The Reliability of Dynamic Random Access Memory Chips

by

Timothy Ward Pelton

B.Sc., University of Victoria, 1983
B.Ed., University of Victoria, 1991

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

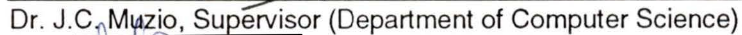
in the Department of Computer Science

ACCEPTED
FACULTY OF GRADUATE STUDIES

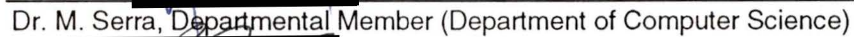
DEAN

DATE 21 Dec '93

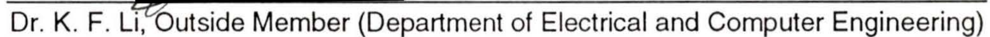
We accept this thesis as conforming
to the required standard



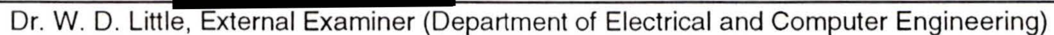
Dr. J.C. Muzio, Supervisor (Department of Computer Science)



Dr. M. Serra, Departmental Member (Department of Computer Science)



Dr. K. F. Li, Outside Member (Department of Electrical and Computer Engineering)



Dr. W. D. Little, External Examiner (Department of Electrical and Computer Engineering)

© TIMOTHY WARD PELTON, 1993

University of Victoria

All rights reserved. Thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

Supervisor: Dr. J.C. Muzio

ABSTRACT

Advances in dynamic random access memory (DRAM) chip capacity have required designers to develop and incorporate many new techniques to ensure viable yields and to maintain adequate reliability. One technique used is the incorporation of error correction code circuitry which increases the yield of chips that are functionally fault free, and reduces the probability of failure by hiding some of the errors caused by new faults.

The majority of DRAM chip failures that occur during normal operations are attributed to transient faults (primarily caused by α-particles). We develop a method for analyzing the reliability of DRAM chips during normal operation and suggest modifications to the chip circuitry to reduce the effect of transient faults.

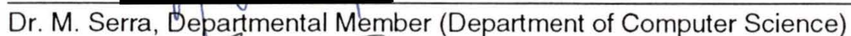
We propose two enhancements which make more extensive use of the built in error correction ability of current DRAM chips. The first scheme is described as parallel scrubbing. Data contained in the chip is checked and corrected (if necessary and possible) in parallel with normal access cycles. The second scheme improves on the first, but has a higher overhead cost. It involves implementing systematic parallel scrubbing using counters to provide addresses to idle areas of memory thereby ensuring that the data in the chip is scrubbed uniformly.

In order to assess the efficacy of our two schemes we have developed a model for estimating the reliability of DRAM chips using Markov and series models. Our model determines the probability that an uncorrectable error exists on the chip over time and accounts for the soft error rate, the distribution of accesses, the access rate, and the number of permanent faults present on the chip. Our model can be adapted to analyze the reliability of any standard DRAM chip.

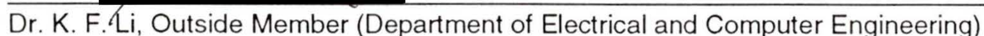
In our analysis we find that our model is helpful in identifying the changing probabilities and trends. We find that a DRAM chip using the first scheme, parallel scrubbing, is almost 31 times less likely to develop an uncorrectable error than the standard DRAM chip, and that a chip using the second scheme, systematic scrubbing, is more than 240,000 times less likely to develop an uncorrectable error. This leads us to the conclusion that our schemes offer a significant potential for improvement to the reliability of DRAM chips.

