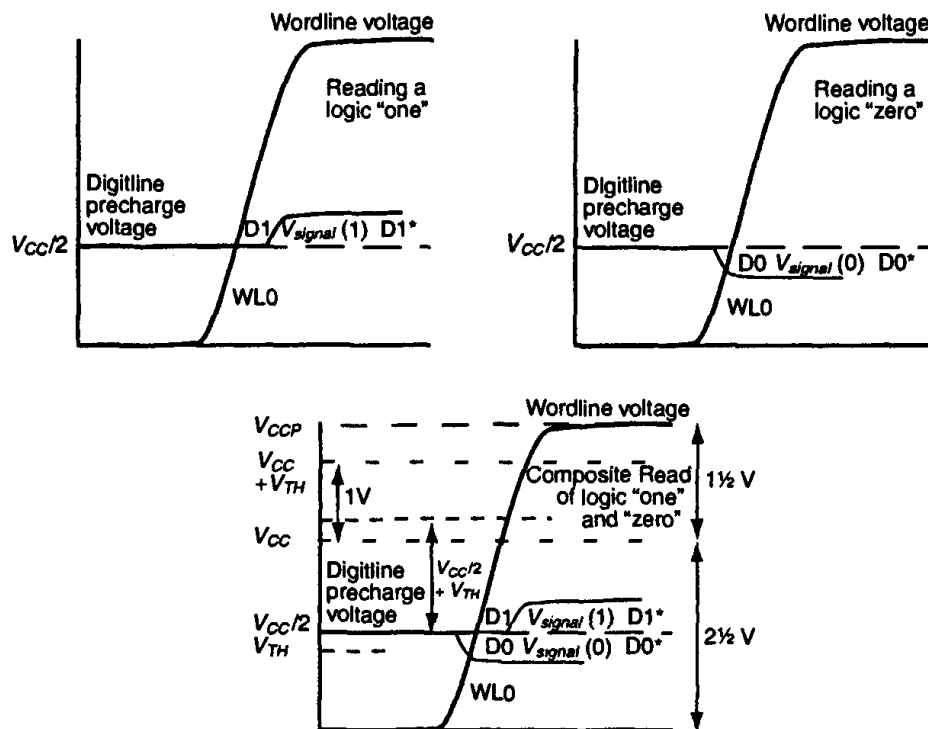


Figure 1.25 Cell access waveforms.

Accessing a DRAM cell results in charge-sharing between the mbit capacitor and the digitline capacitance. This charge-sharing causes the digitline voltage either to increase for a stored logic one or to decrease for a stored logic zero. Ideally, only the digitline connected to the accessed mbit will change. In reality, the other digitline voltage also changes slightly, due to parasitic coupling between digitlines and between the firing wordline and the other digitline. (This is especially true for the folded bitline architecture discussed later.) Nevertheless, a differential voltage develops between the two digitlines. The magnitude of this voltage difference, or signal, is a function of the *mbit capacitance* (C_{mbit}), *digitline capacitance* (C_{digit}), and voltage stored on the cell prior to access (V_{cell}). See Figure 1.26. Accordingly,

$$V_{signal} = V_{cell} \cdot \frac{C_{mbit}}{C_{digit} + C_{mbit}}.$$

A V_{signal} of 235mV is yielded from a design in which $V_{cell} = 1.65$, $C_{mbit} = 50\text{fF}$, and $C_{digit} = 300\text{fF}$.

After the cell has been accessed, sensing occurs. Sensing is essentially the amplification of the digitline signal or the differential voltage between the digitlines. Sensing is necessary to properly read the cell data and refresh the mbit cells. (The reason for forming a digitline pair now becomes apparent.) Figure 1.27 presents a schematic diagram for a simplified sense amplifier circuit: a cross-coupled NMOS pair and a cross-coupled PMOS pair. The sense amplifiers also appear like a pair of cross-coupled inverters in which *ACT* and *NLAT** provide power and ground. The NMOS pair or *Nsense-amp* has a common node labeled *NLAT** (for *Nsense-amp latch*).

Similarly, the *Psense-amp* has a common node labeled *ACT* (for *Active pull-up*). Initially, *NLAT** is biased to $V_{CC}/2$, and *ACT* is biased to V_{SS} or signal ground. Because the digitline pair *D1* and *D1** are both initially at $V_{CC}/2$, the *Nsense-amp* transistors are both OFF. Similarly, both *Psense-amp* transistors are OFF. Again, when the mbit is accessed, a signal develops across the digitline pair. While one digitline contains charge from the cell access, the other digitline does not but serves as a reference for the Sensing operation. The sense amplifiers are generally fired sequentially: the *Nsense-amp* first, then the *Psense-amp*. Although designs vary at this point, the higher drive of NMOS transistors and better V_{TH} matching offer better sensing characteristics by *Nsense-amps* and thus lower error probability compared to *Psense-amps*.

Waveforms for the Sensing operation are shown in Figure 1.28. The *Nsense-amp* is fired by bringing *NLAT** (N sense-amp latch) toward ground. As the voltage difference between *NLAT** and the digitlines (*D1* and *D1** in Figure 1.27) approaches V_{TH} , the NMOS transistor whose gate is connected to the higher voltage digitline begins to conduct. This conduction occurs first in the subthreshold and then in the saturation region as the gate-to-source voltage exceeds V_{TH} and causes the low-voltage digitline to discharge toward the *NLAT** voltage. Ultimately, *NLAT** will reach ground and the digitline will be brought to ground potential. Note that the other NMOS transistor will not conduct: its gate voltage is derived from the low-

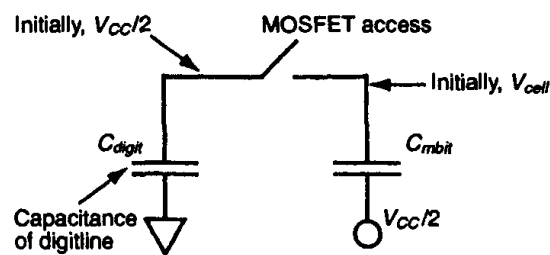


Figure 1.26 DRAM charge-sharing.

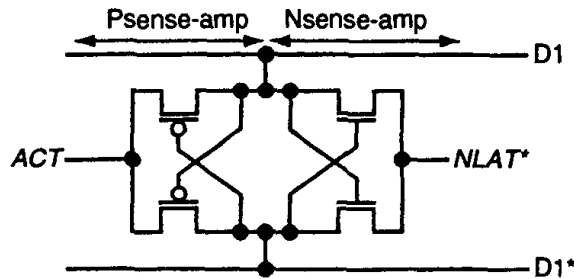


Figure 1.27 Sense amplifier schematic.

voltage digitline, which is being discharged toward ground. In reality, parasitic coupling between digitlines and limited subthreshold conduction by the second transistor result in voltage reduction on the high digitline.

Sometime after the Nsense-amp fires, *ACT* will be brought toward V_{CC} to activate the Psense-amp, which operates in a complementary fashion to the Nsense-amp. With the low-voltage digitline approaching ground, there is a strong signal to drive the appropriate PMOS transistor into conduction. This conduction, again moving from subthreshold to saturation, charges the high-voltage digitline toward *ACT*, ultimately reaching V_{CC} . Because the mbit transistor remains ON, the mbit capacitor is refreshed during the Sensing operation. The voltage, and hence charge, which the mbit capacitor held prior to accessing, is restored to a full level: V_{CC} for a logic one and ground for a logic zero. It should be apparent now why the minimum wordline voltage is a V_{TH} above V_{CC} . If V_{CCP} were anything less, a full V_{CC} level could not be written back into the mbit capacitor. The mbit transistor source voltage V_{source} cannot be greater than $V_{gate} - V_{TH}$ because this would turn OFF the transistor.

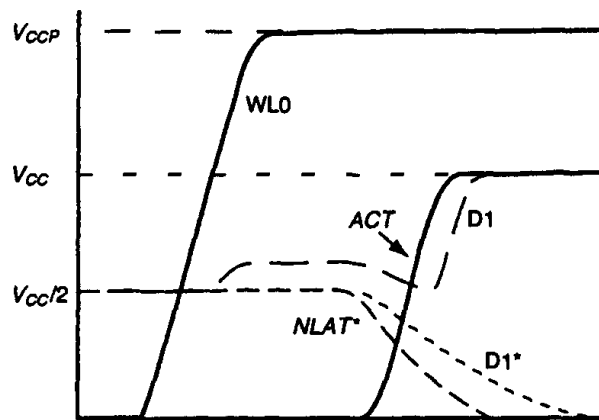


Figure 1.28 Sensing operation waveforms.

1.2.2 Write Operation

A Write operation is similar to a Sensing and Restore operation except that a separate Write driver circuit determines the data that is placed into the cell. The Write driver circuit is generally a tristate inverter connected to the digitlines through a second pair of pass transistors, as shown in Figure 1.29. These pass transistors are referred to as *I/O transistors*. The gate terminals of the *I/O transistors* are connected to a common *column select (CSEL)* signal. The *CSEL* signal is decoded from the column address to select which pair (or multiple pairs) of digitlines is routed to the output pad or, in this case, the Write driver.

In most current DRAM designs, the Write driver simply overdrives the sense amplifiers, which remain ON during the Write operation. After the new data is written into the sense amplifiers, the amplifiers finish the Write cycle by restoring the digitlines to full rail-to-rail voltages. An example is shown in Figure 1.30 in which D1 is initially HIGH after the Sensing operation and LOW after the writing operation. A Write operation usually involves only 2–4 mbits within an array of mbits because a single *CSEL* line is generally connected to only four pairs of *I/O transistors*. The remaining digitlines are accessed through additional *CSEL* lines that correspond to different column address locations.

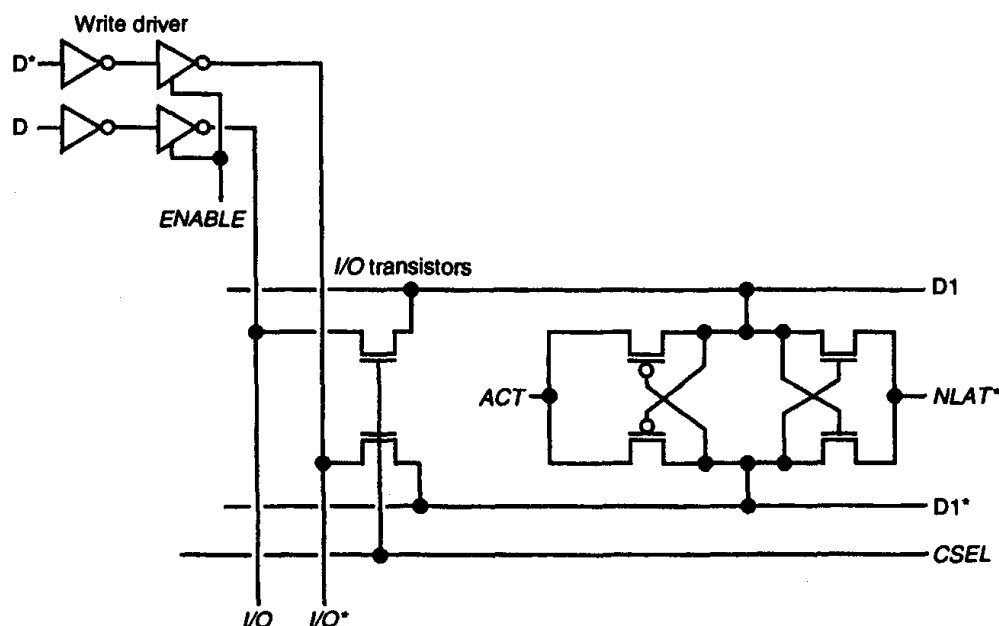


Figure 1.29 Sense amplifier schematic with I/O devices.

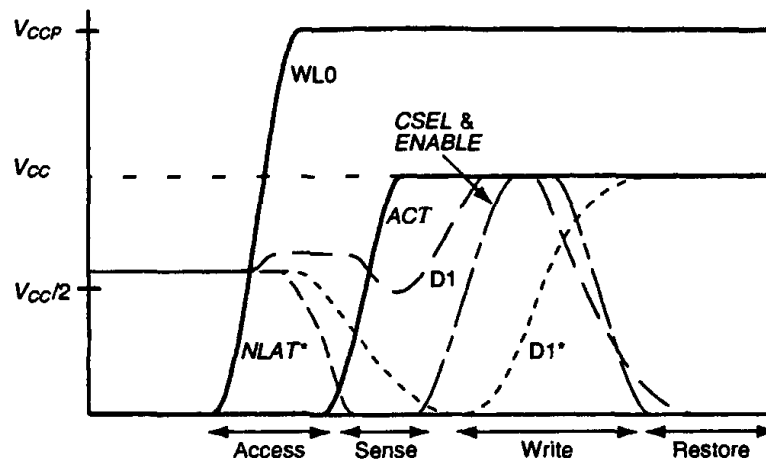


Figure 1.30 Write operation waveforms.

1.2.3 Opening a Row (Summary)

Opening a row of mbits in a DRAM array is a fundamental operation for both reading and writing to the DRAM array. Sometimes the chain of events from a circuit designer's point of view, which lead to an open row, is called the $\overline{RAS}$ timing chain of events below, assuming that for a second-generation DRAM both $\overline{RAS}$ and $\overline{CAS}$ are HIGH. (It's trivial to extend our discussion to third-generation DRAMs where $\overline{RAS}$ and $\overline{CAS}$ are effectively generated from the control logic.)

