NONVOLATILE MEMORY TECHNOLOGIES WITH EMPHASIS ON FLASH

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

IEEE Press Editorial Board
Mohamed E. El-Hawary, *Editor in Chief*

R. Abari	T. Chen	R. J. Herrick
S. Basu	T. G. Croda	S. V. Kartalopoulos
A. Chatterjee	S. Farshchi	M. S. Newman
	B. M. Hammerli	

Kenneth Moore, *Director of IEEE Book and Information Services (BIS)*
Catherine Faduska, *Senior Acquisitions Editor*
Jeanne Audino, *Project Editor*

IEEE Components, Packaging, and Manufacturing Technology Society, Sponsor
CPMT Liaison to IEEE Press, Joe E. Brewer

IEEE Electron Devices Society, Sponsor
EDS Liaison to IEEE Press, Joe E. Brewer

Technical Reviewers
R. Jacob Baker, *Boise State University*
Ashok K. Sharma, *NASA/Goddard Space Flight Center*

NONVOLATILE MEMORY TECHNOLOGIES WITH EMPHASIS ON FLASH

A Comprehensive Guide to Understanding and Using NVM Devices

Edited by

Joe E. Brewer

Manzur Gill

IEEE Press Series on Microelectronic Systems
Stuart K. Tewksbury and Joe E. Brewer, *Series Editors*

IEEE Components, Packaging, and Manufacturing Technology Society, *Sponsor*

IEEE Electron Devices Society, *Sponsor*

IEEE PRESS

WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2008 by the Institute of Electrical and Electronics Engineers, Inc.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey. All rights reserved.
Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978)750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permission.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability of fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com

Library of Congress Cataloging-in-Publication Data is available.

ISBN 978-0471-77002-2

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

FOREWORD

The story of Flash memory is one of a unique technology that was almost a failure. Only after three unsuccessful attempts did Flash succeed in the marketplace. To succeed, Flash had to solve several technical problems and had to create a market for itself. Once these issues were addressed, Flash quickly became the highest volume nonvolatile memory displacing the EPROM only 5 years after its successful entry in the market. Today, Flash is challenging DRAM for the highest volume semiconductor memory used in the world. While the market forecast for Flash is bright, the technology is approaching fundamental limits in its scalability.

Flash memory was born in a time of turmoil during a semiconductor recession in the mid-1980s. The highest volume writeable nonvolatile memory, the EPROM, was under extreme price pressure due to a maturing market with over 20 competitors. In addition, customers were growing weary of the UV erase required to reprogram an EPROM. They wanted the electrically erase capability of the new high-priced EEPROM technology at EPROM prices. This spurred Toshiba, SEEQ, and Intel to search for the "holy grail of nonvolatile memory," the single-transistor electrically erasable memory. In 1985, Toshiba was the first to announce a single-transistor electrically erasable memory and coined the name "Flash memory" as the new device erased in a "flash." Unfortunately, Toshiba's initial Flash product was difficult to use and, as a result, a market failure. SEEQ followed a year later with another complex Flash device that did not succeed in the market. Meanwhile, Intel took a diversion in attempting to develop a single-transistor EEPROM technology by partnering up with Xicor, one of the early EEPROM memory pioneers. While the single-transistor EEPROM looked good on paper, the reality was the cell operating window was nonexistent and the partnership was dissolved. Fortunately for Intel, a parallel internal development on an EPROM-based Flash memory technology was started as a "skunkworks project." By 1985, Intel had a working 64-kb Flash memory in the lab. But to everyone's surprise, the Intel EPROM business unit was not interested in commercializing Flash, claiming that the market would not accept it based on Toshiba's lack of success and a fear of Flash cannibalizing Intel's own EPROM business. If it were not for Gordon Moore and a band of very dedicated pioneers, Intel would never have entered the Flash market. In 1986, Intel formed a separate Flash business unit and introduced a 256-kb Flash memory 2 years later. With over 95% of its manufacturing steps the same as an EPROM, the Intel Flash technology was able to quickly ramp up in volume using existing EPROM fabs. To ensure market success, Intel designed its 256-kb Flash as an EPROM replacement by having the same package pinout and control signals as EPROM devices.

Flash was able to easily cannibalize EPROM embedded applications where cost-effective reprogramming was required. One of the very first commercial Flash memory applications was on an adjustable oil well drill bit. Clearly, the oil drillers did not want to pull up the drill bit every time they needed to adjust it. Other

IPR2025-00213/PHI EX. 1026 Page 5 of 31

1.2 ELEMENTARY MEMORY CONCEPTS

All information processing can be viewed as consisting of the sequential actions of sensing, interpreting/processing, and acting. These actions cannot be accomplished without somehow remembering the item of interest at least long enough to allow the operations to take place, and most likely much longer to allow convenient use of the raw data and/or the end results.

The length of time that the memory can retain the data is the property called *retention*, and the *unpowered retention* time parameter is the measure of *nonvolatility*. A volatile memory will typically have a worst-case retention time of less than a second. A nonvolatile memory is usually specified as meeting a worst-case unpowered retention time of 10 years, but this parameter can vary from days to years depending on the specific memory technology and application.

Integrated circuit nonvolatile memories are frequently classified in terms of the degree of functional flexibility available for modification of stored contents. Table 1.1 summarizes the categories currently in frequent use [1]. This class of memory was evolved from ultraviolet (UV) erasable read-only memory (ROM) devices, and thus the category labels contain "ROM" as a somewhat awkward reminder of that heritage.

Flash memory [2] is an EEPROM where the entire chip or a subarray within the chip may be erased at one time. There are many variants of Flash, but present-day production is dominated by two types: NAND Flash, which is oriented toward data-block storage applications, and common ground NOR Flash, which is suited for code and word addressable data storage.

In general, information processing requires memory, but it is not at all clear that any constraints are placed on the structure or location of the storage relative to the processing elements. That is, the memory may be a separate entity and entirely different technology than the logic, or it may be that the logic is embedded in the memory and be a technology compatible with the logic, or any combination thereof.

At the heart of every memory is some measurable attribute that can assume at least two relatively stable states. Many common memory devices are *charge based* where charge can be injected into or removed from a critical region of a device, and

TABLE 1.1. Nonvolatile Memory Functional Capability Classifications

Acronym	Definition	Description
ROM	Read-only memory	Memory contents defined during manufacture and not modifiable.
EPROM	Erasable programmable ROM	Memory is erased by exposure to UV light and programmed electrically.
EEPROM	Electrically erasable programmable ROM	Memory can be both erased and programmed electrically. The use of "EE" implies block erasure rather than byte erasable.
E^2PROM	Electrically erasable programmable ROM	Memory can be both erased and programmed electrically as for EEPROM, but the use of "E^2" implies byte alterability rather than block erasable.

the presence or absence of the charge can be sensed. The process of setting the charge level is called *writing*, and the process of sensing the charge level is called *reading*. Alternatively, the write operation may be referred to as the *store* operation, and the read operation may be called the *recall* operation.

Dynamic random-access memory (DRAM), a volatile technology, uses charge stored on a capacitor to represent information. Charge levels greater than a certain threshold amount can represent a logic ONE, and charge levels less than another threshold amount can represent a logic ZERO. The two critical levels are chosen to assure unambiguous interpretation of a ZERO or ONE in the presence of normal noise levels. (Here the higher charge level has been called a ONE, but it is arbitrary which level is defined to be the ONE or ZERO.)

Leakage currents and various disturb effects limit the length of time that the capacitor can hold charge, and thus limits "powered" retention to short periods. The word "dynamic" in the name "DRAM" indicates this lack of ability to hold data continuously even while the circuit is connected to power. Each time the data is read, it must be rewritten in order to assure retention, and regular data refresh operations must be performed when the cell is idle. Worst-case retention time (i.e., the shortest retention time for any cell within the chip) is typically specified as about 60 ms. DRAM is a volatile memory in terms of "unpowered" retention because the charge is not maintained when the circuit power supply is turned off.

Flash memory makes use of charge stored on a floating gate to accomplish nonvolatile data storage. Figure 1.1 provides a cartoon cross-section sketch of a floating-gate transistor and its circuit symbol representation. The floating-gate electrode usually consists of a polysilicon layer formed within the gate insulator of a field-effect transistor between the normal gate electrode (the control gate) and the channel. The amount of charge on the floating gate will determine whether the transistor will conduct when a fixed set of "read" bias conditions are applied. The fact that the floating gate is completely surrounded by insulators allows it to retain charge for a long period of time independent of whether the circuit power supply voltage is present. The act of reading the data can be performed without loss of the information.

Figure 1.2 compares an imaginary idealized transistor with no charge layer in the gate insulator with a transistor that has a charge per unit area, Q, at distance d from the silicon channel surface. The impact of the charge on the threshold voltage depends on the amount of charge per unit area, its distance from the silicon surface, and the permittivity of the insulator between the charge and the silicon. In a Flash device the floating gate provides a convenient site for the charge, but other means may serve the same purpose. For example, in silicon oxide nitride oxide silicon (SONOS) transistors the charge will reside in traps within the nitride layer.