Examiners:

Dr. J.C. Muzio, Supervisor (Department of Computer Science)



Dr. M. Serra, Departmental Member (Department of Computer Science)



Dr. K. F. Li, Outside Member (Department of Electrical and Computer Engineering)

Dr. W. D. Little, External Examiner (Department of Electrical and Computer Engineering)

Table of Contents

Table of Contents iv

List of Figures viii

Acknowledgements ix

Dedication x

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Goals 2

1.3 Outline 3

Chapter 2 Background 5

2.1 Memory Architecture 5
2.1.1 The Memory Cell Array 6
2.1.2 DRAM Chip Architecture 8
2.1.3 Memory System Architecture 11

2.2 Faults, Errors and Failures 13
2.2.1 Permanent Faults 14
2.2.2 Intermittent Faults 15
2.2.3 Transient Faults 15
2.2.4 The Effects of Transient Faults in DRAM 16

2.3 Fault Tolerance 18
2.3.1 Fault Tolerant Techniques in RAM 18
2.3.2 Error Correction Codes 20
2.3.3 Fault Tolerant RAM Chips 22

2.4 Summary 23

Chapter 3 Tools for Reliability Analysis. 24

3.1 Probability and Reliability 24
3.1.1 General Probability 24
3.1.2 Probability Relations. 25
3.1.3 Stochastic Processes 27
3.1.4 General Reliability 27

3.2 Failure Mode and Rate 28
3.2.1 Failure Mode 29
3.2.2 Estimating the Failure Rate. 31

3.3 Reliability Models for Analysis 33
3.3.1 Combinatorial Modeling. 33
3.3.2 Markov Modeling 39

3.4 Summary. 43

Chapter 4 Reliability of DRAM Chips 44

4.1 Clarification and Justification 45
4.1.1 Unreliability in a DRAM Chip 45
4.1.2 Justification for Consideration 45

4.2 Basic DRAM Chip Parameters 47
4.2.1 Chip Parameters Defined 47
4.2.2 A Typical DRAM Chip 47
4.2.3 Parameter Values Assigned 48

4.3 Basic Reliability Relationships in DRAM Chips 48
4.3.1 Reliability of a Bit 48
4.3.2 Reliability of an ECC Block 49
4.3.3 Reliability of a DRAM Chip 49

4.4 DRAM Chips Containing Permanent Faults 51
4.4.1 Permanent Faults and their Distribution 51
4.4.2 An ECC Block with a Permanent Fault 51
4.4.3 A DRAM Chip with Permanent Faults 52

4.5 Error Correction 53
4.5.1 Scrubbing 53
4.5.2 Distribution of Scrubbing 55
4.5.3 Effect of Scrubbing on Reliability 57

4.6 Other Factors 58
4.6.1 Internal Factors 59
4.6.2 External Factors 59

4.7 Summary 60

Chapter 5 Scrubbing Schemes 62

5.1 Scrubbing in DRAM Chips 62

5.2 Three Schemes for Scrubbing DRAM Chips 64
5.2.1 Scheme 0 - Standard 16 Mbit Chip with ECC 64
5.2.2 Scheme 1 - Parallel Scrubbing 64
5.2.3 Scheme 2 - Systematic Scrubbing 67
5.2.4 Cost of Implementation of Schemes 68

5.3 Parameters for the Scrubbing Schemes 69
5.3.1 Scheme 0 Parameters 69
5.3.2 Scheme 1 Parameters 69
5.3.3 Scheme 2 Parameters 70

5.4 Markov Models for the ECC Blocks 70
5.4.1 Our Basic Markov Models 71
5.4.2 The Transition Probability Matrices 73
5.4.3 ECC Blocks in the NE state 74
5.4.4 ECC Blocks in the SE state 75
5.4.5 ECC Blocks in the UE state 77
5.4.6 Transition Matrices for Working ECC Blocks 77
5.4.7 Transition Matrix for a faulty ECC Block 79

5.5 Summary 80

Chapter 6 Analysis and Results 81

6.1 Basis of Analysis 82
6.1.1 Software used for Our Analysis 82
6.1.2 Time, the Dependent Variable 82
6.1.3 Calculating the State Probabilities 83
6.1.4 Method of Analysis 84
6.1.5 Terms and Definitions 85