1. Initially, both $\overline{RAS}$ and $\overline{CAS}$ are HIGH. All bitlines in the DRAM are driven to $V_{CC}/2$, while all wordlines are at 0 V. This ensures that all of the mbit's access transistors in the DRAM are OFF.
2. A valid row address is applied to the DRAM and $\overline{RAS}$ goes LOW. While the row address is being latched, on the falling edge of $\overline{RAS}$, and decoded, the bitlines are disconnected from the $V_{CC}/2$ bias and allowed to float. The bitlines at this point are charged to $V_{CC}/2$, and they can be thought of as capacitors.
3. The row address is decoded and applied to the wordline drivers. This forces only one rowline in at least one memory array to V_{CCP} . Driving the wordline to V_{CCP} turns ON the mbits attached to this rowline and causes charge-sharing between the mbit capacitance and the capacitance of the corresponding bitline. The result is a small perturbation (upwards for a logic one and downwards for a logic zero) in the bitline voltages.

4. The next operation is Sensing, which has two purposes: (a) to determine if a logic one or zero was written to the cell and (b) to refresh the contents of the cell by restoring a full logic zero (0 V) or one (V_{CC}) to the capacitor. Following the wordlines going HIGH, the Nsense-amp is fired by driving, via an n-channel MOSFET, $NLAT^*$ to ground. The inputs to the sense amplifier are two bitlines: the bitline we are sensing and the bitline that is not active (a bitline that is still charged to $V_{CC}/2$ —an inactive bitline). Pulling $NLAT^*$ to ground results in one of the bitlines going to ground. Next, the ACT signal is pulled up to V_{CC} , driving the other bitline to V_{CC} . Some important notes:
 - (a) It doesn't matter if a logic one or logic zero was sensed because the inactive and active bitlines are pulled in opposite directions.
 - (b) The contents of the active cell, after opening a row, are restored to full voltage levels (either 0 V or V_{CC}). The entire DRAM can be refreshed by opening each row.

Now that the row is open, we can write to or read from the DRAM. In either case, it is a simple matter of steering data to or from the active array(s) using the column decoder. When writing to the array, buffers set the new logic voltage levels on the bitlines. The row is still open because the wordline remains HIGH. (The row stays open as long as $\overline{RAS}$ is LOW.)

When reading data out of the DRAM, the values sitting on the bitlines are transmitted to the output buffers via the I/O MOSFETs. To increase the speed of the reading operation, this data, in most situations, is transmitted to the output buffer (sometimes called a DQ buffer) either through a helper flip-flop or another sense amplifier.

A note is in order here regarding the word size stored in or read out of the memory array. We may have 512 active bitlines when a single rowline in an array goes HIGH (keeping in mind once again that only one wordline in an array can go HIGH at any given time). This literally means that we could have a word size of 512 bits from the active array. The inherent wide word size has led to the push, at the time of this writing, of embedding DRAM with a processor (for example, graphics or data). The wide word size and the fact that the word doesn't have to be transmitted off-chip can result in lower-power, higher-speed systems. (Because the memory and processor don't need to communicate off-chip, there is no need for power-hungry, high-speed buffers.)

1.2.4 Open/Folded DRAM Array Architectures

Throughout the book, we make a distinction between the open array architecture as shown in Figures 1.22 and 1.24 and the folded DRAM array used in modern DRAMs and seen in Figure 1.31. At the cost of increased layout area, folded arrays increase noise immunity by moving sense amp inputs next to each other. These sense amp inputs come directly from the DRAM array. The term *folded* comes from taking the DRAM arrays seen in Figure 1.24 and folding them together to form the topology seen in Figure 1.31.

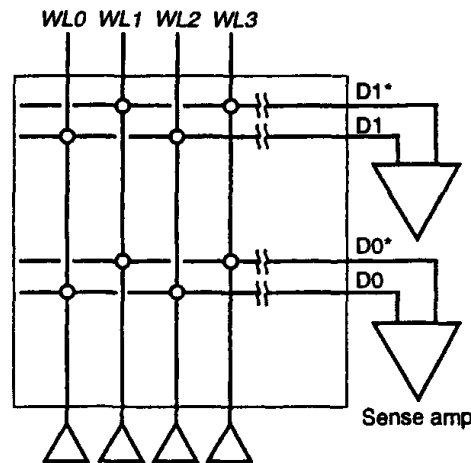


Figure 1.31 A folded DRAM array.

REFERENCES

- [1] R. J. Baker, H. W. Li, and D. E. Boyce, *CMOS: Circuit Design, Layout, and Simulation*. Piscataway, NJ: IEEE Press, 1998.
- [2] Micron Technology, Inc., Synchronous DRAM Data Sheet, 1999.

FOR FURTHER REFERENCE

See the Appendix for additional readings and references.

The DRAM Array

This chapter begins a more detailed examination of standard DRAM array elements. This examination is necessary for a clear understanding of fundamental DRAM elements and how they are used in memory block construction. A common point of reference is required before considering the analysis of competing array architectures. Included in this chapter is a detailed discussion of mbits, array configurations, sense amplifier elements, and row decoder elements.

2.1 THE MBIT CELL

The primary advantage of DRAM over other types of memory technology is low cost. This advantage arises from the simplicity and scaling characteristics of its 1T1C memory cell [1]. Although the DRAM mbit is simple conceptually, its actual design and implementation are highly complex. Therefore, successful, cost-effective DRAM designs require a tremendous amount of process technology.

Figure 2.1 presents the layout of a modern buried capacitor DRAM mbit pair. (*Buried* means that the capacitor is below the digitline.) This type of mbit is also referred to as a *bitline over capacitor* (BOC) cell. Because sharing a contact significantly reduces overall cell size, DRAM mbits are constructed in pairs. In this way, a digitline contact can be shared. The mbits comprise an active area rectangle (in this case, an n+ active area), a pair of polysilicon wordlines, a single digitline contact, a metal or polysilicon digitline, and a pair of cell capacitors formed with an oxide-nitride-oxide dielectric between two layers of polysilicon. For most processes, the wordline polysilicon is silicided to reduce sheet resistance, permitting longer wordline segments without reducing speed. The mbit layout, as shown in Figure

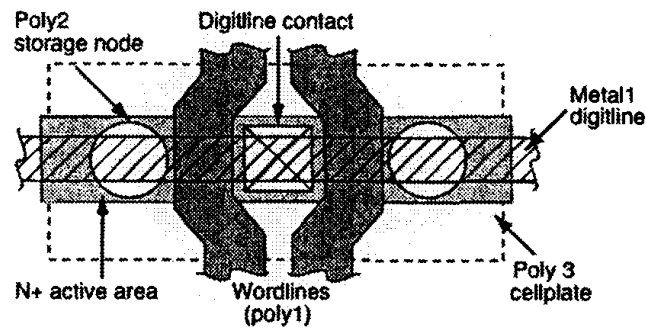


Figure 2.1 Mbit pair layout.

2.1, is essentially under the control of process engineers, for every aspect of the mbit must meet stringent performance and yield criteria.

A small array of mbits appears in Figure 2.2. This figure is useful to illustrate several features of the mbit. First, note that the digitline pitch (width plus space) dictates the active area pitch and the capacitor pitch. Process engineers adjust the active area width and the field oxide width to maximize transistor drive and minimize transistor-to-transistor leakage. Field oxide technology greatly impacts this balance. A thicker field oxide or a shallower junction depth affords a wider transistor active area. Second, the wordline pitch (width plus space) dictates the space available for the digitline contact, transistor length, active area, field poly width, and capacitor length. Optimization of each of these features by process engineers is necessary to maximize capacitance, minimize leakage, and maximize yield. Contact technology, subthreshold transistor characteristics, photolithography, and etch and film technology dictate the overall design.

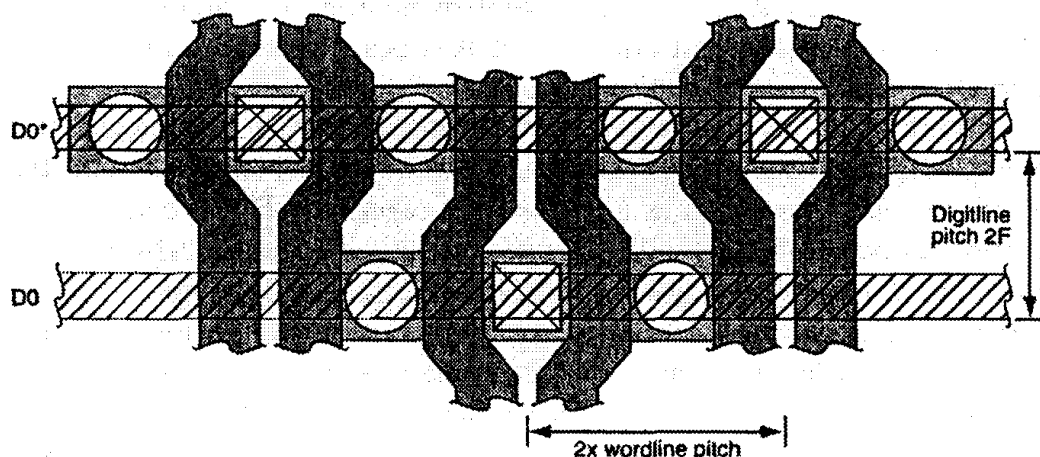


Figure 2.2 Layout to show array pitch.

At this point in the discussion, it is appropriate to introduce the concept of feature size and how it relates to cell size. The mbit shown in Figures 2.1 and 2.2 is by definition an eight-square feature cell ($8F^2$) [2][3]. The intended definition of feature (F) in this case is minimum realizable process dimension but in fact equates to a dimension that is one-half the wordline (row) or digitline (column) pitch. A $0.25\text{ }\mu\text{m}$ process having wordline and digitline pitches of $0.6\text{ }\mu\text{m}$ yields an mbit size that is

$$(8) \cdot (0.3\text{ }\mu\text{m})^2 = 0.72\text{ }\mu\text{m}^2.$$

It is easier to explain the $8F^2$ designation with the aid of Figure 2.3. An imaginary box drawn around the mbit defines the cell's outer boundary. Along the x-axis, this box includes one-half digitline contact feature, one wordline feature, one capacitor feature, one field poly feature, and one-half poly space feature for a total of four features. Along the y-axis, this box contains two one-half field oxide features and one active area feature for a total of two features. The area of the mbit is therefore

$$4F \cdot 2F = 8F^2.$$

The folded array architecture, as shown in Figure 2.2, always produces an $8F^2$ mbit. This results from the fact that each wordline connects or forms a crosspoint with an mbit transistor on every other digitline and must pass around mbit transistors as field poly on the remaining digitlines. The field poly in each mbit cell adds two square features to what would have been a $6F^2$ cell. Although the folded array yields a cell that is 25% larger than other array architectures, it also produces superior signal-to-noise performance, especially when combined with some form of digitline twisting [4]. Superior low-noise performance has made folded array architecture the architecture of choice since the 64kbit generation [5].

A folded array is schematically depicted in Figure 2.4. Sense amplifier circuits placed at the edge of each array connect to both true and complement digitlines (D and D^*) coming from a single array. Optional digitline pair twisting in one or more positions reduces and balances the coupling to adjacent digitline pairs and improves overall signal-to-noise characteristics [4]. Figure 2.5 shows the variety of twisting schemes used throughout the DRAM industry [6].

Ideally, a twisting scheme equalizes the coupling terms from each digitline to all other digitlines, both true and complement. If implemented properly, the noise terms cancel or at least produce only common-mode noise to which the differential sense amplifier is more immune.

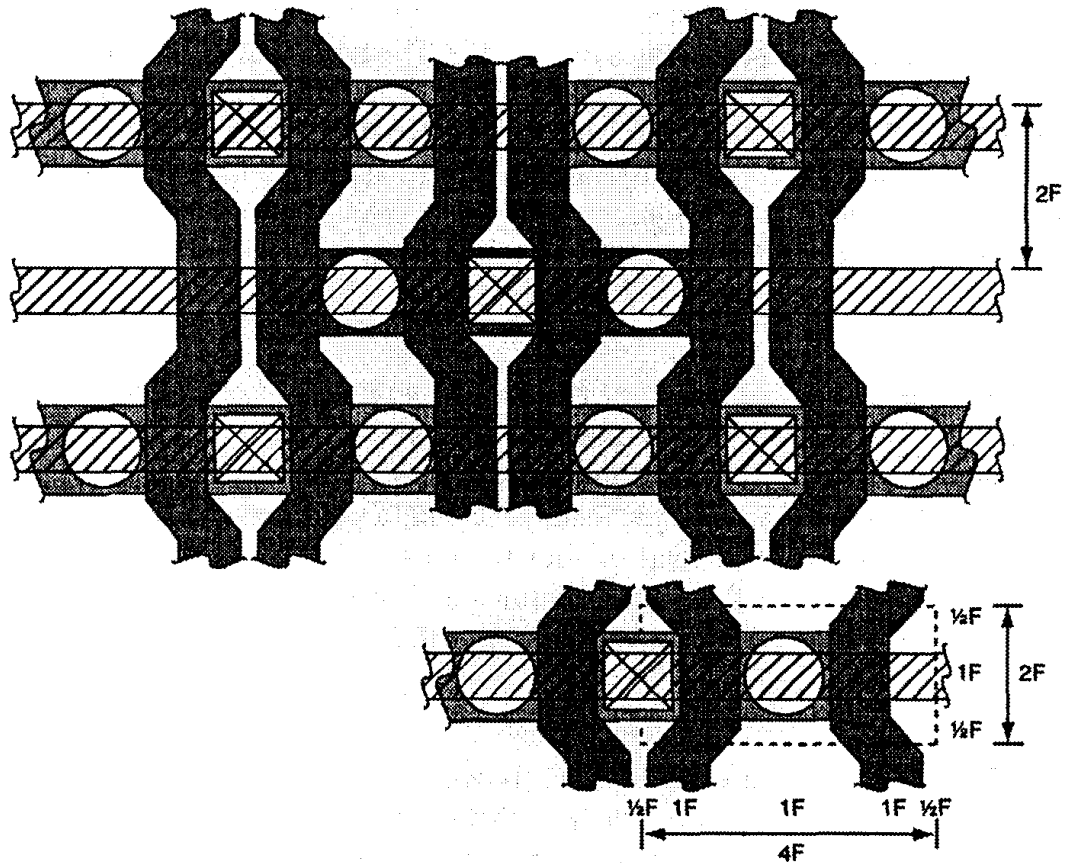


Figure 2.3 Layout to show $8F^2$ derivation.