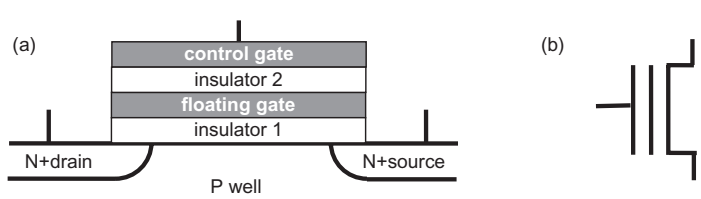

Figure 1.1. Floating-gate transistor: (*a*) elements of the transistor structure and (*b*) circuit symbol.

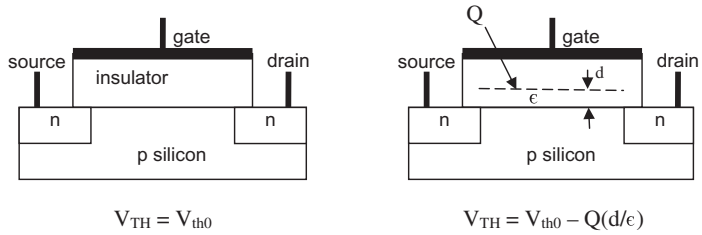

Figure 1.2. V_{TH} shift due to charge in gate insulator.

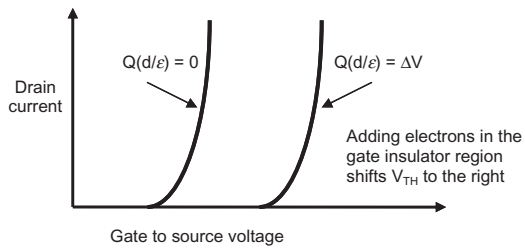

Figure 1.3. Shift of current–voltage characteristics because of inserted charge.

The threshold voltage, of course, impacts how the source-to-drain current of the transistor will change as a function of change in the gate-to-source voltage. Figure 1.3 shows how the characteristic curves can be made to shift as a function of the stored charge. As electrons are added to the charge within the gate insulator region, the curve will move in a positive direction.

For some memory technologies the process of reading destroys the data. This is referred to as *destructive readout* (DRO). For other technologies, Flash, for example, readout can be accomplished without significantly disturbing the data. This is referred to as *nondestructive readout* (NDRO). DRO memory has the disadvantage of requiring that every read operation must be followed by a write operation to restore the data.

Over time some forms of memory organization have been so firmly established that most engineers immediately assume those structures and the parameters that characterize those structures as being the norm. Probably the most pervasive assumption is that the information to be stored and recalled is in a digital binary format.

There are several schemes where a single transistor may be used to store more than just one bit. One approach is to store charge in physically separated parts of the device that can be sensed separately. Currently, the most common example of this concept is the NROM cell discussed later in this chapter. Another approach is to interpret the amount of charge stored in one physical location in the device as a representation of a multibit binary number. In this case the sensing process must reliably distinguish between different quantities of stored charge and the readout process must generate the corresponding binary number.

Consider the "one physical location" approach. A 1-bit-per-cell arrangement is a robust form of storage that allows relaxed margins and comparatively simple

sensing circuitry. The read current needs only to be unambiguously above or below a preset value in order to establish whether a ONE or a ZERO was stored. For a 2-bit-per-cell memory, the recall process must reliably distinguish between four preset levels of charge, and the readout circuitry must translate the detected level into a 2-bit digital format. The storage process and the protection of the cell from disturb conditions must accurately set and maintain those four levels under all operating and nonoperating storage conditions. Considering that the usual requirement is for nonvolatile data retention for 10 years, assuring stability of charge levels and circuit characteristics is quite a challenge. Of course, the complexity rapidly increases as a cell is required to store larger numbers of bits. For example, a 4-bit-per-cell memory must reliably cope with 16 levels and still meet all specifications.

While the heart of a semiconductor memory is the cell, the surrounding circuitry is the mechanism that makes it usable. For economic reasons, cells are packed as close together in a rectangular planar array as available integrated circuit technology and noise management concerns will allow. This X, Y array arrangement contains the cells and conductive lines that allow access to each individual cell.

The lines that run in the X direction (rows) are called *wordlines*, and they are used to select a row of cells during the write or read operations. Wordlines tie to the control gates of the cells in a row. The lines that run in the Y direction (columns) are called *bitlines*. As shown in Figure 1.4, bitline connections for the NOR array architecture are tied to drain terminals of devices in a column. One end of a bitline connects to power and then goes through the array to sensing and writing circuitry. Thus the wordlines activate a specific group of cells in a row, and the bitlines for each intersecting column connect those cells to read and write circuits.

In the example of a NOR architecture, the cells in a column are connected in parallel where all the drain terminals tie to a bitline and all the source terminals tie to a common source line (ground). In this configuration a positive read mode voltage on one wordline while all other wordlines are at zero volts will result in a bitline current that is a function only of the selected row of cells.

This, of course, assumes that a zero control gate-to-source voltage actually turns off the unselected rows. For the NOR organization it is important that the process of initializing the array, called *erase*, not proceed to the point of overerasing transistors to the extent that the threshold voltage becomes negative and the transistors change from operation in the enhancement mode to operation in the depletion mode.

The NAND array architecture, shown in Figure 1.5, achieves higher packing density. Here the bitlines are formed using series-connected strings of cells that do not require contact holes. A string is typically 8, 16, or 32 cells long. If other strings

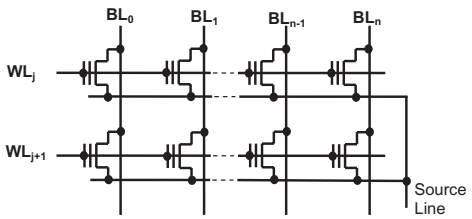

Figure 1.4. NOR array architecture.

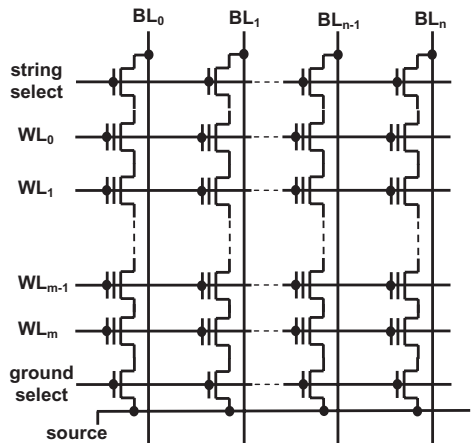

Figure 1.5. NAND array architecture.

were to be tied to the same bitlines, they would be connected in parallel between the bitline and the common source line in a manner similar to individual cells in the NOR architecture.

Reading and writing a NAND device is more complicated than dealing with a NOR device. The wordlines are used to select the transistor of interest in the string. To access data from a string, the reading process requires that all nonselected transistors be turned on while only the selected transistor is allowed to influence the current flow through the string. In contrast to the NOR architecture, it is not objectionable to allow the transistors to be shifted into depletion mode. There are, however, problems if transistors are shifted too far in the enhancement direction. It is important that the distribution of threshold voltages for the programmed state be limited to a specified design range in order to assure proper circuit operation.

There are a number of basic principles of operation that apply to both NOR and NAND organized devices. For a single read operation the individual bits that form a word are made to appear in an input/output register. For a single write operation the word present in an input/output register is used to determine the data inserted into the cells for that word. Reading or writing processes must be designed such that unselected cells are not disturbed while the selected cells are operated on. The design of the array must contend with basic circuit design issues associated with driving heavily loaded transmission lines as well as assuring proper operation of each individual cell.

In order to select a given row and column an integrated memory device is usually provided with a binary address word from external circuitry. The address word is routed to address decoder circuitry that is tightly tied to the sides of the array, and designed to drive the word and bitlines. For reasons of management of loading and data grouping considerations, large memory chips are usually partitioned into many arrays.

A modern integrated memory device incorporates control circuitry that accepts relatively simple commands as inputs and generates timed sequences of signals to accomplish writing, reading, and various other modes of operation. Also, the writing

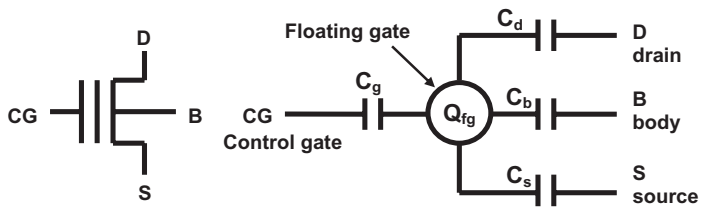

Figure 1.9. Capacitor model for a floating-gate transistor.

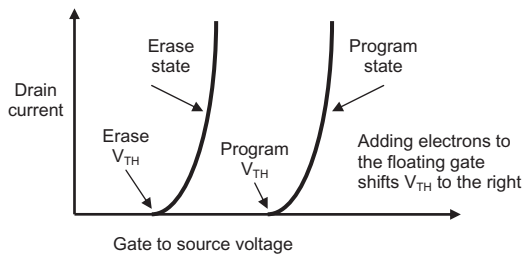

Figure 1.10. Erase state and program state current–voltage characteristics.

coupling ratio, k_d, would be C_d/C_t. In a similar fashion the coupling coefficients k_b and k_s would be C_b/C_t and C_s/C_t. The voltage on the floating gate resulting from pulses applied to the four terminals would be

$$V_{fg} = k_g V_{CG} + k_d V_D + k_s V_S + k_b V_B$$

The usual process of writing (storing) data into a Flash memory requires two operations. First, all of the cells in a common tub (i.e., a sector) are "erased" to initialize the state of the cells. Erasing refers to the removal of all charge from the floating gates. By convention this is usually taken to mean that all of the cells have been cleared to a ONE state. Second, the cells within the tub that are selected by a row address are "programmed" to ZERO or left at ONE in accord with the input data signals. The programming operation may continue over row addresses until all of the data sites in the sector have been programmed.