6.2 Analysis 85
6.2.1 Analysis of an ECC Block in the NE State. 85
6.2.2 Analysis of an ECC Block in the UE State 86
6.2.3 Analysis of an ECC Block in the SE State 89

6.3 Unreliability of ECC Blocks 90

6.4 Analysis of the Reliability of DRAM Chips 93
6.4.1 Chips Containing Permanent Faults 93
6.4.2 Chips with No Permanent Faults 95
6.4.3 Differences in Reliability 96

6.5 Summary 97

Chapter 7 Conclusions **99**

7.1 Overview of Development 99

7.2 Summary of Results 102

7.3 Future Work 103

7.4 Concluding Remarks 105

Bibliography **106**

List of Figures

Figure 2.1 Schematic of part of a memory cell array. 7
Figure 2.2 Functional model of a DRAM chip. 8
Figure 2.3 Tree organization of memory islands. 9
Figure 2.4 A 4 Mbit island, with ECC block interleaving. 11
Figure 2.5 Sample configuration of a 4 MB RAM system. 13
Figure 3.1 An illustration of the failure rate of DRAM chips. 29
Figure 3.2 A system block diagram for a series model. 34
Figure 3.3 A system block diagram for a parallel model. 35
Figure 3.4 A system block diagram for a complex system. 37
Figure 3.5 A Markov model for a 4 bit SEC word. 40
Figure 4.1 Illustration of access rate by memory address. 56
Figure 4.2 Simplified illustration of access rate in memory 56
Figure 5.1 Illustration of ECC block access rate using scheme 1. 65
Figure 5.2 Simplified illustration of access rate using scheme 1. 66
Figure 5.3 Simplified illustration of access rate using scheme 2. 67
Figure 5.4 Markov model for a fault free ECC block 72
Figure 5.5 Markov model for a SEC block with a permanent fault 73
Figure 6.1 Probability of an UE in an ECC block 87
Figure 6.2 Probability of a SE in ignored ECC blocks 89
Figure 6.3 Probability of an UE for each ECC block type. 91
Figure 6.4 Probability of an UE in a DRAM chip 94
Figure 6.5 P(UE) in a chip containing zero permanent faults 95

Acknowledgements

I wish to thank Dr. J. C. Muzio for both his guidance in the development and preparation of this thesis and for his financial support. I acknowledge the contribution of many faculty and students to my growth in understanding of this subject. Specifically, I thank Dr. Muzio, Dr. M. Serra, Dr K. Li, and Dr. V. King for courses which they taught to me.

I thank my family for their encouragement and support throughout my life, especially for helping me to know that I can do anything I want to do if I am willing to put in the effort.

I am thankful for the firm foundation the gospel provides in my life and acknowledge the love that Heavenly Father has for me.

Finally, I wish to thank my dear wife Leslee, for her example, her support, and her love.

To Leslee

Chapter 1 Introduction

1.1 Motivation

Advances in memory technology have been both necessary and instrumental in the development of computer systems. From the 1 Kbit chip of 1970, to the 64 Mbit chip of 1993, dynamic random access memory (DRAM) chips have been keeping pace with the demands of the market in capacity, reliability and cost. As new limitations to future improvements appear poised to slow the progress, new techniques and technologies are developed which allow this phenomenal development rate to continue [Int86b, TP89].

One of the techniques in general use by manufacturers to improve yields and reliability of memory chips involves the implementation of error correction code (ECC) circuitry in the DRAM chip. Many of the chips produced today contain permanent faults (flaws) that cause errors to appear in the chip. These errors are corrected by the error correction circuitry when the data is accessed, effectively hiding (or masking) the effects of the faults. The number of faults that can be tolerated on a chip is related to how the error correction is accomplished and the distribution of the faults. When manufacturers test the chips they only need to determine whether the faults that are present exceed this error correction ability.

There are three types of faults that cause errors: permanent, intermittent, and transient. Manufacturers can identify most permanent and some intermittent faults during testing, and from these results they determine whether a chip is acceptable. Many papers analyze and discuss the effectiveness of error correction with respect to permanent faults and yield [Aic84, FS91, Kal90, Sta92]. However, very few attempt to estimate the reliability of DRAM chips in the presence of transient faults as they occur over time.