Each digitline twist region consumes valuable silicon area. Thus, design engineers resort to the simplest and most efficient twisting scheme to get the job done. Because the coupling between adjacent metal lines is inversely proportional to the line spacing, the signal-to-noise problem gets increasingly worse as DRAMs scale to smaller and smaller dimensions. Hence, the industry trend is toward use of more complex twisting schemes on succeeding generations [6][7].

An alternative to the folded array architecture, popular prior to the 64kbit generation [1], was the open digitline architecture. Seen schematically in Figure 2.6, this architecture also features the sense amplifier circuits between two sets of arrays [8]. Unlike the folded array, however, true and complement digitlines (D and D^*) connected to each sense amplifier pair come from separate arrays [9]. This arrangement precludes using digitline twisting to improve signal-to-noise performance, which is the prevalent reason why the industry switched to folded arrays. Note that unlike the folded array architecture, each wordline in an open digitline architecture connects to mbit transistors on every digitline, creating crosspoint-style arrays.

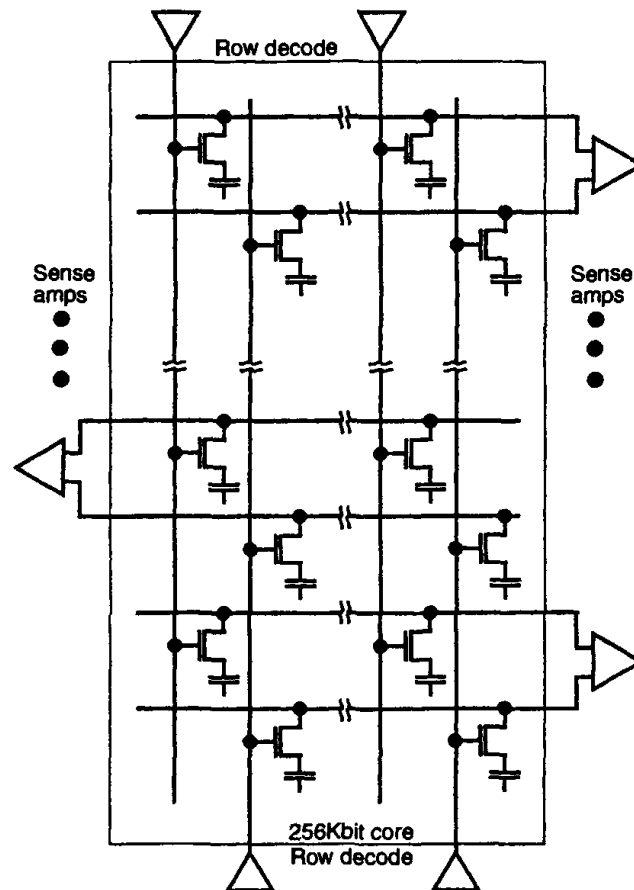


Figure 2.4 Folded digitline array schematic.

This feature permits a 25% reduction in mbit size to only $6F^2$ because the wordlines do not have to pass alternate mbits as field poly. The layout for an array of standard $6F^2$ mbit pairs is shown in Figure 2.7 [2]. A box is drawn around one of the mbits to show the $6F^2$ cell boundary. Again, two mbits share a common digitline contact to improve layout efficiency. Unfortunately, most manufacturers have found that the signal-to-noise problems of open digitline architecture outweigh the benefits derived from reduced array size [8].

Digitline capacitive components, contributed by each mbit, include junction capacitance, digitline-to-cellplate (poly3), digitline-to-wordline, digitline-to-digitline, digitline-to-substrate, and, in some cases, digitline-to-storage cell (poly2) capacitance. Therefore, each mbit connected to the digitline adds a specific amount of capacitance to the digitline. Most modern DRAM designs have no more than 256 mbits connected to a digitline segment.

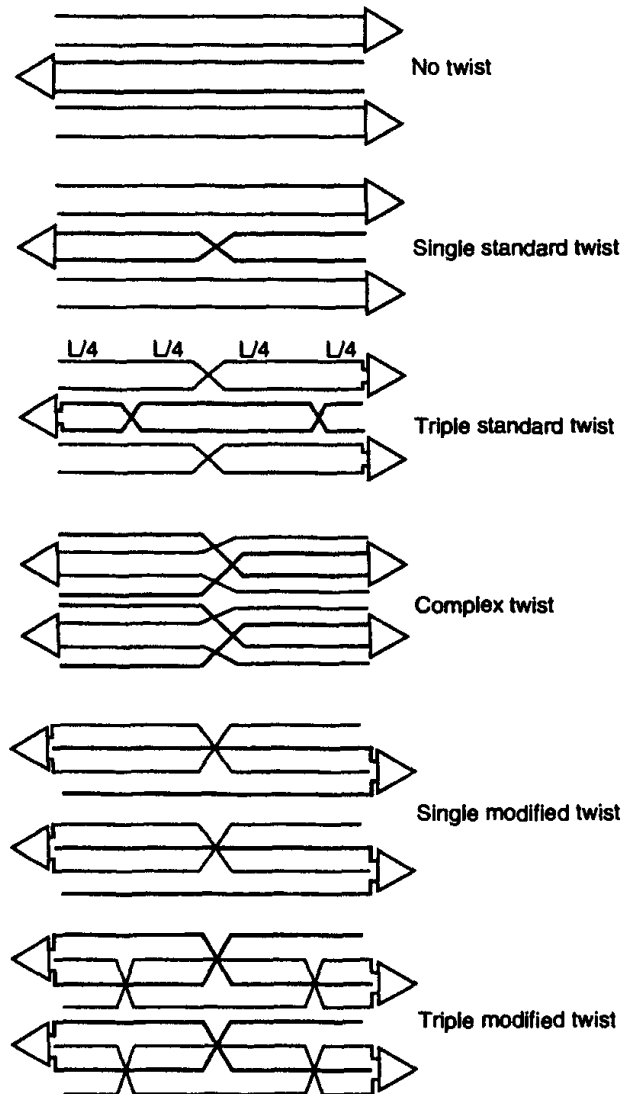


Figure 2.5 Digitline twist schemes.

Two factors dictate this quantity. First, for a given cell size, as determined by row and column pitches, a maximum storage capacitance can be achieved without resorting to exotic processes or excessive cell height. For processes in which the digitline is above the storage capacitor (buried capacitor), contact technology determines the maximum allowable cell height. This fixes the volume available (cell area multiplied by cell height) in which to build the storage capacitor. Second, as the digitline capacitance increases, the power associated with charging and discharging this capacitance during Read and Write operations increases. Any given wordline

essentially accesses (crosses) all of the columns within a DRAM. For a 256-Meg DRAM, each wordline crosses 16,384 columns. With a multiplier such as that, it is easy to appreciate why limits to digitline capacitance are necessary to keep power dissipation low.

Figure 2.8 presents a process cross section for the buried capacitor mbit depicted in Figure 2.2, and Figure 2.9 shows a SEM image of the buried capacitor mbit. This type of mbit, employing a buried capacitor structure, places the digitline physically above the storage capacitor [10]. The digitline is constructed from either metal or polycide, while the digitline contact is formed using a metal or polysilicon plug technology. The mbit capacitor is formed with polysilicon (poly2) as the bottom plate, an oxide-nitride-oxide (ONO) dielectric, and a sheet of polysilicon (poly3). This top sheet of polysilicon becomes a common node shared by all mbit capacitors. The capacitor shape can be simple, such as a rectangle, or complex, such as concentric cylinders or stacked discs. The most complex capacitor structures are the topic of many DRAM process papers [11][12][13].

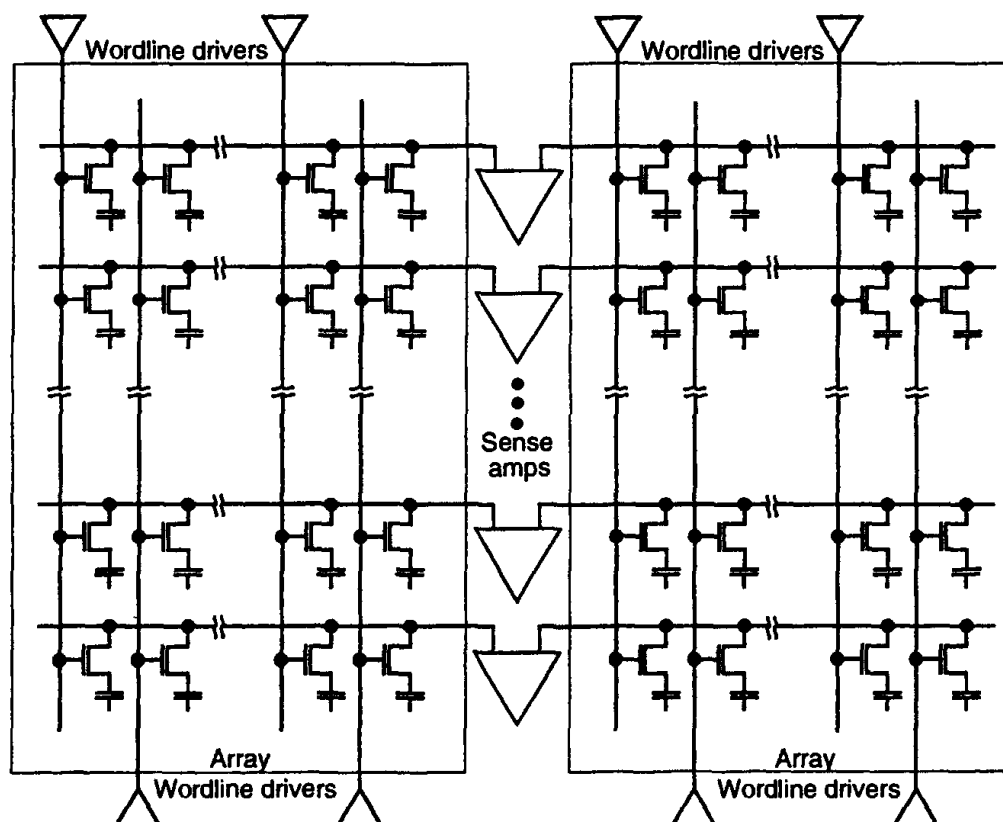


Figure 2.6 Open digitline array schematic.

The ONO dielectric undergoes optimization to achieve maximum capacitance with minimum leakage. It must also tolerate the maximum DRAM operating voltage without breaking down. For this reason, the cell-plate (poly3) is normally biased at $+V_{CC}/2$ V, ensuring that the dielectric has no more than $V_{CC}/2$ V across it for either stored logic state, a logic one at $+V_{CC}/2$ V or a logic zero at $-V_{CC}/2$ V.

Two other basic mbit configurations are used in the DRAM industry. The first, shown in Figures 2.10, 2.11, and 2.12, is referred to as a *buried digitline* or *capacitor over bitline* (COB) cell [14][15]. The digitline in this cell is almost always formed of polysilicon rather than of metal.

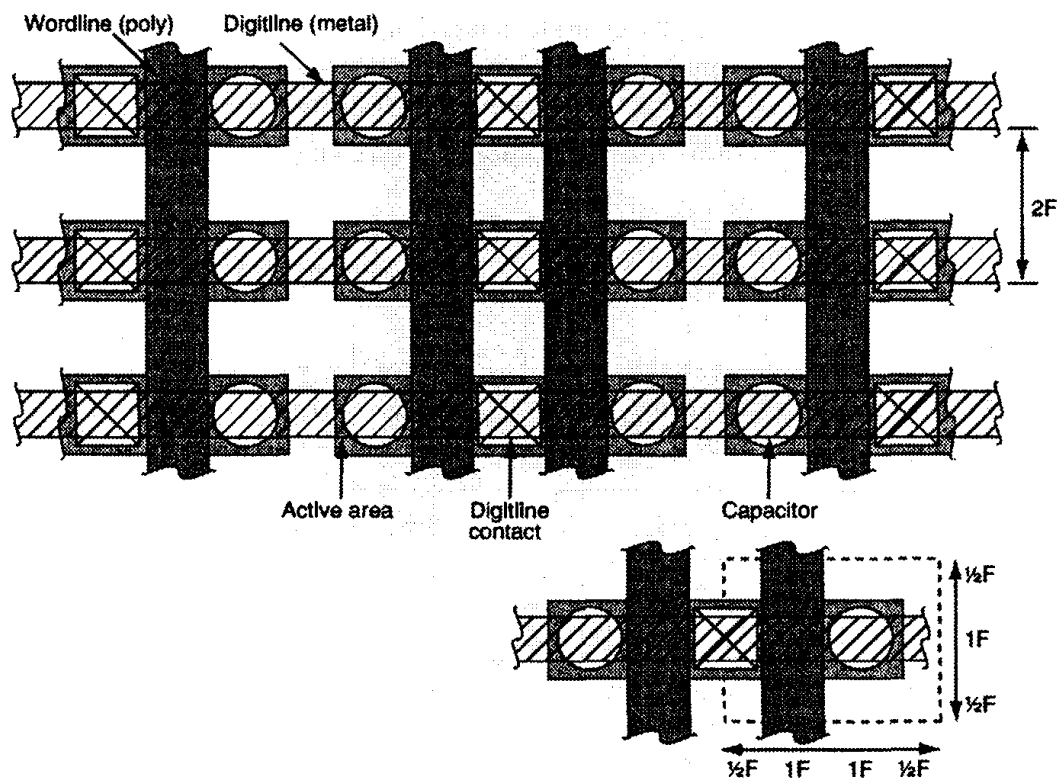


Figure 2.7 Open digitline array layout.
(Feature size (F) is equal to one-half digitline pitch.)