Erasing moves the threshold voltage in a negative direction while writing moves the threshold voltage in a positive direction. Figure 1.10 shows the relative relationship of the erase and program states in terms of the resulting current–voltage characteristics. Both the magnitude of shift and the statistical distribution of threshold voltages after an erase or program is a major design issue.

The erase procedure is complicated by several concerns. First, erasing (removal of electrons) shifts the threshold voltage negatively from a positive value toward a value nearer to zero. If continued too far, the threshold voltage can go through zero and become negative. This is referred to as *overerase*, and the transistor changes from enhancement mode to depletion mode. In NOR arrays depletion must be avoided in order for the array to work properly.

Second, the erase process is sensitive to the initial state of the transistor. Erase voltages are applied to transistors located in the same tub simultaneously. If some

acteristics that define the ultimate retention potential of the approach. Natural decay tracking points to very large retention on the order of thousands or millions of years. Natural decay is so slow as to not be a factor in determining practical retention specifications. At the present time a typical unpowered retention specification is 10 years.

Defects associated with materials, details of device geometry, or aspects of circuit design can impact retention. Each of these three potential problem areas can result in the addition or removal of charge to/from the floating gate. Gate insulator or interpoly insulator defects are typical causes of retention degradation. Phenomena associated with ionic contamination or traps can also be contributing factors. Management of these issues is a function of the general state of the integrated circuit reliability arts and of the specific practices and equipment of given manufacturers.

1.3.4 Endurance

Achievement of nonvolatility depends on exploitation of some natural phenomena peculiar to a given technology approach. In most nonvolatile technologies the normal processes employed to write and/or read cells will result in stresses that eventually degrade the properties of the memory or disturb the contents of the memory. *Endurance* is the term used to describe the ability of a device to withstand these stresses, and it is quantified as a minimum number of erase–write cycles or write–read cycles that the chip can be expected to survive. For quite a number of years the industry has used 100,000 cycles as the minimum competitive endurance requirement.

Knowledge of the reliability physics of the memory technology is essential in order to develop meaningful endurance specifications. The endurance capabilities of a device are a function of both the intrinsic properties of the technology and the quality control of the production line. The impact of cycling stress on retention is a key aspect of endurance, and window closure or shift is a concern. The *window* for a Flash memory is the difference between the erased state threshold voltage and the programmed state threshold voltage. This is the difference that must be reliably detected by a readout sense amplifier.

Both retention and endurance pose difficult test and verification quandaries. Economics and the limits of the human life span make direct observation of retention periods of decades impractical. The process of testing endurance by cycling each part implies that the testing would consume much of the useful life time of the product.

1.4 FLASH MEMORY AND FLASH CELL VARIATIONS

Flash has been used as the example technology throughout the discussions above where a simple stacked gate transistor was used to illustrate several points. In practice the design of Flash cells has several flavors each of which has advantages and disadvantages. Figure 1.11 groups some of the existing Flash varieties by addressing method and by program and erase technique (see Figure 5 in Bez et al. [2]). As mentioned earlier, the common ground NOR used for both code and data storage, and the NAND used for mass storage, presently dominate the market.

ment of comparative technology potential that will be useful to the semiconductor industry for making research investment decisions.

REFERENCES

1. IEEE Std. 1005-1998, IEEE Standard Definitions and Characterization of Floating Gate Semiconductor Arrays, Feb. 1999.
2. R. Bez, E. Camerlenghi, A. Modelli, and A. Visconi, "Introduction to Flash Memory," *Proc IEEE*, Vol. 91, pp. 489–502, Apr. 2003.
3. F. Masuoka, M. Assano, H. Iwahashi, T. Komuro, and S. Tanaka, "A New Flash EEPROM Cell Using Triple Polysilicon Technology," *IEEE IEDM Tech. Dig.*, pp. 464–467, 1984.
4. M. Gill, and S. Lai, "Floating Gate Flash Memories," in W. D. Brown and J. E. Brewer (Eds.), *Nonvolatile Semiconductor Memory Technology*, IEEE Press, Piscataway, NJ, 1998.
5. F. Masuoka, M. Assano, H. Iwahashi, T. Komuro, and S. Tanaka, "A 256 K Flash EEPROM Using Triple Polysilicon Technology," *IEEE ISSCC Tech. Dig.*, pp. 168–169, 1985.
6. F. Masuoka, M. Assano, H. Iwahashi, T. Komuro, N. Tozawa, and S. Tanaka, "A 258 K Flash EEPROM Using Triple Polysilicon Technology," *IEEE J. Solid-State Circuits*, Vol. SC-22, pp. 548–552, 1987.
7. S. Mukherjee, T. Chang, R. Pan, M. Knecht, and D. Hu, "A Single Transistor EEPROM Cell and Its Implementation in a 512 K CMOS EEPROM," *IEEE IEDM Tech. Dig.*, pp. 616–619, 1985.
8. G. Verma and N. Mielke, "Reliability Performance of ETOX Based Flash Memories," *Proc. IRPS*, pp. 158–166, 1988.
9. V. N. Kynett, A. Baker, M. Fandrich, G. Hoekstra, O. Jungroth, J. Kreifels, S. Wells, and M. D. Winston, "An In-System Reprogrammable 256 K CMOS Flash Memory," *IEEE J. Solid-State Circuits*, Vol. 23, No. 5, pp. 1157–1163, Oct. 1988.
10. B. Eitan, P. Pavan, I. Bloom, E. Aloni, A. Frommer, and D. Finzi, "NROM: A Novel Localized Trapping, 2-Bit Nonvolatile Memory Cell," *IEEE Electron Device Lett.*, Vol. 21, No. 11, pp. 543–545, Nov. 2000.
11. B. Eitan, G. Cohen, A. Shappir, E. Lusky, A. Givant, M. Janai, I. Bloom, Y. Polansky, O. Dadashev, A. Lavan, R. Sahar, and E. Maayan, "4-Bit per Cell NROM Reliability," *IEEE IEDM Tech. Dig.*, pp. 539–542, Dec. 5, 2005.
12. G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, Vol. 38, No. 8, Apr. 19, 1965.
13. V. Kynett, M. Fandrich, J. Anderson, P. Dix, O. Jungroth, J. Kreifels, R. Lodenquai, B. Vajdic, S. Wells, M. Winston, and L. Yang, "A 90 ns One Million Erase/Program Cycle 1-Mbit Flash Memory," *IEEE J. Solid-State Circuits*, Vol. 24, No. 5, pp. 1259–1264, Oct. 1989.
14. A. Fazio, "A High Density High Performance 180 nm Generation ETOX™ Flash Memory Technology," *IEEE IEDM Tech. Dig.*, pp. 267–270, Dec. 5–8, 1999.
15. S. N. Keeney, "A 130 nm Generation High Density ETOX™ Flash Memory Technology," *IEEE IEDM Tech. Dig.*, pp. 2.5-1–2.5-4, Dec. 2–5, 2001.
16. G. Atwood, "Future Directions and Challenges for ETOX Flash Memory Scaling," *IEEE Trans. Devices Mater. Reliabil.*, Vol. 4, No. 3, pp. 301–305, Sept. 2004.
17. R. Koval, V. Bhachawat, C. Chang, M. Hajra, D. Kencke, Y. Kim, C. Kuo, T. Parent, M. Wei, B-J. Woo, and A. Fazio, "Flash ETOXTM Virtual Ground Architecture: A Future Scaling Direction," paper presented at the 2005 Symposium on VLSI Technology Digest of Technical Papers, June 14–16, 2005, pp. 204–205.
18. J. E. Brewer, V. V. Zhirnov, and J. A. Hutchby, "Memory Technology for the Post CMOS Era," *IEEE Circuits Devices Mag.*, Vol. 21, No. 2, pp. 13–20, Mar.–Apr. 2005.

Flash devices, cycles are counted on a block basis and erases on one block does not affect the reliability or performance of the others. The cycle count is relevant because the device's performance specifications, such as block erase time or byte program time, are only guaranteed to hold up to the maximum erase cycling specification. This is because erasing a Flash block puts wear on the oxide layer insulating the floating gate, which stores the information in the chip. A widely held misunderstanding of the cycling spec is that the device will fail or cease operating once the max cycling spec has been reached. This is not true; in most cases, the device will continue operating well after the max cycling spec has been exceeded. The spec signifies the amount of erases for which the manufacturer can guarantee the device's performance specs.

Erase cycling specifications have improved dramatically since the introduction of Flash. The first bulk-erase Flash devices were rated for 1000 cycles, but 10,000-cycle specs were soon commonplace. The addition of embedded write state machines to Flash devices brought tighter control of the erase process and made 100,000 cycles possible. Today, typical devices are rated for 100,000 to 1,000,000 cycles, but improved media management techniques (see Section 2.4.1.7) have decreased the number of erases used in typical data storage processes, so 100,000 cycles is generally enough for most applications.