It is estimated that 98% of failures that occur in DRAM chips during normal operation can be attributed to soft errors caused by transient faults [Maz88, Sai82, vdG91]. Transient faults occur randomly over time and the resulting errors must be handled by the chip or the system concurrently with normal operation. A built in error correction ability can remove some of the errors caused by transient or intermittent faults thereby increasing the reliability of the DRAM chips. However, estimation of the reliability of DRAM chips is difficult because of the complexity added by the use of error correction schemes and the reluctance of manufacturers to publish error rates.

1.2 Goals

In this thesis we develop a model that allows us to estimate the reliability of a DRAM chip that has an error correction ability. This estimate is found as the complement of the probability that errors will exceed the capacity of the error correction scheme over time. Specifically, our model takes into consideration the soft error rate, distribution of accesses, the access rate and the number of permanent faults present on the chip.

We propose two enhancements to the standard error correction scheme that could be incorporated in DRAM chips at a relatively low cost. The reliabilities of the standard scheme and our two schemes are estimated by using our model and the results are compared.

1.3 Outline

In chapter 2 the architectures of a DRAM cell, a DRAM chip and a memory system are described. We discuss faults, errors and failures and how they occur in DRAM chips with a special emphasis on the cause and effect of transient faults. Finally the techniques used to improve fault tolerance in DRAM chips are covered, and we focus on the use of error correction schemes.

Chapter 3 develops the mathematics behind our model by reviewing probability relations and defining reliability. The causes of failures in DRAM chips are considered and a realistic failure rate is estimated for use in our reliability analysis. We conclude this chapter by describing different approaches that may be used to analyze the reliability of complex systems, and identifying the methods that are most appropriate for use in our model.

Chapter 4 clarifies and defines terms that we use in our analysis. We describe the standard DRAM chip that we are using to evaluate the efficacy of our model and our schemes. We then explain the effects of permanent faults, scrubbing, and access distribution on the reliability of the chip. There are two distinct parts to the model, the reliability of individual blocks of data, and the reliability of the chip based on types of data blocks that are contained within it.

In chapter 5 we describe the standard scrubbing scheme and our two enhanced schemes along with their associated parameters. From these parameters we build models to describe the reliability of each of the types of data blocks that we are concerned with.

Chapter 6 describes the platform which we use for our analysis along with our method. The analysis of each type of data block is followed by discussions explaining the effects observed. The reliability of our chip is assessed when different numbers of permanent faults are present. And finally, the reliability estimates for the DRAM chips incorporating each of the three error correction schemes are generated and compared.

Chapter 7 concludes our thesis by summarizing our results and suggesting areas for further research.

Chapter 2 Background

The purpose of this chapter is to review the architecture of random access memory (RAM), the faults that occur in memory cells, and the techniques used to tolerate these faults. This provides the basic background for our discussion of fault tolerance in RAM chips and the potential for improvement.

2.1 Memory Architecture

There are two types of random access memory, namely, static (SRAM) and dynamic (DRAM). Our discussion and analysis focus on the dynamic RAM devices because they are the most prevalent and have the highest capacities [vdG91]. However, many of the points that are covered also apply to static memory devices. In this section we discuss the architecture of memory cell arrays, RAM chips, and memory systems.

2.1.1 The Memory Cell Array.

Dynamic random access memory (DRAM) chips use the simplest form of memory cell which is constructed with a single transistor and a capacitor, known as the 1-T cell. A bit of information is stored in the capacitor and access is controlled by the transistor.

Recently developed multimegabit DRAM chips use 3D capacitors (either trench or stacked), in place of planar capacitors, to reduce the required silicon area for each cell while keeping the cell capacitances at levels that limit susceptibility to faults [TP89]. The transistors have also been scaled down and use a P-channel in order to minimize the potential for disturbing neighboring cells [Int86a].