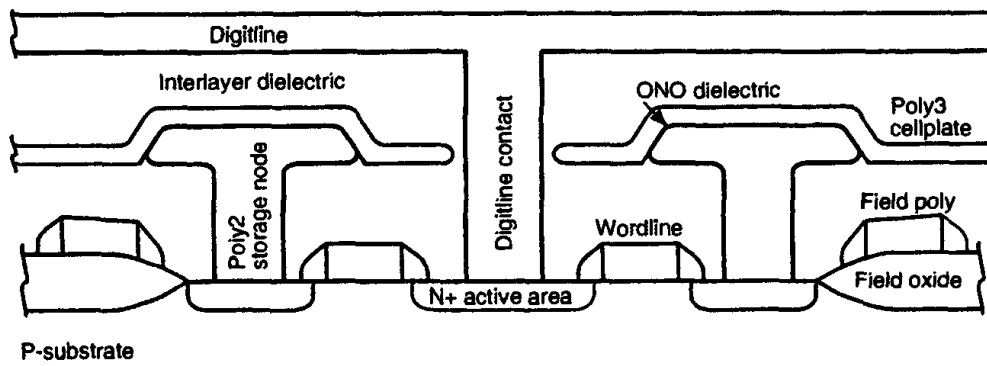


Figure 2.8 Buried capacitor cell process cross section.

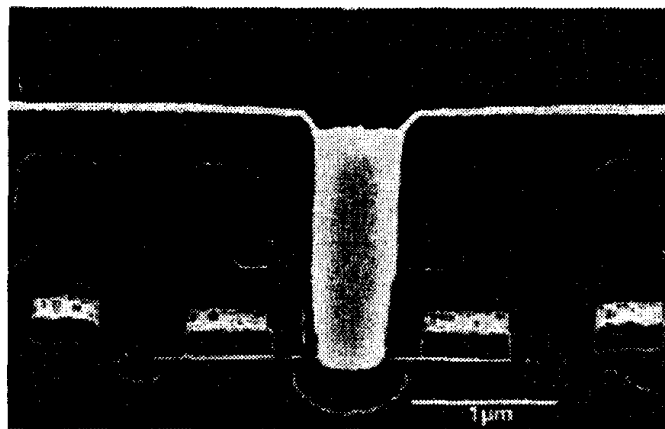


Figure 2.9 Buried capacitor cell process SEM image.

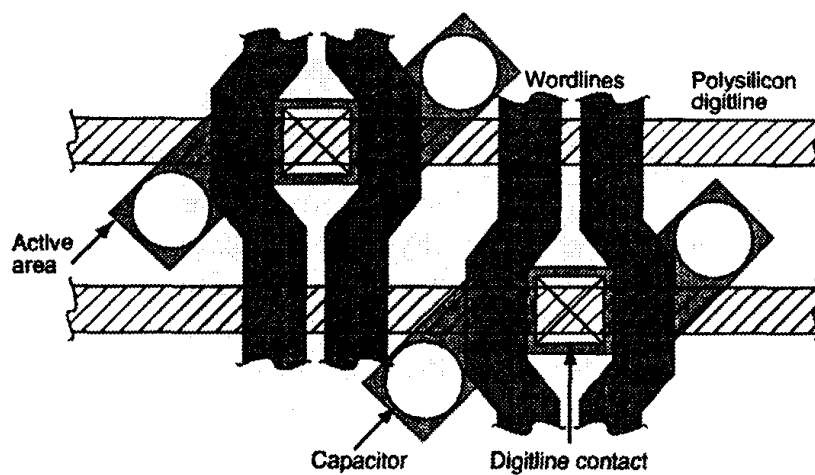


Figure 2.10 Buried digitline mbit cell layout.

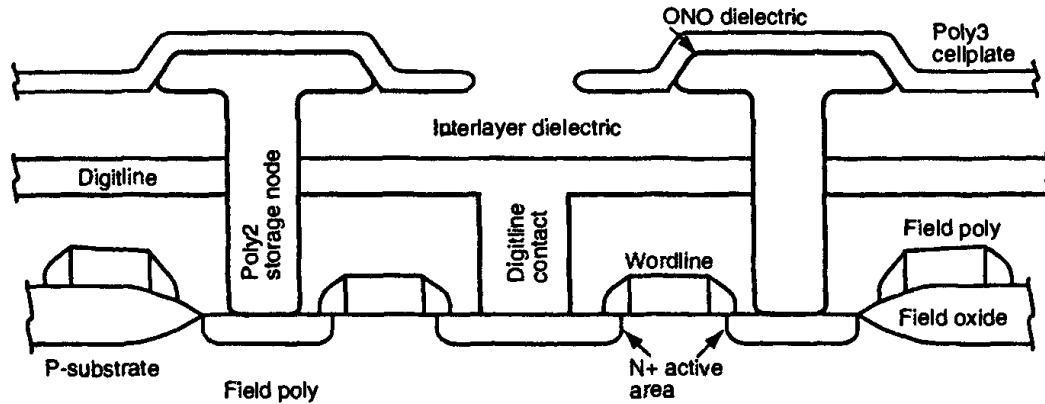


Figure 2.11 Buried digitline mbit process cross section.

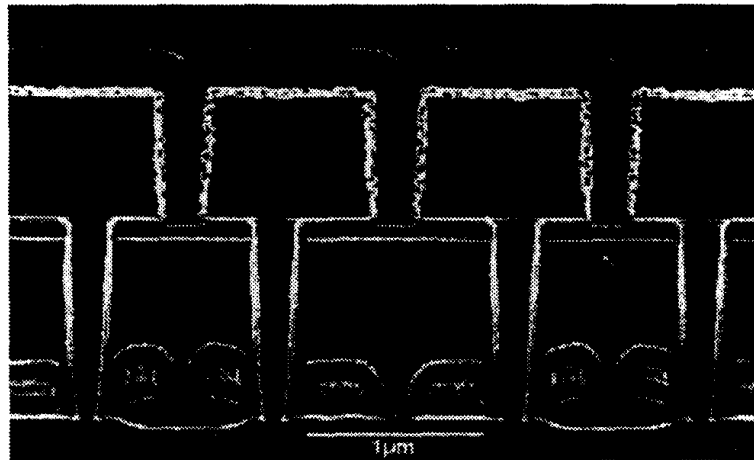


Figure 2.12 Buried digitline mbit process SEM image.

As viewed from the top, the active area is normally bent or angled to accommodate the storage capacitor contact that must drop between digitlines. An advantage of the buried digitline cell over the buried capacitor cell of Figure 2.8 is that its digitline is physically very close to the silicon surface, making digitline contacts much easier to produce. The angled active area, however, reduces the effective active area pitch, constraining the isolation process even further. In buried digitline cells, it is also very difficult to form the capacitor contact. Because the digitline is at or near minimum pitch for the process, insertion of a contact between digitlines can be difficult.

Figures 2.13 and 2.14 present a process cross section of the third type of mbit used in the construction of DRAMs. Using trench storage capacitors, this cell is accordingly called a *trench cell* [12][13]. Trench capacitors are formed in the silicon substrate, rather than above the substrate, after etching deep holes into the wafer. The storage node is a doped polysilicon plug, which is deposited in the hole following growth or deposition of the capacitor dielectric. Contact between the storage node plug and the transistor drain is usually made through a poly strap.

With most trench capacitor designs, the substrate serves as the common-node connection to the capacitors, preventing the use of $+V_{CC}/2$ bias and thinner dielectrics. The substrate is heavily doped around the capacitor to reduce resistance and improve the capacitor's CV characteristics. A real advantage of the trench cell is that the capacitance can be increased by merely etching a deeper hole into the substrate [16].

Furthermore, the capacitor does not add stack height to the design, greatly simplifying contact technology. The disadvantage of trench capacitor technology is the difficulty associated with reliably building capacitors in deep silicon holes and connecting the trench capacitor to the transistor drain terminal.

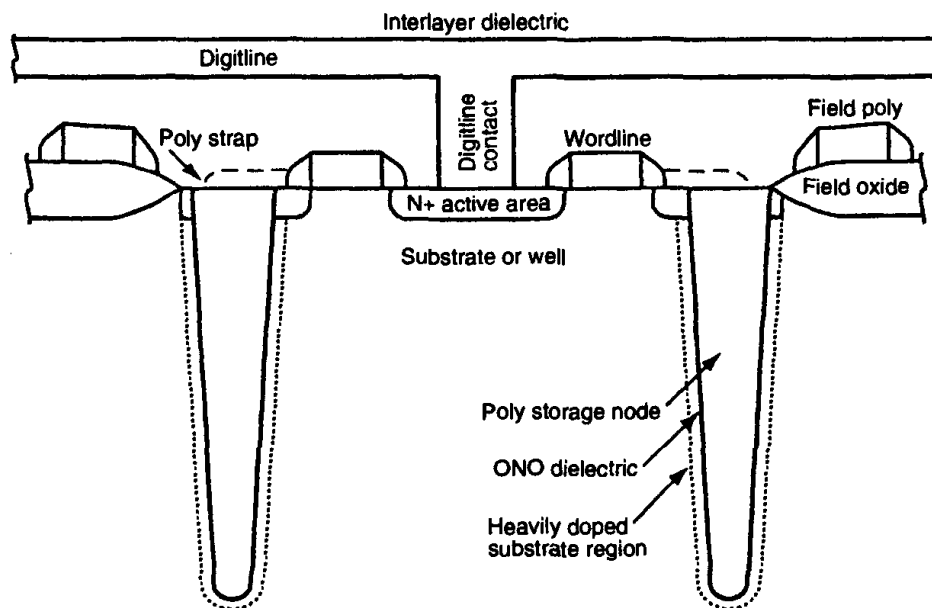


Figure 2.13 Trench capacitor mbit process cross section.



Figure 2.14 Trench capacitor mbit process SEM image.

2.2 THE SENSE AMP

The term *sense amplifier* actually refers to a collection of circuit elements that pitch up to the digitlines of a DRAM array. This collection most generally includes isolation transistors, devices for digitline equilibration and bias, one or more Nsense-amplifiers, one or more Psense-amplifiers, and devices connecting selected digitlines to *I/O* signal lines. All of the circuits along with the wordline driver circuits, as discussed in Section 2.3, are called *pitch cells*. This designation comes from the requirement that the physical layout for these circuits is constrained by the digitline and wordline pitches of an array of mbits. For example, the sense amplifiers for a specific digitline pair (column) are generally laid out within the space of four digitlines. With one sense amplifier for every four digitlines, this is commonly referred to as quarter pitch or four pitch.

2.2.1 Equilibration and Bias Circuits

The first elements analyzed are the equilibration and bias circuits. As we can recall from the discussions of DRAM operation in Section 1.1, the digitlines start at $V_{CC}/2$ V prior to cell Access and Sensing [17]. It is vitally

important to the Sensing operation that both digitlines, which form a column pair, are of the same voltage before the wordline is fired. Any offset voltage appearing between the pair directly reduces the effective signal produced during the Access operation [5]. Equilibration of the digitlines is accomplished with one or more NMOS transistors connected between the digitline conductors. NMOS is used because of its higher drive capability and the resulting faster equilibration.

An equilibration transistor, together with bias transistors, is shown in Figure 2.15. The gate terminal is connected to a signal called *equilibrate* (*EQ*). *EQ* is held to V_{CC} whenever the external row address strobe signal $\overline{RAS}$ is HIGH. This indicates an inactive or *PRECHARGE* state for the DRAM. After $\overline{RAS}$ has fallen, *EQ* transitions LOW, turning the equilibration transistor OFF just prior to any wordline firing. *EQ* will again transition HIGH at the end of a $\overline{RAS}$ cycle to force equilibration of the digitlines. The equilibration transistor is sized large enough to ensure rapid equilibration of the digitlines to prepare the part for a subsequent access.

As shown in Figure 2.15, two more NMOS transistors accompany the *EQ* transistor to provide a bias level of $V_{CC}/2$ V. These devices operate in conjunction with equilibration to ensure that the digitline pair remains at the prescribed voltage for Sensing. Normally, digitlines that are at V_{CC} and ground equilibrate to $V_{CC}/2$ V [5]. The bias devices ensure that this occurs and also that the digitlines remain at $V_{CC}/2$, despite leakage paths that would otherwise discharge them. Again, for the same reasons as for the equilibration transistor, NMOS transistors are used. Most often, the bias and equilibration transistors are integrated to reduce their overall size. $V_{CC}/2$ *PRECHARGE* is used on most modern DRAMs because it reduces power consumption and Read-Write times and improves Sensing operations. Power consumption is reduced because a $V_{CC}/2$ *PRECHARGE* voltage can be obtained by equilibrating the digitlines (which are at V_{CC} and ground, respectively) at the end of each cycle.

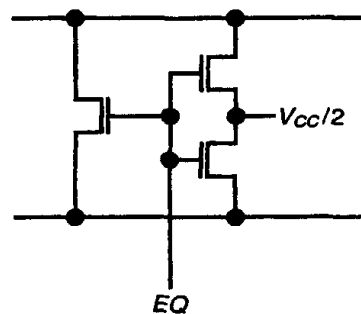


Figure 2.15 Equilibration schematic.