Suspends. One of the first issues with executing system code directly from Flash and expecting to write to that same Flash device is the *read while write* problem. This problem refers to the fact that a standard single-partition Flash device cannot be read while a write operation (program or erase) is in progress. Since the internal write state machine is manipulating the internal control signals and voltages to perform the write operation on the cells of the Flash array, those same Flash cells cannot be read at the same time. Because the system is usually architected such that CPU instructions are fetched from the Flash, a catch-22 situation is created as follows: Where does the CPU obtain the instructions it needs to control the Flash write operation when those same instructions will become unavailable as soon as the Flash write operation begins? This problem is magnified in a code+data application since writes of data to the Flash occur much more often than code modifications.

The standard method for dealing with the read-while-write issue is to copy the minimal code needed to write to the Flash into RAM and to execute from RAM during the write operation. However, the hole in this method is the inability to execute additional code from the Flash in the case that an interrupt occurs during the Flash write operation. To alleviate this constraint, write suspends were added to the Flash write state machine.

Write suspends allow the CPU to interrupt an in-progress program or erase operation, read from the Flash array, then resume the suspended operation. Recent Flash devices support erase suspend to read, program suspend to read, erase suspend to program, and even the nested erase suspend to program suspend to read. When executing the Flash write code from RAM, an interrupt handler can poll for interrupts and suspend the write operation to allow code execution in order to service an interrupt. This method of interrupt handling is sometimes called *software read-while-write*.

Suspends incur some performance penalties in the area of suspend latency and write performance. Suspend latency is the time the write state machine takes to

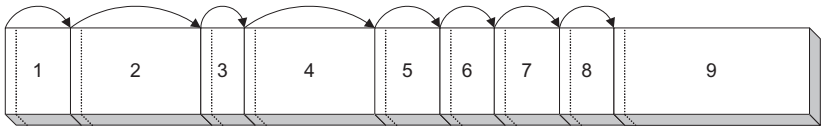

Figure 2.6. Example of linked-list media manager.

span multiple Flash erase blocks. See Figure 2.6, which illustrates the use of a linked list.

Advanced Media Management Functions. A media manager can also implement advanced features, such as reclamation or wear leveling, which are above the capability of standard Flash drivers.

Reclamation is an essential function of a media manager since it makes portions of the Flash memory usable again by erasing blocks. As information is updated in the Flash memory, the old information is not immediately erased (since you can only erase a block of Flash), but is marked "invalid." When enough of the bits in a block are marked invalid, the media manager will reclaim that block. This method of managing data storage allows many pieces of data to be stored and updated within a block before a reclaim and erase are necessary, which increases the effective block cycling of the Flash device. The steps necessary to reclaim a block of Flash include copying the valid information from the block to be reclaimed to a new block, updating the tables used to translate parameter numbers to a physical location in memory, and finally erasing the old block. While this is occurring, special attention must be paid to power loss recovery. If power were lost to the system in the middle of a reclaim, there is a possibility of losing information if proper power loss mechanisms are not used.

Wear leveling is useful in any system that writes information to a Flash device. If a media manager is not aware of the need for wear leveling, one block can tend to be used exclusively for a particular piece of information. This can wear out a Flash device earlier than it should, especially when a piece of information is updated at a rapid rate. A robust media manager keeps statistics on the number of times each block is erased and will occasionally move data around such that the piece of data being updated at a rapid rate is moved to a block that has not been erased as many times. Generally a wear-leveling algorithm will ensure that the erase counts of the block that has been erased the most number of times and the block that has been erased the least number of times stay close to each other. This extends the amount of time before the Flash reaches its cycling specification.

Flash Filing Systems. Finally, the most sophisticated form of Flash software is a filing system. A filing system not only makes Flash's large erase blocks transparent to the system, but allows the system to access data via file names and write, modify, and delete these files. Generally multiple files may be open at one time. A filing system can use a standardized translation protocol such as FTL (Flash translation layer) to transform disk-style file allocation table (FAT) file sectors to locations in the linear Flash memory or a filing system can use a proprietary method to create this translation. With a filing system, requests for files stored on the Flash memory appear to the system as though the files are stored on a standard disk drive. A filing system includes all of the features of a media manager in addition to its file capabilities.

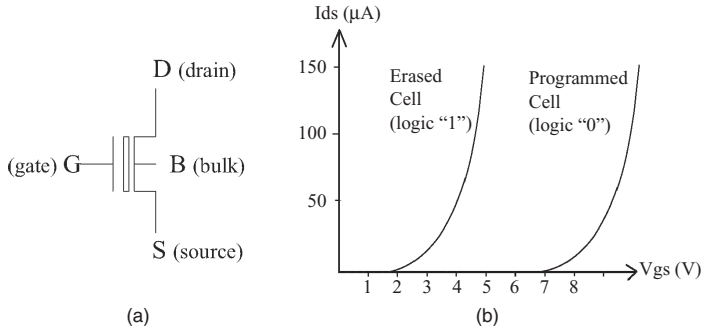

Figure 3.1. (*a*) Flash cell schematic symbol and (*b*) programmed and erased cell *IV* characteristics.

3.2.1 Cell Programming

Two basic physical mechanisms are usually exploited to program a Flash cell: channel hot electron (CHE) injection [4] and Fowler–Nordheim (FN) tunneling. Channel hot electron programming consists of applying a relatively high voltage, on the order of 4 to 6 V, to the cell drain D in Figure 3.1, a high voltage (8 to 11 V) to the cell gate G while source S and bulk B are kept at 0 V. With this biasing regime a fairly large current (0.3 to 1 mA) flows in the cell and the hot electrons generated in the channel acquire sufficient energy to jump the gate oxide barrier and get trapped into the floating gate. Most NOR-type Flash architectures adopt CHE programming. The CHE programming process takes a few microseconds to shift the cell threshold voltage from the erased value (about 2 V) to the programmed value (>7 V)—see Figure 3.1(*b*). Conventionally, a programmed cell stores a logic 0 value. FN tunneling programming is used in NAND-type [5], AND-type, and DINOR-type [6, 7] Flash architectures. FN programming is achieved by applying a high voltage of about 20 V to the cell gate while drain, source, and bulk are grounded. FN programming is slower than CHE programming. FN programming time is in the range of a few milliseconds. One important advantage of FN programming is that it requires very small programming current (<1 nA) per cell thus allowing many cells to be programmed at a time.

3.2.2 Cell Erase

Cell erasure is accomplished by FN tunneling. In a NOR-structured Flash memory array, for example, cell gate G is biased at a negative (with respect to bulk B potential) voltage of about –10 V; 4 to 6 V are applied to the drain D, source S is floated, and bulk B is kept at 0 V. With the above biasing regime a high electric field is applied across the gate oxide that pushes electrons out of the floating gate, thus reducing the device threshold voltage. An erased cell conventionally stores a logic 1 value. FN erase usually takes a few milliseconds.

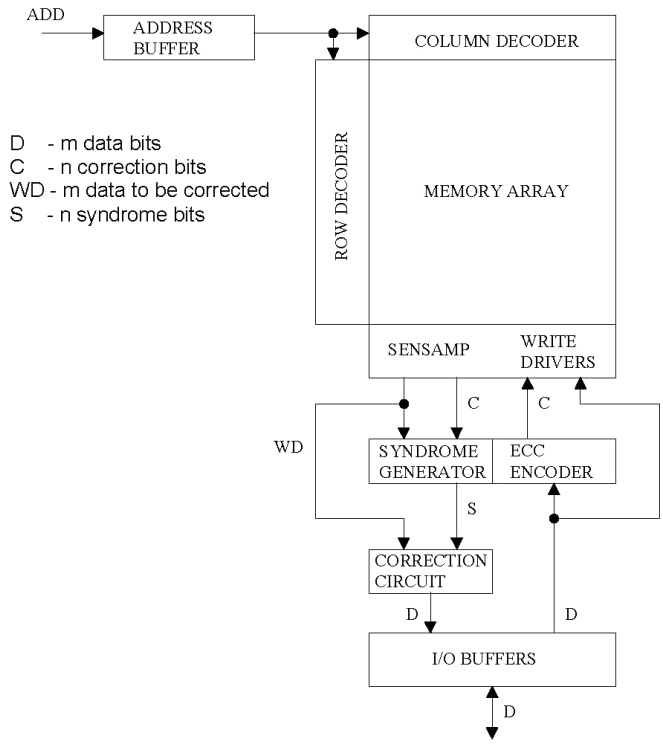

Figure 3.20. On-chip ECC for Flash memory.

3.5.1 On-Chip ECC and Endurance/Retention in Flash Memories

Error correction coding can be used in Flash memories to increase endurance and improve retention. After a certain number of erase/program operations a Flash memory can have defective cells due to oxide degradation or charge trapping. As soon as this happens, and provided that the ECC circuit is able to repair the amount of failing bits, the repair will be automatically performed. The apparent endurance of the chip is thus extended to a greater number of cycling operations.

After a certain lifetime a Flash memory can have a degradation of the cell thresholds due to charge loss or gain and, therefore, can undergo hard or random errors. On-chip ECC can correct these errors as soon as they happen in the application. The apparent data retention of the Flash memory is thus improved.

Redundancy is not useful to fix errors due to endurance or data retention because it can only be set up for fixed locations one time at the factory.