Cells are accessed in groups called words (these are different from memory system words). Access is accomplished by sending an electrical pulse down an array word line . The pulse closes the transistors connecting each of the bit lines to their respective capacitors giving access to the individual charges (see figure 2.1).

During a read operation, a set of bit lines are precharged (to half the supply voltage), the transistors are closed, and the charges in the capacitors cause the bit line voltages to swing. These small swings are amplified by the sense amplifiers to voltage values that represent logic 1's and 0's [Ito90].

For a write operation, the set of bit lines are precharged, the transistors are closed by a pulse on the word line, and the charges are saved in the capacitors [Ito90]. It is interesting to note that in some chips the charges in half of the capacitors are mapped to a logical 1 and the other half to a logical 0 [Ito90, Coc93].

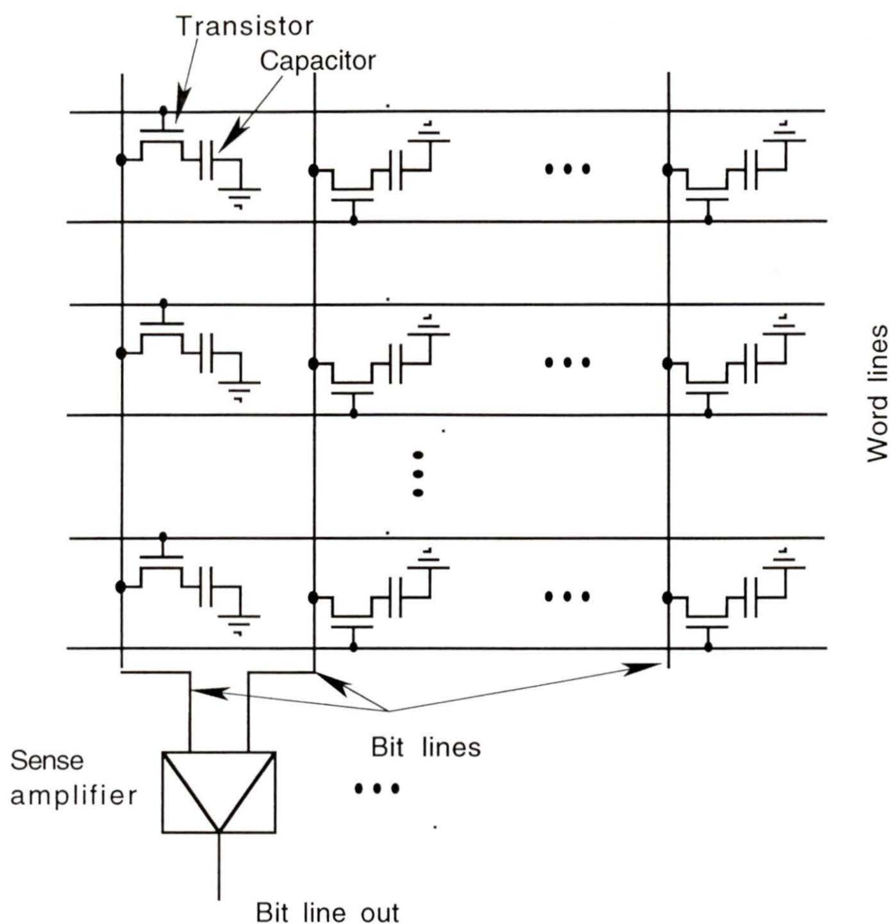


Figure 2.1 Schematic of part of a memory cell array.

This type of memory is known as dynamic RAM because the charges in the capacitors leak out over time requiring the cells to be regularly refreshed in order to maintain the information contained in them. The refresh activity consists of sending a pulse down a word line, causing each bit of the word to be sensed and amplified, and then feeding the amplified bit values back into their corresponding cells [Ito90]. The memory chip must cycle through each word line within a minimum time period (called the refresh cycle) in order to keep the charges that are held in the capacitors above threshold values. One typical 4Mbit DRAM chip has a 16 ms refresh cycle time [TP89].