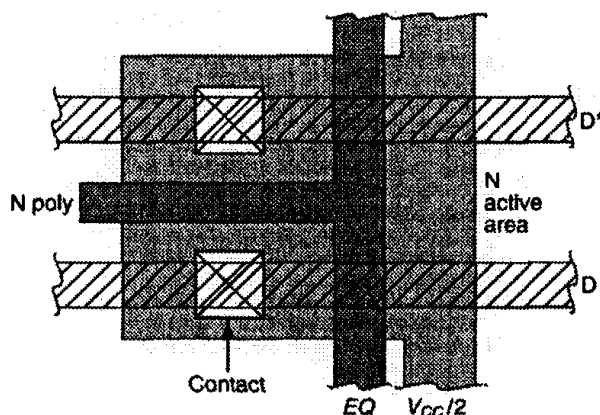


Figure 2.16 Equilibration and bias circuit layout.

The charge-sharing between the digitlines produces $V_{CC}/2$ without additional I_{CC} current. The IBM® PMOS 16-Meg mbit DRAM designs are exceptions: they equilibrate and bias the digitlines to V_{CC} [18]. Because the wordlines and digitlines are both at V_{CC} when the part is inactive, row-to-column shorts that exist do not contribute to increased standby current. $V_{CC}/2$ PRECHARGE DRAMs, on the other hand, suffer higher standby current with row-to-column shorts because the wordlines and digitlines are at different potentials when the part is inactive. A layout for the equilibration and bias circuits is shown in Figure 2.16.

2.2.2 Isolation Devices

Isolation devices are important to the sense amplifier circuits. These devices are NMOS transistors placed between the array digitlines and the sense amplifiers (see Figure 2.19). Isolation transistors are physically located on both ends of the sense amplifier layout. In quarter-pitch sense amplifier designs, there is one isolation transistor for every two digitlines. Although this is twice the active area width and space of an array, it nevertheless sets the limit for isolation processing in the pitch cells.

The isolation devices provide two functions. First, if the sense amps are positioned between and connected to two arrays, they electrically isolate one of the two arrays. This is necessary whenever a wordline fires in one

array because isolation of the second array reduces the digitline capacitance driven by the sense amplifiers, thus speeding Read-Write times, reducing power consumption, and extending Refresh for the isolated array. Second, the isolation devices provide resistance between the sense amplifier and the digitlines. This resistance stabilizes the sense amplifiers and speeds up the Sensing operation by somewhat isolating the highly capacitive digitlines from the low-capacitance sense nodes [19]. Capacitance of the sense nodes between isolation transistors is generally less than 15fF, permitting the sense amplifier to latch much faster than if it were solidly connected to the digitlines. The isolation transistors slow Write-Back to the mbits, but this is far less of a problem than initial Sensing.

2.2.3 Input/Output Transistors

The input/output (I/O) transistors allow data to be read from and written to specific digitline pairs. A single I/O transistor is connected to each sense node as shown in Figure 2.17. The outputs of each I/O transistor are connected to I/O signal pairs. Commonly, there are two pairs of I/O signal lines, which permit four I/O transistors to share a single *column select (CSEL)* control signal. DRAM designs employing two or more metal layers run the column select lines across the arrays in either Metal2 or Metal3. Each column select can activate four I/O transistors on each side of an array to connect four digitline pairs (columns) to peripheral data path circuits. The I/O transistors must be sized carefully to ensure that instability is not introduced into the sense amplifiers by the I/O bias voltage or remnant voltages on the I/O lines.

Although designs vary significantly as to the numerical ratio, I/O transistors are designed to be two to eight times smaller than the Nsense-amplifier transistors. This is sometimes referred to as *beta ratio*. A beta ratio between five and eight is considered standard; however, it can only be verified with silicon. Simulations fail to adequately predict sense amplifier instability, although theory would predict better stability with higher beta ratio and better Write times with lower beta ratio. During Write, the sense amplifier remains ON and must be overdriven by the Write driver (see Section 1.2.2).

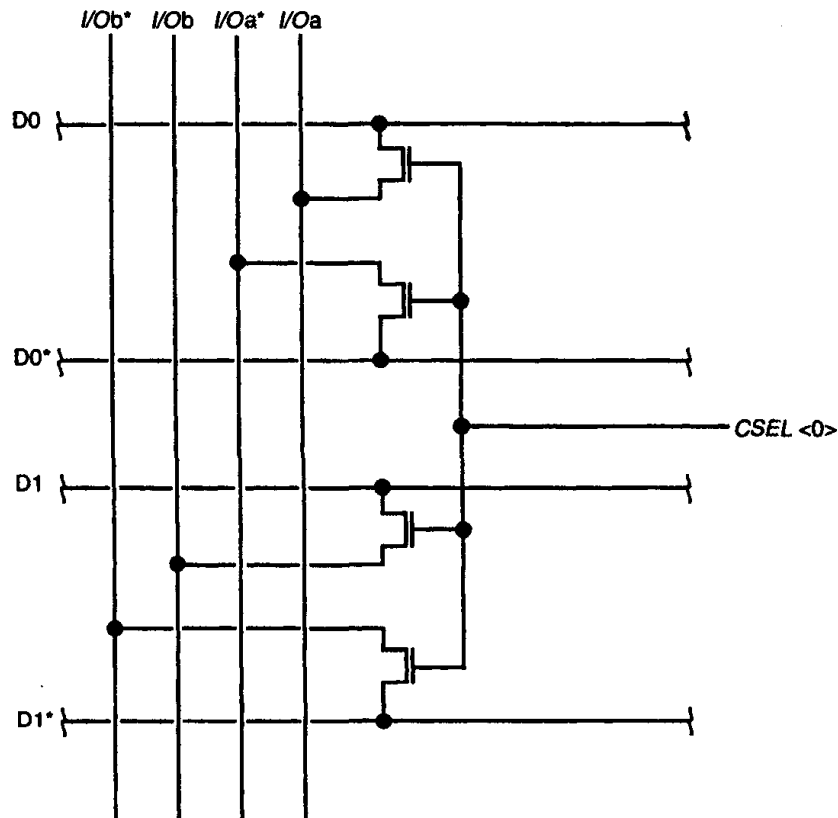


Figure 2.17 I/O transistors.

2.2.4 Nsense- and Psense-Amplifiers

The fundamental elements of any sense amplifier block are the Nsense-amplifier and the Psense-amplifier. These amplifiers, as previously discussed, work together to detect the access signal voltage and drive the digit-lines, accordingly to V_{CC} and ground. The Nsense-amplifier depicted in Figure 2.18 consists of cross-coupled NMOS transistors. The Nsense-amplifier drives the low-potential digitline to ground. Similarly, the Psense-amplifier consists of cross-coupled PMOS transistors and drives the high-potential digitline to V_{CC} .

The sense amplifiers are carefully designed to guarantee correct detection and amplification of the small signal voltage produced during cell access (less than 200mV) [5]. Matching of transistor V_{TH} , transconductance, and junction capacitance within close tolerances helps ensure reliable sense amplifier operation. Ultimately, the layout dictates the overall balance and performance of the sense amplifier block. As a result, a tremendous amount of time is spent ensuring that the sense amplifier layout is optimum. Sym-

metry and exact duplication of elements are critical to a successful design. This includes balanced coupling to all sources of noise, such as *I/O* lines and latch signals (*NLAT** and *ACT*). Balance is especially critical for layout residing inside the isolation transistors. And because the sense node capacitance is very low, it is more sensitive to noise and circuit imbalances.

While the majority of DRAM designs latch the digitlines to V_{CC} and ground, a growing number of designs are beginning to reduce these levels. Various technical papers report improved Refresh times and lower power dissipation through reductions in latch voltages [20][21]. At first, this appears contradictory: writing a smaller charge into the memory cell would require refreshing the cell more often. However, by maintaining lower *drain-to-source voltages* (V_{DS}) and negative *gate-to-source voltages* (V_{GS}) across nonaccessed mbit transistors, substantially lower subthreshold leakage and longer Refresh times can be realized, despite the smaller stored charge. An important concern that we are not mentioning here is the leakage resulting from defects in the silicon crystal structure. These defects can result in an increase in the drain/substrate diode saturation current and can practically limit the Refresh times.

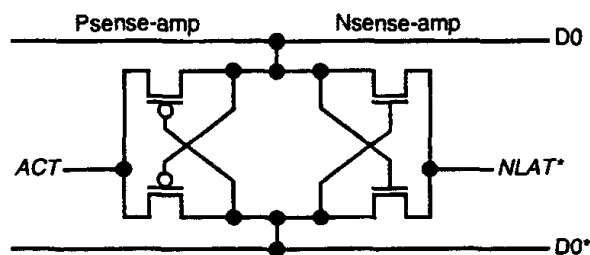


Figure 2.18 Basic sense amplifier block.

Most designs that implement reduced latch voltages generally raise the *Nsense*-amplifier latch voltage without lowering the *Psense*-amplifier latch voltage. Designated as boosted sense ground designs, they write data into each mbit using full V_{CC} for a logic one and boosted ground for a logic zero. The sense ground level is generally a few hundred millivolts above true ground. In standard DRAMs, which drive digitlines fully to ground, the V_{GS} of nonaccessed mbits becomes zero when the digitlines are latched. This results in high subthreshold leakage for a stored one level because full V_{CC} exists across the mbit transistor while the V_{GS} is held to zero. Stored zero levels do not suffer from prolonged subthreshold leakage: any amount of cell leakage produces a negative V_{GS} for the transistor. The net effect is that

a stored one level leaks away much faster than a stored zero level. One's level retention, therefore, establishes the maximum Refresh period for most DRAM designs. Boosted sense ground extends Refresh by reducing sub-threshold leakage for stored ones. This is accomplished by guaranteeing negative gate-to-source bias on nonaccessed mbit transistors. The benefit of extended Refresh from these designs is somewhat diminished, though, by the added complexity of generating boosted ground levels and the problem of digitlines that no longer equilibrate at $V_{CC}/2$ V.

2.2.5 Rate of Activation

The rate at which the sense amplifiers are activated has been the subject of some debate. A variety of designs use multistage circuits to control the rate at which $NLAT^*$ fires. Especially prevalent with boosted sense ground designs are two-stage circuits that initially drive $NLAT^*$ quickly toward true ground to speed sensing and then bring $NLAT^*$ to the boosted ground level to reduce cell leakage. An alternative to this approach, again using two-stage drivers, drives $NLAT^*$ to ground, slowly at first to limit current and digitline disturbances. This is followed by a second phase in which $NLAT^*$ is driven more strongly toward ground to complete the Sensing operation. This phase usually occurs in conjunction with ACT activation. Although these two designs have contrary operation, each meets specific performance objectives: trading off noise and speed.

2.2.6 Configurations

Figure 2.19 shows a sense amplifier block commonly used in double- or triple-metal designs. It features two Psense-amplifiers outside the isolation transistors, a pair of $EQ/bias$ (EQb) devices, a single Nsense-amplifier, and a single I/O transistor for each digitline. Because only half of the sense amplifiers for each array are on one side, this design is quarter pitch, as are the designs in Figures 2.20 and 2.21. Placement of the Psense-amplifiers outside the isolation devices is necessary because a full one level (V_{CC}) cannot pass through unless the gate terminal of the ISO transistors is driven above V_{CC} . $EQ/bias$ transistors are placed outside of the ISO devices to permit continued equilibration of digitlines in arrays that are isolated. The I/O transistor gate terminals are connected to a common $CSEL$ signal for four adjacent digitlines. Each of the four I/O transistors is tied to a separate I/O bus. This sense amplifier, though simple to implement, is somewhat larger than other designs due to the presence of two Psense-amplifiers.

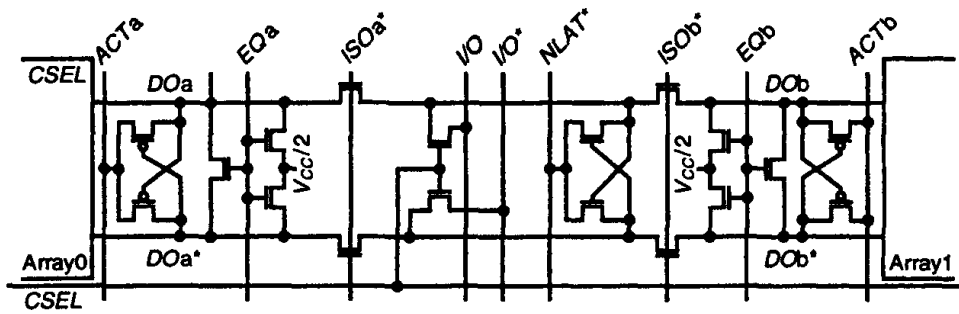


Figure 2.19 Standard sense amplifier block.

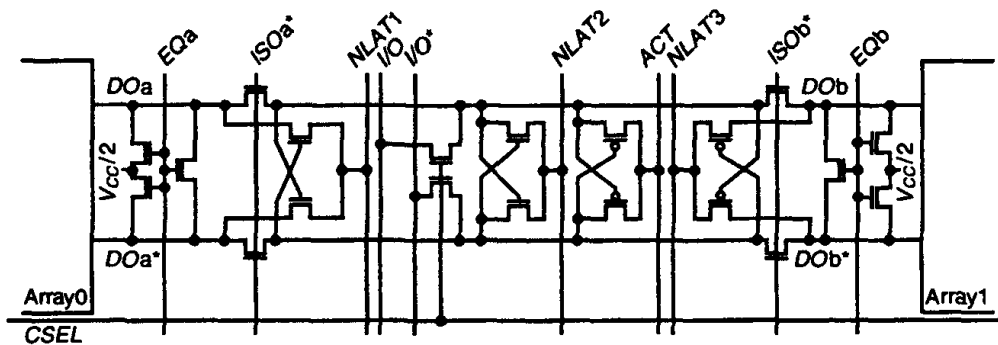


Figure 2.20 Complex sense amplifier block.