Flash memory ECC can be linked to circuitry to facilitate the performance of tests to screen chips that are weak for endurance [54] or retention. A Flash memory can in fact be provided with an internal address counter and an *internal repairs*

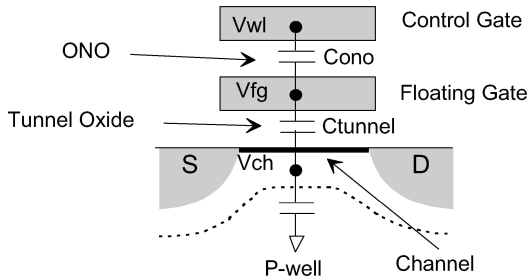

Figure 6.17. Capacitance model for self-boosted program-inhibit scheme.

With the SSL transistors turned on and the GSL transistors turned off, the bitline voltages for cells to be programmed are set to 0 V, while the bitline voltages for cells to be program inhibited are set to V_{cc}. When the program voltage is applied to the control gate of the selected cell, the large potential difference between the floating gate and channel causes FN tunneling of electrons for the programming cells. In program-inhibited cells, the V_{cc} bitline initially precharges the associated channel. When the wordlines of the unit NAND string rise (selected wordline to the program voltage and unselected wordlines to the pass voltage), the series capacitance through the control gate, floating gate, channel, and bulk are coupled and the channel potential is boosted automatically. Assuming a single boosted pass cell, and the model of Figure 6.17, the boosted channel voltage, V_{ch}, can be estimated as follows:

$$V_{ch} = \frac{C_{ins}}{C_{ins} + C_{channel}} V_{cg}$$

where C_{ins} is $C_{ono} || C_{tunnel}$.

It is calculated that the floating channel voltage rises to approximately 80% of the control gate voltage. Thus, channel voltages of program-inhibited cells are boosted to approximately 8 V when program and pass voltages are raised to 10 V. This self-boosted channel voltage is enough to prevent the program-inhibited cell from FN tunneling if the junction leakage current is lower than 1 nA.

The page program operation is explained by the miniarray of two blocks, as shown in Figure 6.18. At first, the 0-programmed bitlines are precharged to V_{cc}, and the 1-inhibited bitlines become 0 V as forced by the load data in the page buffers. In this case, BL0 and BL1 stand for 0-programmed bitline and 1-inhibited bitline, respectively. In case that the wordline, WL1, in the selected block is being programmed, the common gate line, S1, becomes the program voltage, V_{pgm}, and the other common gate lines (S0, S2, S3, . . . , S15) become the pass voltage, V_{pass}. The block select node, BSEL, is boosted to over $V_{pgm} + V_{th}$ by the local charge pump circuit, which is allocated in each block decoder. Here, V_{th} is the threshold voltage for the transfer gates of the wordlines so that the transfer gates are turned on in the linear region and V_{pgm} is smoothly applied to WL1. The other wordlines in the unselected blocks are "low" floating because the block select node, BSEL, is 0 V, however, the string select lines of SSL and GSL in the unselected blocks become 0 V by the grounded transistors of Tr1 and Tr2.

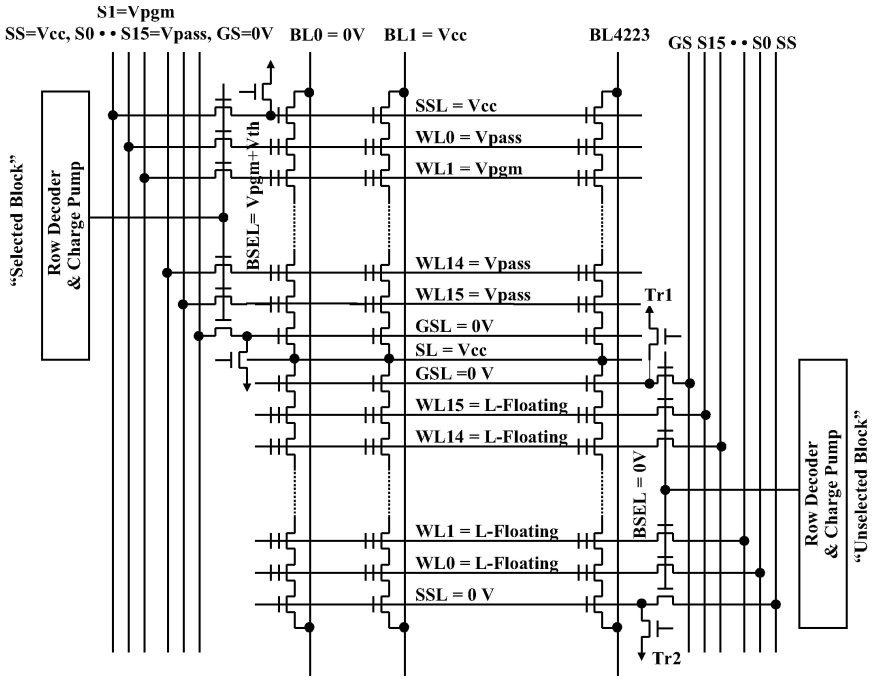

Figure 6.18. Page program operation.

6.3.4 Read Operation

In case of the read operation, the selected wordline and the pass wordlines in the selected block become 0 V and V_{read}, respectively, via the common gate lines, as shown in Figure 6.19. V_{read} is slightly higher than V_{cc} of 3.3 V. In the 64-Mbit NAND EEPROM, V_{read} is set to 4.5 V. The other wordlines in the unselected blocks are "low" floating, however, the string select lines of SSL and GSL in the unselected blocks are set to 0 V by the grounded transistors of Tr1 and Tr2. Therefore, the unselected NAND string doesn't flow any current.

6.4 PROGRAM THRESHOLD CONTROL AND PROGRAM V_t SPREAD REDUCTION

6.4.1 Bit-by-Bit Verify Circuit

In the NAND EEPROM, 528-byte memory cells are simultaneously programmed. If the memory cells are programmed with the same program time, the threshold voltages have a wide distribution resulting from the differences in the program characteristics of the memory cells. This wide program threshold voltage distribution causes difficulties in operating the NAND EEPROM with 3 V. The new verify circuit, which is composed of only two transistors, results in a simple intelligent program algorithm for 3-V-only operation [15, 16].

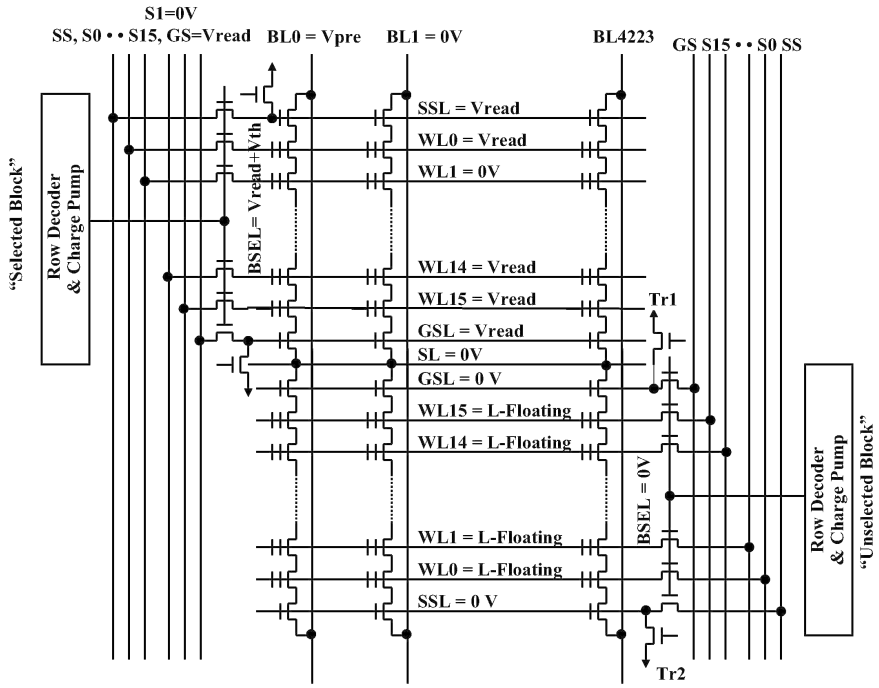

Figure 6.19. Read operation.

Figure 6.20 illustrates a circuit diagram for a bit-by-bit verify circuit and a read/write circuit. A bit-by-bit circuit, composed of only two transistors, is connected to each bitline. Two bitlines and bit-by-bit verify circuits share a common read/write circuit like in an open-bitline architecture of a DRAM. The read/write circuit acts as a flip-flop-type differential sense amplifier in the read operation and as a data latch circuit in the program operation. Figures 6.21, 6.22, and 6.23 show the clock timing diagrams of each operation of program, read, and verify-read in the case that a control gate of CG4 in the array in Figure 6.20(*a*) is selected.

In the program operation, the initial program data are loaded by a page sequence and latched into the read/write circuits. The memory cells, which share the same control gate, are programmed simultaneously. For 1-programming, the bitlines are charged up to 8 V for the power supply voltage of V_{rw} for the read/write circuits, which is pumped up from the external power supply of V_{cc}. For 0-programming, the bitlines are grounded. The program is accomplished by applying 20 V to the selected control gate of CG4 while 10 V is applied to the unselected control gates of CG1-3, 5-8, as shown in Figure 6.21.