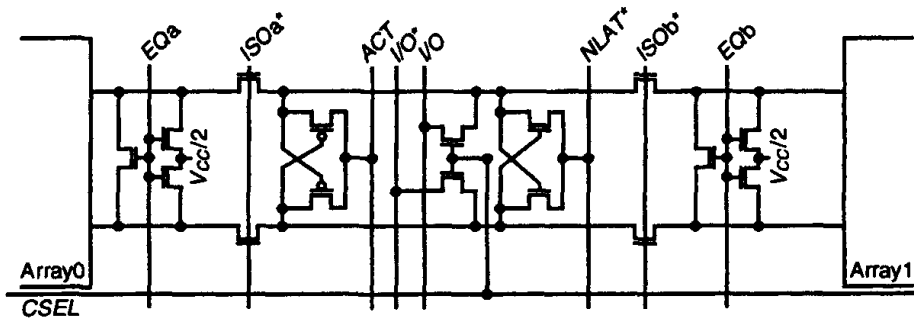


Figure 2.21 Reduced sense amplifier block.

Figure 2.20 shows a second, more complicated style of sense amplifier block. This design employs a single Psense-amplifier and three sets of Nsense-amplifiers. Because the Psense-amplifier is within the isolation transistors, the isolation devices must be the NMOS depletion type, the PMOS enhancement type, or the NMOS enhancement type with boosted

gate drive to permit writing a full logic one into the array mbits. The triple Nsense-amplifier is suggestive of PMOS isolation transistors; it prevents full zero levels to be written unless the Nsense-amplifiers are placed adjacent to the arrays. In this more complicated style of sense amplifier block, using three Nsense-amplifiers guarantees faster sensing and higher stability than a similar design using only two Nsense-amplifiers. The inside Nsense-amplifier is fired before the outside Nsense-amplifiers. However, this design will not yield a minimum layout, an objective that must be traded off against performance needs.

The sense amplifier block of Figure 2.21 can be considered a reduced configuration. This design has only one Nsense-amp and one Psense-amp, both of which are placed within the isolation transistors. To write full logic levels, either the isolation transistors must be depletion mode devices or the gate voltage must be boosted above V_{CC} by at least one V_{TH} . This design still uses a pair of EQ /bias circuits to maintain equilibration on isolated arrays.

Only a handful of designs operates with a single EQ /bias circuit inside the isolation devices, as shown in Figure 2.22. Historically, DRAM engineers tended to shy away from designs that permitted digitlines to float for extended periods of time. However, as of this writing, at least three manufacturers in volume production have designs using this scheme.

A sense amplifier design for single-metal DRAMs is shown in Figure 2.23. Prevalent on 1-Meg and 4-Meg designs, single-metal processes conceded to multi-metal processes at the 16-Meg generation. Unlike the sense amplifiers shown in Figures 2.19, 2.20, 2.21, and 2.22, single-metal sense amps are laid out at half pitch: one amplifier for every two array digitlines. This type of layout is extremely difficult and places tight constraints on process design margins. With the loss of Metal2, the column select signals are not brought across the memory arrays. Generating column select signals locally for each set of I/O transistors requires a full column decode block.

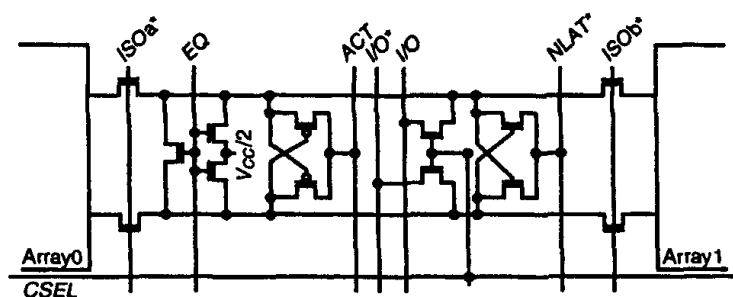


Figure 2.22 Minimum sense amplifier block.

As shown in Figure 2.23, the Nsense-amp and Psense-amps are placed on separate ends of the arrays. Sharing sense amps between arrays is especially beneficial for single-metal designs. As illustrated in Figure 2.23, two Psense-amps share a single Nsense-amp. In this case, with the I/O devices on only one end, the right Psense-amp is activated only when the right array is accessed. Conversely, the left Psense-amp is always activated, regardless of the accessed array, because all Read and Write data must pass through the left Psense-amp to get to the I/O devices.

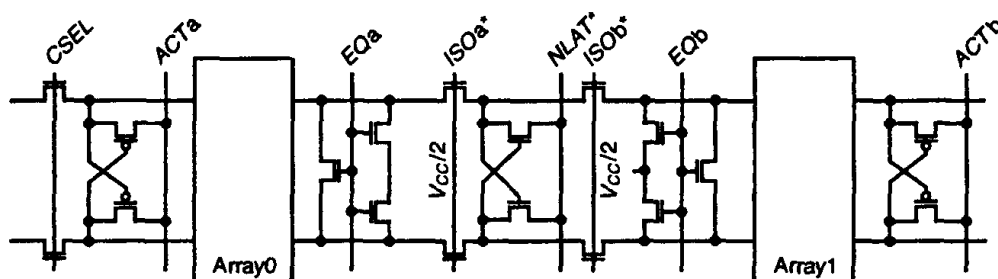


Figure 2.23 Single-metal sense amplifier block.

2.2.7 Operation

A set of signal waveforms is illustrated in Figure 2.24 for the sense amplifier of Figure 2.19. These waveforms depict a Read-Modify-Write cycle (Late Write) in which the cell data is first read out and then new data is written back. In this example, a one level is read out of the cell, as indicated by D^* rising above D during cell access. A one level is always $+V_{CC}/2$ in the mbit cell, regardless of whether it is connected to a true or complement digitline. The correlation between mbit cell data and the data appearing at the DRAM's data terminal (DQ) is a function of the data topology and the presence of data scrambling. Data or topo scrambling is implemented at the circuit level: it ensures that the mbit data state and the DQ logic level are in agreement. An mbit one level ($+V_{CC}/2$) corresponds to a logic one at the DQ , and an mbit zero level ($-V_{CC}/2$) corresponds to a logic zero at the DQ terminal.

Writing specific data patterns into the memory arrays is important to DRAM testing. Each type of data pattern identifies the weaknesses or sensitivities of each cell to the data in surrounding cells. These patterns include solids, row stripes, column stripes, diagonals, checkerboards, and a variety of moving patterns. Test equipment must be programmed with the data topology of each type of DRAM to correctly write each pattern. Often the

tester itself guarantees that the pattern is correctly written into the arrays, unscrambling the complicated data and address topology as necessary to write a specific pattern. On some newer DRAM designs, part of this task is implemented on the DRAM itself, in the form of a topo scrambler, such that the mbit data state matches the DQ logic level. This implementation somewhat simplifies tester programming.

Returning to Figure 2.24, we see that a wordline is fired in Array1. Prior to this, $ISOa^*$ will go LOW to isolate Array0, and EQb will go LOW to disable the EQ /bias transistors connected to Array1. The wordline then fires HIGH, accessing an mbit, which dumps charge onto $D0^*$. $NLAT^*$, which is initially at $V_{CC}/2$, drives LOW to begin the Sensing operation and pulls $D0$ toward ground. Then, ACT fires, moving from ground to V_{CC} , activating the Psense-amplifier and driving $D0^*$ toward V_{CC} . After separation has commenced, $CSEL0$ rises to V_{CC} , turning ON the I/O transistors so that the cell data can be read by peripheral circuits. The I/O lines are biased at a voltage close to V_{CC} , which causes $D1$ to rise while the column is active. After the Read is complete, Write drivers in the periphery turn ON and drive the I/O lines to opposite data states (in our example).

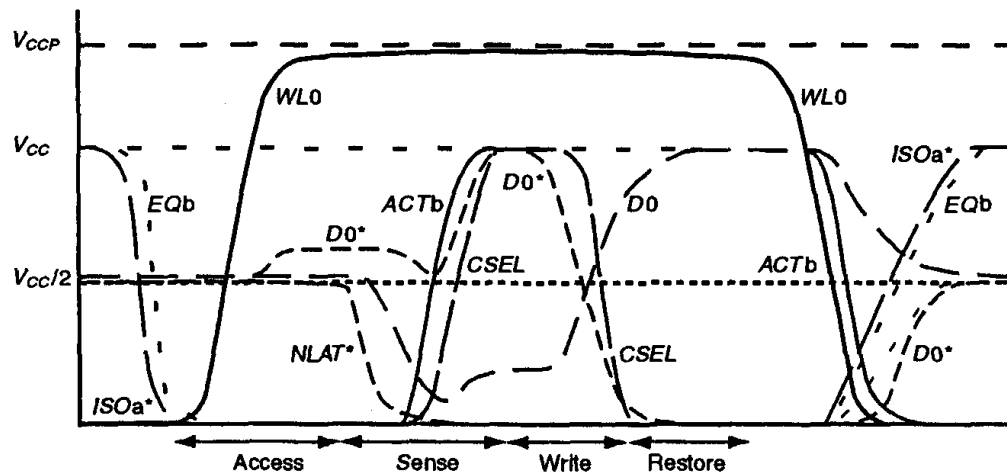


Figure 2.24 Waveforms for the Read-Modify-Write cycle.

This new data propagates through the I/O devices and writes over the existing data stored in the sense amplifiers. Once the sense amplifiers latch the new data, the Write drivers and the I/O devices can be shut down, allowing the sense amplifiers to finish restoring the digitlines to full levels. Following this restoration, the wordline transitions LOW to shut OFF the mbit transistor. Finally, EQb and $ISOa^*$ fire HIGH to equilibrate the digitlines

back to $V_{CC}/2$ and reconnect Array0 to the sense amplifiers. The timing for each event of Figure 2.24 depends on circuit design, transistor sizes, layout, device performance, parasitics, and temperature. While timing for each event must be minimized to achieve optimum DRAM performance, it cannot be pushed so far as to eliminate all timing margins. Margins are necessary to ensure proper device operation over the expected range of process variations and the wide range of operating conditions.

Again, there is not one set of timing waveforms that covers all design options. The sense amps of Figures 2.19–2.23 all require slightly different signals and timings. Various designs actually fire the Psense-amplifier prior to or coincident with the Nsense-amplifier. This obviously places greater constraints on the Psense-amplifier design and layout, but these constraints are balanced by potential performance benefits. Similarly, the sequence of events as well as the voltages for each signal can vary. There are almost as many designs for sense amplifier blocks as there are DRAM design engineers. Each design reflects various influences, preconceptions, technologies, and levels of understanding. The bottom line is to maximize yield and performance and minimize everything else.

2.3 ROW DECODER ELEMENTS

Row decode circuits are similar to sense amplifier circuits in that they pitch up to mbit arrays and have a variety of implementations. A row decode block is comprised of two basic elements: a wordline driver and an address decoder tree. There are three basic configurations for wordline driver circuits: the NOR driver, the inverter (CMOS) driver, and the bootstrap driver. In addition, the drivers and associated decode trees can be configured either as local row decodes for each array section or as global row decodes that drive a multitude of array sections.

Global row decodes connect to multiple arrays through metal wordline straps. The straps are stitched to the polysilicon wordlines at specific intervals dictated by the polysilicon resistance and the desired RC wordline time constant. Most processes that strap wordlines with metal do not silicide the polysilicon, although doing so would reduce the number of stitch regions required. Strapping wordlines and using global row decoders obviously reduces die size [22], very dramatically in some cases. The disadvantage of strapping is that it requires an additional metal layer at minimum array pitch. This puts a tremendous burden on process technologists in which three conductors are at minimum pitch: wordlines, digitlines, and wordline straps.

Local row decoders, on the other hand, require additional die size rather than metal straps. It is highly advantageous to reduce the polysilicon resistance in order to stretch the wordline length and lower the number of row decodes needed. This is commonly achieved with silicided polysilicon processes. On large DRAMs, such as the 1Gb, the area penalty can be prohibitive, making low-resistance wordlines all the more necessary.

2.3.1 Bootstrap Wordline Driver

The bootstrap wordline driver shown in Figure 2.25 is built exclusively from NMOS transistors [22], resulting in the smallest layout of the three types of driver circuits. The absence of PMOS transistors eliminates large Nwell regions from the layout. As the name denotes, this driver relies on bootstrapping principles to bias the output transistor's gate terminal. This bias voltage must be high enough to allow the NMOS transistor to drive the wordline to the boosted wordline voltage V_{CCP} .

Operation of the bootstrap driver is described with help from Figure 2.26. Initially, the driver is OFF with the wordline and *PHASE* terminals at ground. The wordline is held at ground by transistor M2 because the decoder output signal *DEC** is at V_{CC} . The gate of M3, the pass transistor, is fixed at V_{CC} . The signals *DEC* and *DEC** are fed from a decode circuit that will be discussed later. As a complement pair, *DEC* and *DEC** represent the first of two terms necessary to decode the correct wordline. *PHASE0*, which is also fed from a decode circuit, represents the second term. If *DEC* rises to V_{CC} and *DEC** drops to ground, as determined by the decoder, the boot node labeled B1 will rise to $V_{CC} - V_{TN}$ V, and M2 will turn OFF. The wordline continues to be held to ground by M1 because *PHASE0* is still grounded. After B1 rises to $V_{CC} - V_{TN}$, the *PHASE0* signal fires to the boosted wordline voltage V_{CCP} . Because of the gate-to-drain and gate-to-source capacitance of M1, the gate of M1 boots to an elevated voltage, V_{boot} . This voltage is determined by the parasitic capacitance of node B1, C_{GS1} , C_{GD1} , V_{CCP} , and the initial voltage at B1, $V_{CC} - V_{TN}$. Accordingly,

$$V_{boot} \cong \frac{(V_{CCP} \cdot C_{GD1})}{(C_{GS1} + C_{GD1} + C_{B1})} + (V_{CC} - V_{TN}).$$

In conjunction with the wordline voltage rising from ground to V_{CCP} , the gate-to-source capacitance of M1 provides a secondary boost to the boot node. The secondary boost helps to ensure that the boot voltage is adequate to drive the wordline to a full V_{CCP} level.

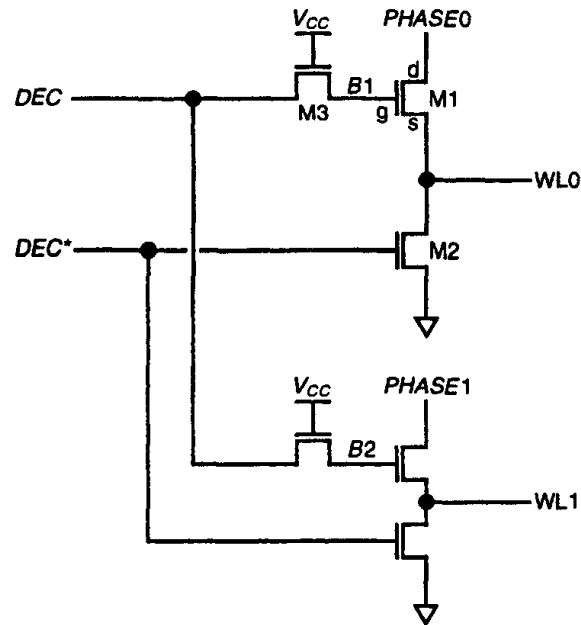


Figure 2.25 Bootstrap wordline driver.

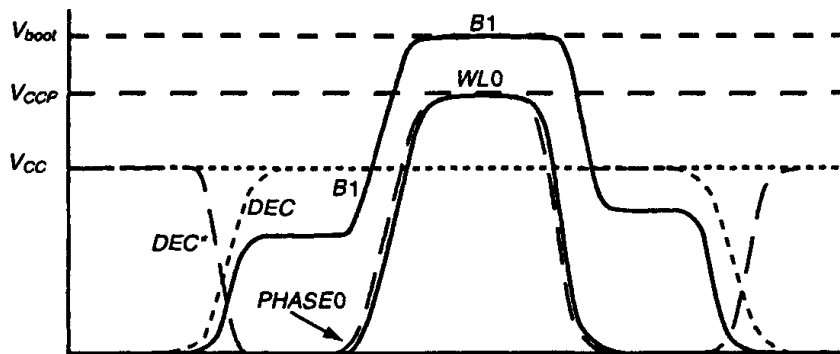


Figure 2.26 Bootstrap operation waveforms.

The layout of the boot node is very important to the bootstrap wordline driver. First, the parasitic capacitance of node *B1*, which includes routing, junction, and overlap components, must be minimized to achieve maximum boot efficiency. Second, charge leakage from the boot node must be minimized to ensure adequate V_{GS} for transistor *M1* such that the wordline remains at V_{CCP} for the maximum $\overline{RAS}$ low period. Low leakage is often achieved by minimizing the source area for *M3* or using donut gate structures that surround the source area, as illustrated in Figure 2.27.

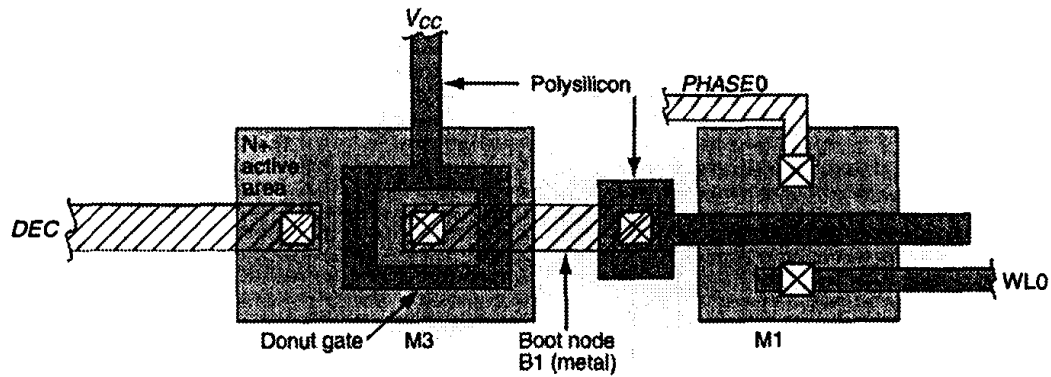


Figure 2.27 Donut gate structure layout.

The bootstrap driver is turned OFF by first driving the *PHASE0* signal to ground. M1 remains ON because node *B1* cannot drop below $V_{CC} - V_{TH}$; M1 substantially discharges the wordline toward ground. This is followed by the address decoder turning OFF, bringing *DEC* to ground and *DEC** to V_{CC} . With *DEC** at V_{CC} , transistor M2 turns ON and fully clamps the wordline to ground. A voltage level translator is required for the *PHASE* signal because it operates between ground and the boosted voltage V_{CCP} . For a global row decode configuration, this requirement is not much of a burden. For a local row decode configuration, however, the requirement for level translators can be very troublesome. Generally, these translators are placed either in the array gap cells at the intersection of the sense amplifier blocks and row decode blocks or distributed throughout the row decode block itself. The translators require both PMOS and NMOS transistors and must be capable of driving large capacitive loads. Layout of the translators is exceedingly difficult, especially because the overall layout needs to be as small as possible.

2.3.2 NOR Driver

The second type of wordline driver is similar to the bootstrap driver in that two decode terms drive the output transistor from separate terminals. The NOR driver, as shown in Figure 2.28, uses a PMOS transistor for M1 and does not rely on bootstrapping to derive the gate bias. Rather, the gate is driven by a voltage translator that converts *DEC* from V_{CC} to V_{CCP} voltage levels. This conversion is necessary to ensure that M1 remains OFF for unselected wordlines as the *PHASE* signal, which is shared by multiple drivers, is driven to V_{CCP} .

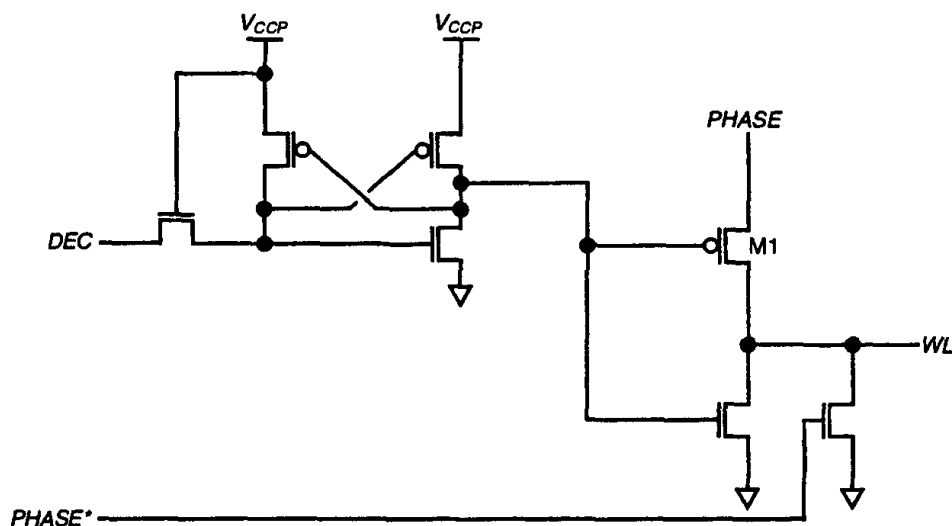


Figure 2.28 NOR driver.

To fire a specific wordline, *DEC* must be HIGH and the appropriate *PHASE* must fire HIGH. Generally, there are four to eight *PHASE* signals per row decoder block. The NOR driver requires one level translator for each *PHASE* and *DEC* signal. By comparison, the bootstrap driver only requires level translators for the *PHASE* signal.

2.3.3 CMOS Driver

The final wordline driver configuration is shown in Figure 2.29. In general, it lacks a specific name: it is sometimes referred to as a CMOS inverter driver or a CMOS driver. Unlike the first two drivers, the output transistor M1 has its source terminal permanently connected to V_{CCP} . This driver, therefore, features a voltage translator for each and every wordline. Both decode terms *DEC* and *PHASE** combine to drive the output stage through the translator. The advantage of this driver, other than simple operation, is low power consumption. Power is conserved because the translators drive only the small capacitance associated with a single driver. The *PHASE* translators of both the bootstrap and NOR drivers must charge considerable junction capacitance. The disadvantages of the CMOS driver are layout complexity and standby leakage current. Standby leakage current is a product of V_{CCP} voltage applied to M1 and its junction and subthreshold leakage currents. For a large DRAM with high numbers of wordline drivers, this leakage current can easily exceed the entire standby current budget unless great care is exercised in designing output transistor M1.

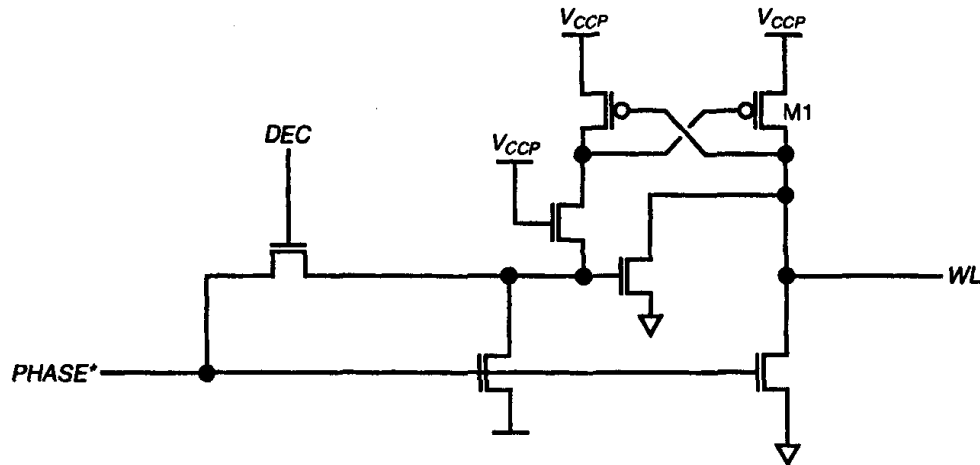


Figure 2.29 CMOS driver.

2.3.4 Address Decode Trees

With the wordline driver circuits behind us, we can turn our attention to the address decoder tree. There is no big secret to address decoding in the row decoder network. Just about any type of logic suffices: static, dynamic, pass gate, or a combination thereof. With any type of logic, however, the primary objectives in decoder design are to maximize speed and minimize die area. Because a great variety of methods have been used to implement row address decoder trees, it is next to impossible to cover them all. Instead, we will give an insight into the possibilities by discussing a few of them.

Regardless of the type of logic with which a row decoder is implemented, the layout must completely reside beneath the row address signal lines to constitute an efficient, minimized design. In other words, the metal address tracks dictate the die area available for the decoder. Any additional tracks necessary to complete the design constitute wasted silicon. For DRAM designs requiring global row decode schemes, the penalty for inefficient design may be insignificant; however, for distributed local row decode schemes, the die area penalty may be significant. As with mbits and sense amplifiers, time spent optimizing row decode circuits is time well spent.

2.3.5 Static Tree

The most obvious form of address decode tree uses static CMOS logic. As shown in Figure 2.30, a simple tree can be designed using two-input NAND gates. While easy to design schematically, static logic address trees are not popular. They waste silicon and are very difficult to lay out efficiently. Static logic requires two transistors for each address term, one

NMOS and one PMOS, which can be significant for many address terms. Furthermore, static gates must be cascaded to accumulate address terms, adding gate delays at each level. For these and other reasons, static logic gates are not used in row decode address trees in today's state-of-the-art DRAMs.

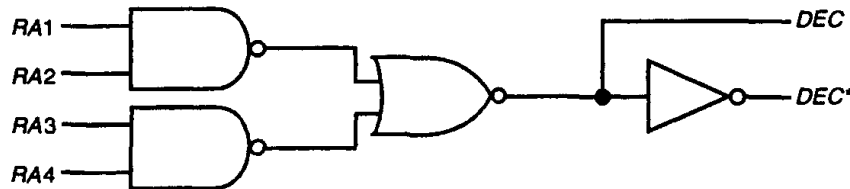


Figure 2.30 Static decode tree.

2.3.6 P&E Tree

The second type of address tree uses dynamic logic, the most prevalent being *precharge and evaluate* (P&E) logic. Used by the majority of DRAM manufacturers, P&E address trees come in a variety of configurations, although the differences between them can be subtle. Figure 2.31 shows a simplified schematic for one version of a P&E address tree designed for use with bootstrapped wordline drivers. P&E address tree circuits feature one or more PMOS *PRECHARGE* transistors and a cascade of NMOS *ENABLE* transistors M2–M4. This P&E design uses half of the transistors required by the static address tree of Figure 2.30. As a result, the layout of the P&E tree is much smaller than that of the static tree and fits more easily under the address lines. The *PRE* transistor is usually driven by a *PRECHARGE** signal under the control of the $\overline{RAS}$ chain logic. *PRECHARGE** and transistor M1 ensure that *DEC** is precharged HIGH, disabling the wordline driver and preparing the tree for row address activation.