In the read operation, the selected bitlines of BLai are precharged to $\frac{3}{5}V_{cc}$ and the dummy bitlines of BLbi are precharged to $\frac{1}{2}V_{cc}$ at t1 in Figure 6.22. After precharging, the unselected control gates of CG1-3, 5-8 are raised to V_{cc} while the selected control gate of CG4 is grounded <t2>. If the V_{th} of the selected memory cell is smaller than 0 V, a read current flows and the selected bitline voltage decreases below $\frac{1}{2}V_{cc}$ (1-read). Conversely, if the V_{th} is larger than 0 V, no cell read current

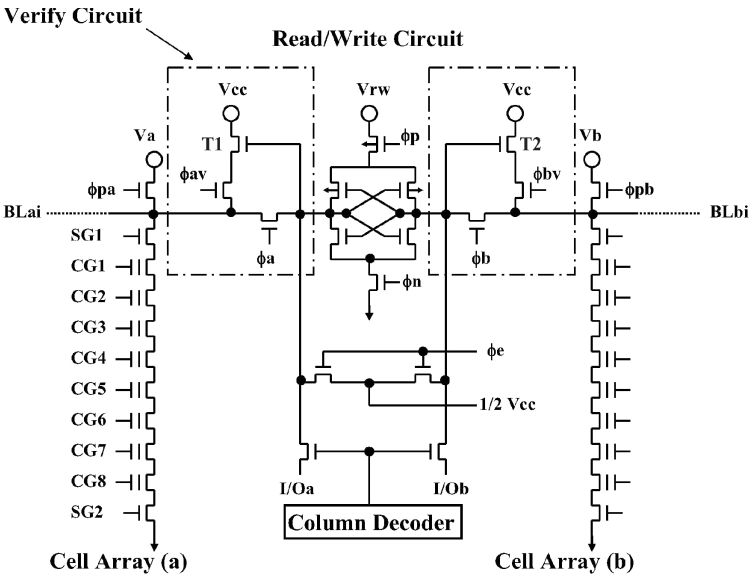

Figure 6.20. Bit-by-bit verify circuit and a read/write circuit.

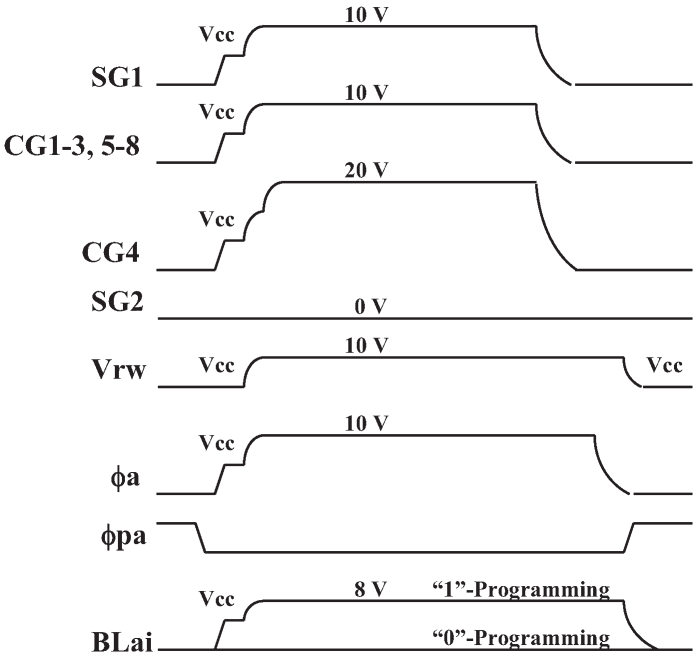

Figure 6.21. Program clock timing diagram.

1. An embedded Flash memory is designed as a module that can readily be integrated into the host logic device. A stand-alone Flash memory is a self-contained chip that has I/O pads or pins to interface with other logic devices.
2. A Flash module for embedded applications must meet and be optimized for special requirements on array size, memory configuration, speed, and particular bus protocol for the host logic device and is considered a custom or semicustom design. Stand-alone Flash memory products typically have a common set of features and specs.
3. A Flash memory module designed for embedded applications must provide special test features, such as scan path or joint test action group (JTAG) interface capabilities called for by the host logic device. A stand-alone Flash memory may not have these requirements.
4. A Flash memory module for embedded applications is typically designed to support extensive reuse of intellectual property (IP) for multiple product families and multiple products within a product family. For this reason, a Flash memory module must be designed with a built-in memory generator or compiler to provide ease for designing new derivatives. A stand-alone Flash chip is typically designed and optimized for one specific array configuration.
5. Design of an embedded Flash memory can be rather logic intensive due to the required memory control block (MCB) and bus interface unit (BIU). For this reason, behavioral model, synthesis, and automated place-and-route design approaches are employed to shorten the design cycle time. A stand-alone Flash memory on the other hand may rely extensively on custom design to minimize silicon area and maximize speed.
6. The embedded Flash memory module typically shares the same tester with the host logic device, which costs more than a dedicated memory tester for stand-alone Flash memory because of high speed and high pin-count requirements for the host logic device. For this reason design for test time reduction is more critical for the embedded Flash memory than for the stand-alone Flash memory.
7. An embedded Flash module core once designed is to be reused by many different design groups from different companies, with varied degrees of knowledge and expertise in Flash memory technology and, therefore, good portability, good documentation, and ease of use are critical to the Flash memory core design. For the stand-alone Flash memory, there is typically one design for a given generation of technology and little reuse, if any.
8. An embedded Flash memory has less direct accessibility, observability, and controllability than stand-alone Flash memory due to lack of accessible I/O pins. Also, supply noise and substrate operating temperature may increase for an embedded Flash memory product due to integration of other modules on the same chip.
9. In order to serve as wide a range of product families as possible, the embedded Flash core (FMC) must be designed to meet the highest performance and density required among the product families to be served. Also, performance margins required for the embedded Flash memory have to be

higher than that for the stand-alone Flash memory because, unlike the stand-alone flash memory that can be specified for multiple speed grades, there is typically only one speed grade for the host logic device.

10. An embedded Flash memory module may involve special requirements of very high speed operation, very low voltage operation, or ease of design and manufacturing not available for a stand-alone Flash memory. For this reason, selection of the Flash memory cell, Flash memory process, and Flash array architecture for embedded application can be different from the stand-alone Flash memory and are optimized for the particular application space.

9.5.2 Flash Module Design for Embedded Applications

As previously stated, a Flash memory module for embedded applications consists of a Flash memory core (FMC) and a bus interface unit (BIU). The FMC comprises two key parts, a Flash memory array (FMA) and a memory control block (MCB). The FMA contains all the basic array elements that include cells, decoders, sense amplifiers, and program data buffers. The MCB contains all the essential control logic, register sets, state machines, high-voltage charge pumps, voltage regulators, level shifters, voltage switches, and I/O buffers required to support proper FMA operations. A simplified block diagram of the Flash module is shown in Figure 9.13, and Figure 9.14 shows a simplified FMA.

Flash memory core operations include array program, array erase, program/erase verify read, regular read, and any special test operations for device characterization or quality/reliability enhancement. The BIU, as the name implies, provides all interface needs between the Flash memory core and the host logic device, or specifically the bus of the host logic device such as MCU or DSP, to which the Flash memory module is connected. FMC is the essential portion of an embedded Flash memory module and contains all basic elements required for a stand-alone Flash memory. Figure 9.15 is a block diagram for the FMC showing key circuit blocks required for the MCB. The FMC is typically designed to be portable, reusable, and easily reconfigurable for other embedded Flash memory applications.

While the FMC can be a common design for many applications, the BIU is typically application or product family specific and must be designed to meet

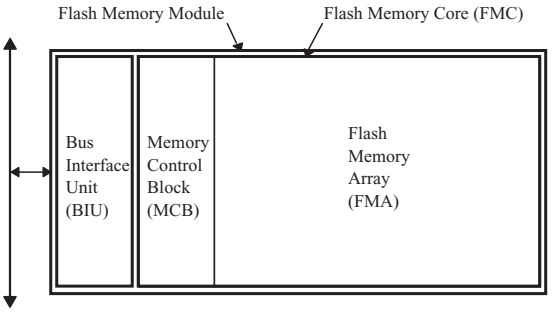

Figure 9.13. Embedded Flash memory module consists of bus interface unit (BIU) and Flash memory core (FMC) that contains Flash memory array (FMA) and memory control block (BCB).

floating gate. For single-bit/cell devices, the transistor either has little charge (<5000 electrons) on the floating gate and thus stores a 1 or it has a lot of charge (>30,000 electrons) on the floating gate and thus stores a 0. When the memory cell is read, the presence or absence of charge is determined by sensing the change in the behavior of the memory transistor due to the stored charge. The stored charge is manifested as a change in the threshold voltage of the memory cell transistor. Figure 12.5 illustrates the threshold voltage distributions for a half-million cell ($\frac{1}{2}$-Mc) array block. After erasure or programming, the threshold voltage of every memory cell transistor in the $\frac{1}{2}$-Mc block is measured, and a histogram of the results is presented. Erased cells (data 1) have threshold voltages less than 3.1 V, while programmed cells (data 0) have threshold voltages greater than 5 V.

The charge storage ability of the Flash memory cell is a key to the storage of multiple bits in a single cell. The Flash cell is an analog storage device not a digital storage device. It stores charge (quantized at a single electron) not bits. By using a controlled programming technique, it is possible to place a precise amount of charge on the floating gate. If charge can be accurately placed to one of four charge states (or ranges), then the cell can be said to store 2 bits. Each of the four charge states is associated with a 2-bit data pattern. Figure 12.6 illustrates the threshold voltage distributions for a $\frac{1}{2}$-Mc block for 2-bit-per-cell storage. After erasure or precise

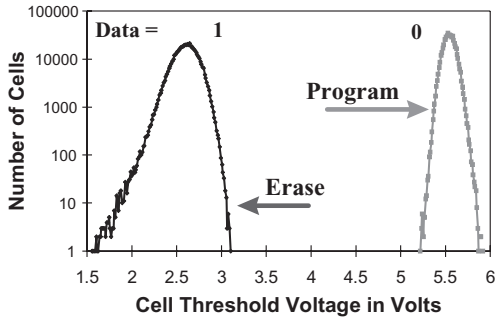

Figure 12.5. Single-bit/cell array threshold voltage histogram.