M7 is a weak PMOS transistor driven by the *DEC* inverter (M5 and M6). Together, M7 and the inverter form a latch to ensure that *DEC** remains HIGH for all decoders that are not selected by the row addresses. At the beginning of any $\overline{RAS}$ cycle, *PRECHARGE** is LOW and the row addresses are all disabled (LOW). After $\overline{RAS}$ falls, *PRECHARGE** transitions HIGH to turn OFF M1; then the row addresses are enabled. If RA1–RA3 all go HIGH, then M2–M4 turn ON, overpowering M7 and driving *DEC** to ground and subsequently *DEC* to V_{CC} . The output of this tree segment normally drives four bootstrapped wordline drivers, each connected to a separate *PHASE* signal. Therefore, for an array with 256 wordlines, there will be 64 such decode trees.

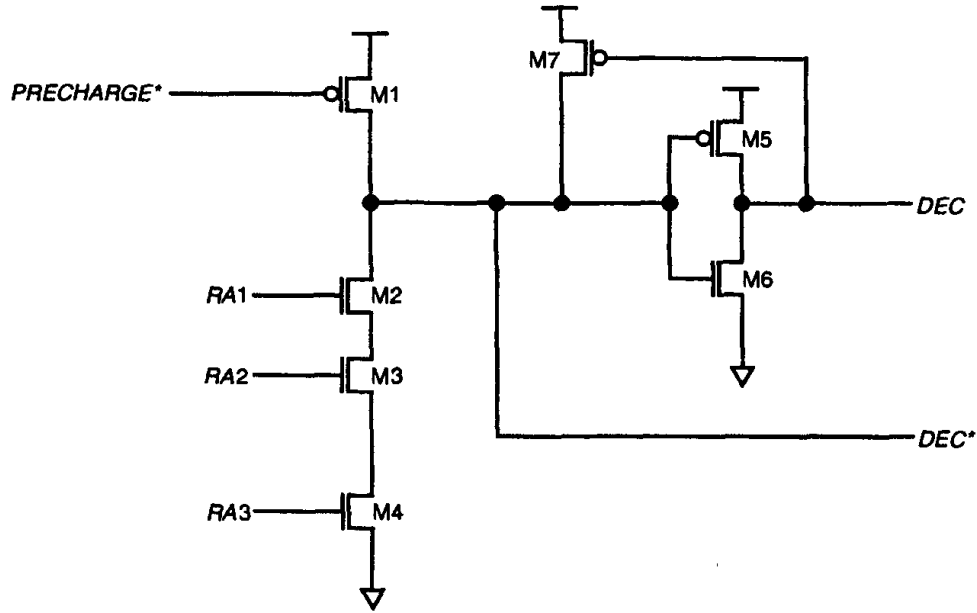


Figure 2.31 P&E decode tree.

2.3.7 Predecoding

The row address lines shown as $RA1$ – $RA3$ can be either true and complement or predecoded. Predecoded address lines are formed by logically combining (AND) addresses as shown in Table 2.1.

Table 2.1 Predecoded address truth table.

RA 0	RA 1	PR01<n>	PR01<0>	PR01<1>	PR01<2>	PR01<3>
0	0	0	1	0	0	0
1	0	1	0	1	0	0
0	1	2	0	0	1	0
1	1	3	0	0	0	1

The advantages of using predecoded addresses include lower power (fewer signals make transitions during address changes) and higher efficiency (only three transistors are necessary to decode six addresses for the circuit of Figure 2.31). Predecoding is especially beneficial in redundancy circuits. In fact, predecoded addresses are used throughout most DRAM designs today.

2.3.8 Pass Transistor Tree

The final type of address tree to be examined, shown in Figure 2.32, uses pass transistor logic. Pass transistor address trees are similar to P&E trees in numerous ways. Both designs use PMOS *PRECHARGE* transistors and NMOS address *ENABLE* transistors. Unlike P&E logic, however, the NMOS cascade does not terminate at ground. Rather, the cascade of M2–M4 goes to a *PHASE** signal, which is HIGH during *PRECHARGE* and LOW during *EVALUATE*. The address signals operate the same as in the P&E tree: HIGH to select and LOW to deselect. The pass transistor tree is shown integrated into a CMOS wordline driver. This is necessary because the pass transistor tree and the CMOS wordline driver are generally used together and their operation is complementary. The cross-coupled PMOS transistors of the CMOS level translator provide a latch necessary to keep the final interstage node biased at V_{CC} . Again the latch has a weak pull-up, which is easily overpowered by the cascaded NMOS *ENABLE* transistors. A pass transistor address tree is not used with bootstrapped wordline drivers because the *PHASE* signal feeds into the address tree logic rather than into the driver, as required by the bootstrap driver.

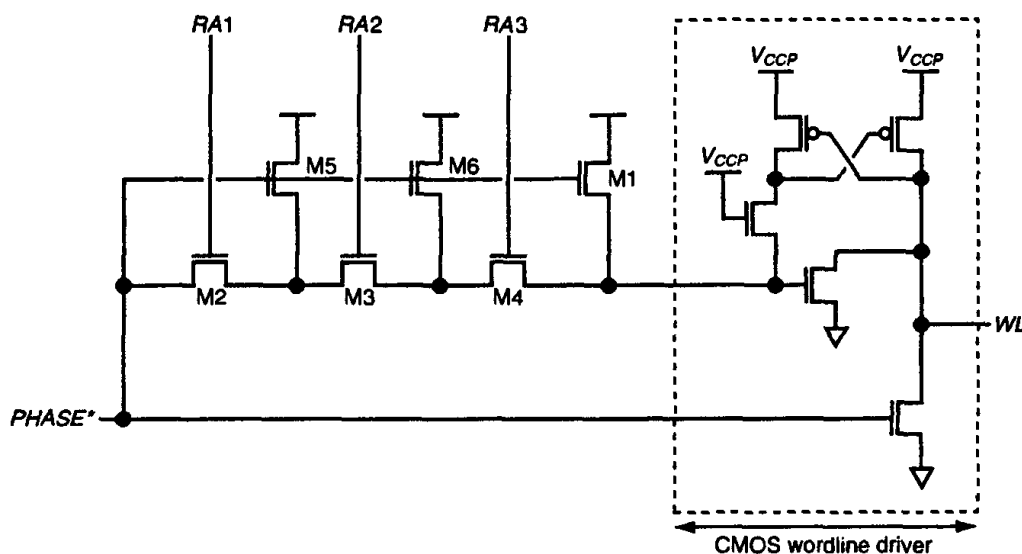


Figure 2.32 Pass transistor decode tree.

2.4 DISCUSSION

We have briefly examined the basic elements required in DRAM row decoder blocks. Numerous variations are possible. No single design is best for all applications. As with sense amplifiers, design depends on technology and performance and cost trade-offs.

REFERENCES

- [1] K. Itoh, "Trends in Megabit DRAM Circuit Design," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 778–791, June 1990.
- [2] D. Takashima, S. Watanabe, H. Nakano, Y. Oowaki, and K. Ohuchi, "Open/Folded Bit-Line Arrangement for Ultra-High-Density DRAM's," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 539–542, April 1994.
- [3] Hideto Hidaka, Yoshio Matsuda, and Kazuyasu Fujishima, "A Divided/Shared Bit-Line Sensing Scheme for ULSI DRAM Cores," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 473–477, April 1991.
- [4] M. Aoki, Y. Nakagome, M. Horiguchi, H. Tanaka, S. Ikenaga, J. Etoh, Y. Kawamoto, S. Kimura, E. Takeda, H. Sunami, and K. Itoh, "A 60-ns 16-Mbit CMOS DRAM with a Transposed Data-Line Structure," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1113–1119, October 1988.
- [5] R. Kraus and K. Hoffmann, "Optimized Sensing Scheme of DRAMs," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 895–899, August 1989.
- [6] T. Yoshihara, H. Hidaka, Y. Matsuda, and K. Fujishima, "A Twisted Bitline Technique for Multi-Mb DRAMs," *1988 IEEE ISSCC Digest of Technical Papers*, pp. 238–239.
- [7] Yukihito Oowaki, Kenji Tsuchida, Yohji Watanabe, Daisaburo Takashima, Masako Ohta, Hiroaki Nakano, Shigeyoshi Watanabe, Akihiro Nitayama, Fumio Horiguchi, Kazunori Ohuchi, and Fujio Masuoka, "A 33-ns 64Mb DRAM," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1498–1505, November 1991.
- [8] M. Inoue, H. Kotani, T. Yamada, H. Yamauchi, A. Fujiwara, J. Matsushima, H. Akamatsu, M. Fukumoto, M. Kubota, I. Nakao, N. Aoi, G. Fuse, S. Ogawa, S. Odanaka, A. Ueno, and Y. Yamamoto, "A 16Mb DRAM with an Open Bit-Line Architecture," *1988 IEEE ISSCC Digest of Technical Papers*, pp. 246–247.
- [9] Y. Kubota, Y. Iwase, K. Iguchi, J. Takagi, T. Watanabe, and K. Sakiyama, "Alternatively-Activated Open Bitline Technique for High Density DRAM's," *IEICE Trans. Electron.*, vol. E75-C, pp. 1259–1266, October 1992.
- [10] T. Hamada, N. Tanabe, H. Watanabe, K. Takeuchi, N. Kasai, H. Hada, K. Shibahara, K. Tokashiki, K. Nakajima, S. Hirasawa, E. Ikawa, T. Saeki, E. Kakehashi, S. Ohya, and T. Kunio, "A Split-Level Diagonal Bit-Line (SLDB) Stacked Capacitor Cell for 256Mb DRAMs," *1992 IEDM Technical Digest*, pp. 799–802.
- [11] Toshinori Morihara, Yoshikazu Ohno, Takahisa Eimori, Toshiharu Katayama, Shinichi Satoh, Tadashi Nishimura, and Hirokazu Miyoshi, "Disk-Shaped Stacked Capacitor Cell for 256Mb Dynamic Random-Access Memory,"

- Japan Journal of Applied Physics*, vol. 33, Part 1, pp. 4570–4575, August 1994.
- [12] J. H. Ahn, Y. W. Park, J. H. Shin, S. T. Kim, S. P. Shim, S. W. Nam, W. M. Park, H. B. Shin, C. S. Choi, K. T. Kim, D. Chin, O. H. Kwon, and C. G. Hwang, "Micro Villus Patterning (MVP) Technology for 256Mb DRAM Stack Cell," 1992 Symposium on VLSI Tech. Digest of Technical Papers, pp. 12–13.
- [13] Kazuhiko Sagara, Tokuo Kure, Shoji Shukuri, Jiro Yugami, Norio Hasegawa, Hidekazu Goto, and Hisaomi Yamashita, "Recessed Memory Array Technology for a Double Cylindrical Stacked Capacitor Cell of 256M DRAM," *IEICE Trans. Electron.*, vol. E75-C, pp. 1313–1322, November 1992.
- [14] S. Ohya, "Semiconductor Memory Device Having Stacked-Type Capacitor of Large Capacitance," United States Patent Number 5,298,775, March 29, 1994.
- [15] M. Sakao, N. Kasai, T. Ishijima, E. Ikawa, H. Watanabe, K. Terada, and T. Kikkawa, "A Capacitor-Over-Bit-Line (COB) Cell with Hemispherical-Grain Storage Node for 64Mb DRAMs," 1990 *IEDM Technical Digest*, pp. 655–658.
- [16] G. Bronner, H. Aochi, M. Gall, J. Gambino, S. Gernhardt, E. Hammerl, H. Ho, J. Iba, H. Ishiuchi, M. Jaso, R. Kleinhenz, T. Mii, M. Narita, L. Nesbit, W. Neumueller, A. Nitayama, T. Ohiwa, S. Parke, J. Ryan, T. Sato, H. Takato, and S. Yoshikawa, "A Fully Planarized 0.25 μ m CMOS Technology for 256Mbit DRAM and Beyond," 1995 Symposium on VLSI Tech. Digest of Technical Papers, pp. 15–16.
- [17] N. C.-C. Lu and H. H. Chao, "Half- V_{DD} / Bit-Line Sensing Scheme in CMOS DRAMs," in *IEEE Journal of Solid-State Circuits*, vol. SC19, p. 451, August 1984.
- [18] E. Adler; J. K. DeBrosse; S. F. Geissler; S. J. Holmes; M. D. Jaffe; J. B. Johnson; C. W. Koburger, III; J. B. Lasky; B. Lloyd; G. L. Miles; J. S. Nakos; W. P. Noble, Jr.; S. H. Voldman; M. Armacost; and R. Ferguson; "The Evolution of IBM CMOS DRAM Technology;" *IBM Journal of Research and Development*, vol. 39, pp. 167–188, March 1995.
- [19] R. Kraus, "Analysis and Reduction of Sense-Amplifier Offset," in *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 1028–1033, August 1989.
- [20] M. Asakura, T. Ohishi, M. Tsukude, S. Tomishima, H. Hidaka, K. Arimoto, K. Fujishima, T. Eimori, Y. Ohno, T. Nishimura, M. Yasunaga, T. Kondoh, S. I. Satoh, T. Yoshihara, and K. Demizu, "A 34ns 256Mb DRAM with Boosted Sense-Ground Scheme," 1994 *IEEE ISSCC Digest of Technical Papers*, pp. 140–141.
- [21] T. Ooishi, K. Hamade, M. Asakura, K. Yasuda, H. Hidaka, H. Miyamoto, and H. Ozaki, "An Automatic Temperature Compensation of Internal Sense Ground for Sub-Quarter Micron DRAMs," 1994 Symposium on VLSI Circuits Digest of Technical Papers, pp. 77–78.

- [22] K. Noda, T. Saeki, A. Tsujimoto, T. Murotani, and K. Koyama, "A Boosted Dual Word-line Decoding Scheme for 256 Mb DRAMs," 1992 Symposium on VLSI Circuits Digest of Technical Papers, pp. 112–113.