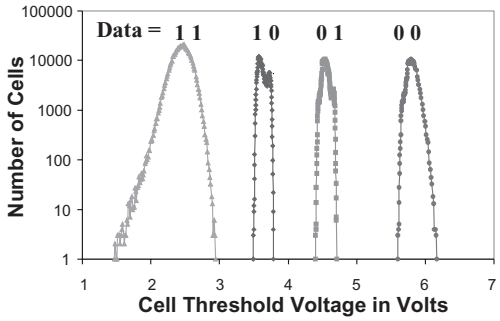

Figure 12.6. Two-bit/cell array threshold voltage histogram.

programming to one of three program states, the threshold of each of the $\frac{1}{2}$-Mc blocks is measured and plotted as a histogram. Note the precise control of the center two states, each approximately 0.3 V (or 3000) electrons in width.

Higher bit/cell densities are possible by even more precise charge placement control. Three bits per cell requires eight distinct charge states; 4 bits per cell requires 16 distinct charge states. In general, the number of states required is equal to $2N$ where N is the desired number of bits.

The ability to precisely place charge on the floating gate and at some later time sense the amount of charge that was stored has required substantial innovations and extensive characterization and understanding of cell device physics, memory design, and memory test. The aspects are discussed later in this chapter.

12.3.2 Evolution of MLC Memory Technology Development

This section will outline the development of the Intel StrataFlash memory technology from conception in 1992 to production in 1997, highlighting the key innovations along the way. The 64-Mbit product initially introduced in 1997 differs markedly from the 1992 view of what a 2-bit/cell product might look like. Today, MLC products have become the mainstream Flash memory technology. The learning that has occurred over the years has enabled the development of a 2-bit/cell memory device that functionally looks identical to a 1-bit/cell device, far exceeding the capability that was considered possible when development began.

12.3.3 Multilevel Cell Concept

Storage of analog data in a floating-gate memory device is not a new concept. It was suggested as early as 1971 for erasable programmable ROM (EPROM) devices [1] and was implemented on E^2PROM devices for use in neural networks, voice recorders, and toys as early as 1982. These analog storage applications can tolerate a high error rate and thus do not place stringent requirements on the memory reliability or accuracy. Neural networks are, by their nature, fault tolerant. Voice storage and simple talking toys can tolerate a few lost bits without any audible impact. These high error rate lossy memories are generally not usable for mainstream digital storage and thus have had limited acceptance. The goal of the MLC program was to produce a 2-bit/cell digital storage technology capable of penetrating the larger nonvolatile memory market, enabling the growth of new digital Flash memory applications.

12.3.3.1 The 1992 View of MLC. In the early 1990s, Flash memory was considered as a potential replacement for hard disks at lower densities for applications that require small, rugged, and low-power storage. One of the main issues for use of Flash in this application was the high cost of the Flash memory as compared to that of magnetic storage. A lower cost Flash memory was required. The hard disk requirements are much relaxed over silicon memory due to the inclusion of error correction in the hard disk subsystem, the block transfer of data (no byte access), and the relative low read performance. Multilevel cell technology appeared to be an ideal solution for the solid-state disk, addressing the lower cost through 2-bit-per-cell (and later 3 or 4 bits/cell) technology. The use of error correction and the large block transfer of data in the solid-state disk would address any reliability

issues with multilevel storage. The Intel MLC program was thus started with a goal of a high-density, low-cost, solid-state disk.

The basic techniques for accurate charge placement and sensing were developed in the lab and implemented into a 32-Mbit silicon test chip. During this time frame, the three major challenges for multibit storage were identified:

- *Precise Charge Placement*: The Flash memory cell programming must be very accurately controlled, requiring a detailed understanding of the physics of programming as well as the control and timing of the voltages applied to the cell.
- *Precise Charge Sensing*: The read operation of an MLC memory is basically an analog to digital conversion of the analog charge stored in the memory cell to digital data, a concept new to memory devices.
- *Stable Charge Storage*: Meeting the data retention goals would require the stored charge to be stable with a leakage rate of less than one electron per day.

The 32-Mbit test chip clearly demonstrated the ability to store multiple bits in a single cell. Based on the functionality of this device, the MLC technology was announced in 1994.

12.3.3.2 The First MLC "Product." With the knowledge gained from the 32-Mbit test chip, the first attempt at a 2-bit/cell storage product was started. This device was aimed at the solid-state disk goal. The solid-state disk system would include error correction and would generate nonstandard voltages to interface to the 2-bit/cell memory device. A special dc-to-dc (direct current) voltage converter was commissioned that would generate 12 V ± 1% and 5.5 ± 1%. The MLC part required these precise supply voltages to perform the accurate program and read operations. Also designed to be integrated with the other control logic of the solid-state disk was an error corrector. A paper based on this 32-Mbit MLC memory was presented at the prestigious International Solid State Circuits Conference (ISSCC) in 1995 [2], winning the best paper of the conference award. The 32-Mbit chip became the workhorse for the MLC technology development effort, demonstrating the ability of MLC to meet stringent reliability requirements and to produce yield equivalent to single bits per cell Flash memories. It was also used to develop the MLC testing and to debug the manufacturing process for test and packaging.

12.3.3.3 Question of Reliability. The primary concern for MLC was the reliability of the storage of the multiple charge states. Charge states would be separated by a few thousand electrons in an MLC device, and a loss of one electron per day from the floating gate could result in a bit error after 10 years of storage. To understand the detailed physics of charge storage, a large experiment was started to monitor the charge storage behavior of 200 billion cells (2×10^{11} cells). This massive experiment could resolve changes in the stored charge of as small as 100 electrons on all of the cells under evaluation. The rate of charge loss was accelerated through the use of elevated temperatures. The knowledge gained and models developed based on this experiment have resulted in changes to the design of both the product and the process, allowing removal of the error correction requirement for 2 bits per cells [3]. This data fundamentally changed the direction of the multibit storage program.

12.3.3.4 Removing the Constraints. Toward the end of 1995, the MLC project had grown from a small research effort to a full-blown program. Almost 2 years worth of reliability data was showing excellent performance, indicating that the error corrector was not required. The 32-Mbit device had demonstrated the viability of the circuit techniques and the device physics used for the precision program and read operations. Moreover, the yield was looking excellent, and the manufacturing issues were understood. Test circuits had demonstrated the ability to provide the required voltages and voltage regulation on the memory chip, eliminating the need for the external dc-to-dc converter. It became clear that the project could accomplish much more than the initial vision of a solid-state disk. The team believed that it was possible to remove the two major requirements initially envisioned for MLC: error correction and precision external power supplies. The solid-state disk market, while developing, had not reached the desired volume levels. The decision was made to not take the 32-Mbit device to production and focus on the design of an MLC 2-bit/cell part with functionality substantially equivalent to the standard 1-bit/cell products.

12.3.3.5 The 1997 View of MLC. The first 2-bit/cell Intel StrataFlash memory device was introduced in September of 1997, a 64-Mbit device. This device has functionality that is largely equivalent to the standard 1-bit/cell Flash products. A highlight comparison of the Intel StrataFlash memory features to an Intel 16-Mbit single-bit/cell product is shown in Table 12.1.

Read performance was in line with expectations for memories of 32- and 64-Mbit densities with about a 20% increase in read access time for a doubling of memory density. Two bits/cell doubles the erase block size as compared to 1 bit/cell since each cell now stores twice as much data. The power supply was maintained at the 5 V industry standard used at that time. The 2 bits/cell write performance was maintained equivalent to 1 bit/cell, even with the more complex (and slower) precision write algorithm, through the use of an 8-byte write buffer and a higher write bandwidth into the array. The 10,000 erase/write endurance specification was more than acceptable for virtually all Flash applications and easily justified by the reduced cost.

TABLE 12.1. Comparison of 1-Bit/Cell and 2-Bit/Cell Product Features

	1-Bit/CellFlash Memory	Intel StrataFlash 2-Bit/Cell Memory	
Density	16 Mbits	32 Mbits	64 Mbits
Read speed	100 ns	120 ns	150 ns
Block size	64 kbytes	128 kbytes	
Architecture	×8	×8 / ×16	
V_{cc} power supply (±10%)	5 V	5 V	
V_{pp} (program/erase voltage)	5 V or 12 V	5 V	
Effective write speed	6 μs/byte	6 μs/byte	
I_{ccr} (read current)	35 mA	55 mA	
$I_{ppw} + I_{ccw}$ (write current)	75 mA	90 mA	
Endurance	100,000 cycles	10,000 cycles	
Operating temperature	Extended	Commercial	

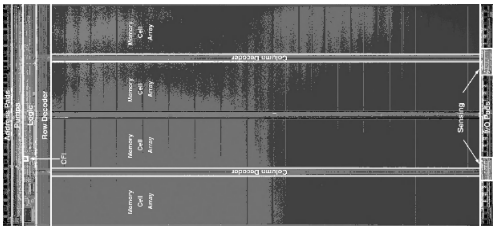

Figure 12.7. Intel StrataFlash 64-Mbit memory.

The 64-Mbit device integrated all of the knowledge gained from the two previous test vehicles and advanced beyond them with the introduction of precision internal voltage regulation and internal test capability. The 64-Mbit 2 bit/cell Intel StrataFlash memory was less than 5% larger than the 32-Mbit 1 bit/cell device on the 0.4 μm ETOX Flash memory process, delivering on the promise of 2× the bits in 1× the space and setting a new cost paradigm for Flash memory devices. A photomicrograph of the 64-Mbit Intel StrataFlash memory is shown in Figure 12.7.

12.4 VIEW OF MLC TODAY

Today, further advances in MLC capabilities have resulted in the functionality of MLC products to be indistinguishable from their single-bit-per-cell counterparts. Random byte read access times below 90 nS, V_{cc} of 1.8 V, and flexible read-while-write features on state-of-the-art lithography are the mainstream for both products [4].

12.4.1 Multilevel Cell Key Features

The concept of MLC is ideally suited to the Flash memory cell. The cell operation is governed by electron charge storage on an electrically isolated floating gate. The amount of charge stored modulates the Flash cell's transistor characteristic. MLC requires three basic elements: (1) accurate control of the amount of charge stored, or placed, on the floating gate such that multiple charge levels, or multiple bits, can be stored within each cell, an operation called placement; (2) accurate measurement of the transistor characteristics to determine which charge level, or data bit, is stored, an operation called sensing; and (3) accurate charge storage, such that the charge level, or data bit, remains intact over time, an operation called retention. These elements are achieved by exploiting stable device operation regions and by the direct cell access of the ETOX Flash memory array.

12.4.2 Flash Cell Structure and Operation

An explanation of MLC first requires a review of the Flash memory cell. Key points relevant to MLC operation are covered here. The ETOX Flash memory cell and products [5] have a long manufacturing history, having evolved in the late 1980s from EPROMs, which had been an industry standard from the early 1970s.

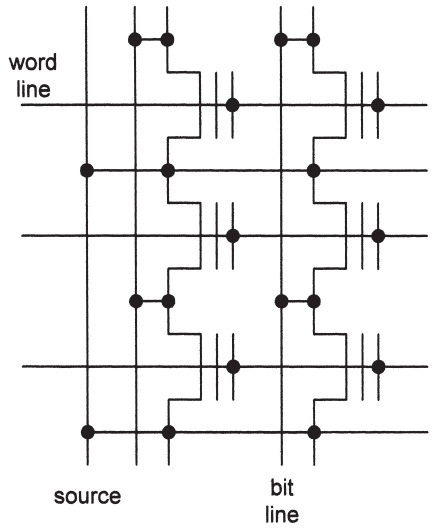

Figure 13.104. NOVORAM/FGRAM architecture.

13.8.4 NOVORAM/FGRAM Summary

Because of its high density (at comparable speed) NOVORAM/FGRAM may present a serious challenge for DRAM and SRAM even at the present technological level, since its nonvolatility is very important for many electronics applications. However, the most important feature of such memories is high scalability, limited mostly by the exponentially growing sensitivity of readout MOSFETs to fabrication spreads. The use of advanced double-gate SOI MOSFETs may push this limit to physical gate length ~5 nm. Apparently, an extension of this limit is possible using individual compensation of the threshold gate voltage of each readout MOSFET using an additional floating gate [Fig. 13.103(*b*)].

13.9 PHASE CHANGE MEMORIES

Joe E. Brewer, Greg Atwood, and Roberto Bez

13.9.1 Introduction

Phase change memory (PCM) technology is currently (2006) the subject of intense research and development by multiple companies and is a promising future alternative to Flash. In terms of cell size, die size, and cost it is approximately on a par with

Flash. It offers fast random read performance ~50 ns, fast write performance ~100 ns, and good data retention >10 years. PCM is a "direct write" technology. There is no need to erase a block of data and then program it. The technology can be made to fit into a CMOS fabrication process. It exceeds Flash in bit granularity and in endurance. In the long term it appears to have good scalability, and it has multilevel potential.

Phase change memory is certainly not a new concept. For example, the September 28, 1970, issue of *Electronics Magazine* [184] contained the die picture shown in Figure 13.105. This was a 1970 chip with a 256-bit capacity that measured 122 × 131 mils. Reset was accomplished by <200 mA at <25 V for 5 μs. Set required 5 mA at ~25 V for 10 ms. The read mode current was 2.5 mA with a supply <5 V. The history of the technology can be traced back to multiple patents filed by Stanford R. Ovshinsky beginning in the 1960s.

This primitive circa 1970 chip did not prove to be producible or competitive with the emerging floating-gate and MNOS technologies. Soon interest waned and the approach was abandoned. Later phase change technology experienced a rebirth in a new context—optical memory. Chalcogenide films were found to offer advantages when used as the memory medium for rotating disks. The rewritable phase change compact disk (CD-RW), the rewritable digital versatile disk (DVD-RW), the random-access memory digital versatile disk (DVD-RAM), and the blue-ray disk are commercially successful examples.

The optical disk in many ways is a simpler platform for investigation of the phase change phenomena. The active bit storage regions are not individually contacted. It was not necessary to first develop compatible electrode contacts and devise suitable heating elements. The medium is programmed by a high-power laser beam and interrogated by a lower power laser combined with an optical sensor. The primary issue of fast reliable and reversible phase conversion could be examined without the distraction of dealing with these other issues. The disk business provided

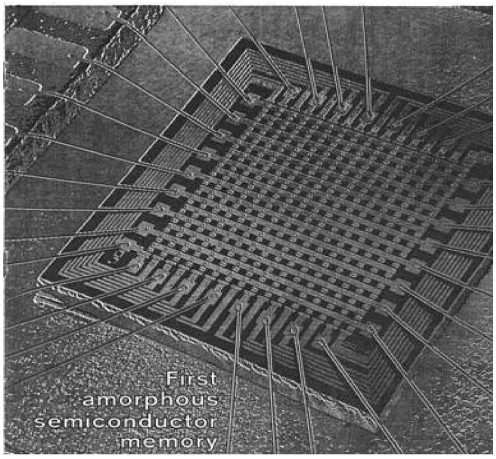

Figure 13.105. Photo titled "First Amorphous Semiconductor Memory." (From [184].)

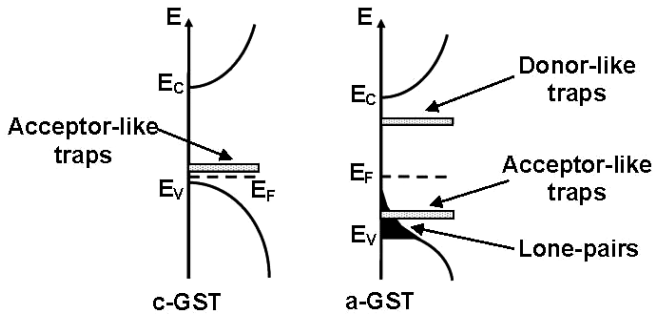

Figure 13.115. Band diagrams of c-GST and a-GST proposed by [194].

Comparatively speaking, the c-GST state offers relatively high mobility and high carrier concentration while the a-GST state has relatively low mobility and low carrier concentration. In the c-GST state lattice defects are present in concentrations as high as 15% [189]. The defects act as shallow acceptorlike traps causing the material to be a defect semiconductor that exhibits degenerate P-type conductivity with a bandlike mobility [194]. The optical bandgap of the c-GST is 0.5 eV [189].

When the c-GST is melted, the lattice and the lattice defects disappear, and the resulting structural disorder produces different kinds of electronic states. The most numerous are spatially localized C_2^0 lone-pair valence band tail states that cause a very low trap-limited hole mobility. In addition there are valence alternation pair (VAP) states believed to be caused by special defect bonding configurations between the nonbonding orbitals on the chalcogen atoms [189]. These states are the donorlike and acceptorlike traps shown in Figure 13.115. The donorlike traps are described as being C_3^+ states, and the acceptorlike traps are C_1^- states [194].

Pirovano et al. [194] describes the conduction processes for a-GST in the following way. When the voltage is low, the conduction is ohmic. As the electric field increases, impact ionization takes place, and the current rises exponentially due to secondary hole generation. Initially, most of the secondary electrons get trapped in the donorlike traps. Further increase in the voltage begins to fill up the traps and the free-electron density increases. The electron quasi-Fermi level is forced to move closer to E_C. Just beyond V_{th} impact ionization dominates over carrier recombination, the traps are filled and a voltage snap-back occurs. Now higher current can flow at a lower voltage. Impact ionization still takes place, but at a reduced multiplication rate. A much higher free-carrier density now sustains the current.

In Flash and many other memory technologies it is necessary to perform two steps to establish the state of the individual cells. Typically, an erase operation is followed by a program operation. In contrast, PCM is a direct write technology. No matter what the original state of the cell is, the final state of the cell can be established by the magnitude of current passed through the cell. For PCM it is appropriate to speak of a "read" operation and a "write" operation, but no doubt the legacy of Flash will lead to maintaining the terminology of "programming" as opposed to "write."

Figure 13.116 provides an alternative view of the writing process. Here resistance for a particular cell is plotted against the programming current. The procedure