2.1.2 DRAM Chip Architecture

As shown in figure 2.2, the basic functional parts of a memory chip include an address decoder circuit, a memory cell array, refresh circuit, and the read-write circuit. This thesis is primarily concerned with the reliability of the memory cell array, and how it can be improved by modifying some of the peripheral circuitry (the address decoder and the read-write logic).

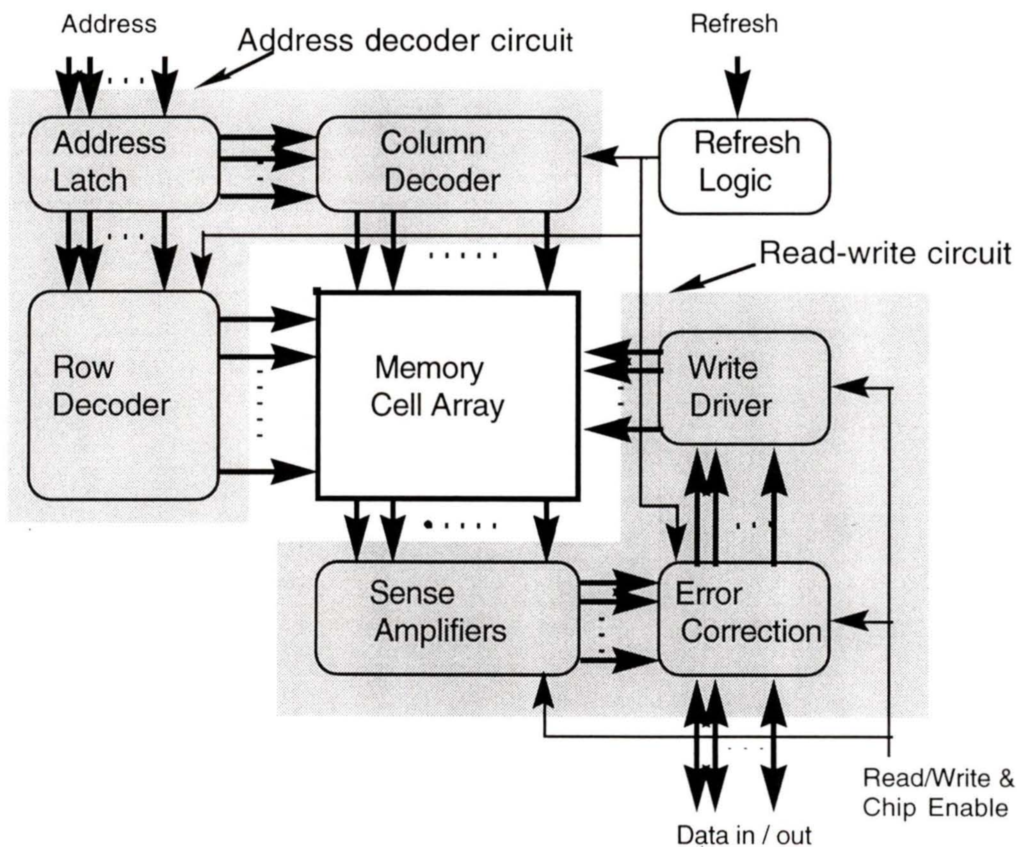


Figure 2.2 Functional model of a DRAM chip.

Because of limitations in the sense amplifiers and the word line and bit line capacitances, the chip is actually composed of multiple memory modules, called memory islands and some shared circuitry. Typically the shared circuitry includes data registers, refresh logic and read-write control, while each island contains its own address decoder and error correction circuitry [Ari90].

In the new 4, 16, and 64 Mbit chips, islands are typically 256Kbits,1Mbits or 4Mbits [SL92, Pra88, Ari90]. The addressing circuitry accessing these islands may be in the form of another array or a tree [Pra88]. Implementation of a second array or a tree structure (as shown in figure 2.3) to organize the memory islands, has allowed for parallel operations to be incorporated into chips. This parallelism improves the speed of testing, enhances performance in block accesses and increases the production yields [Pra88].

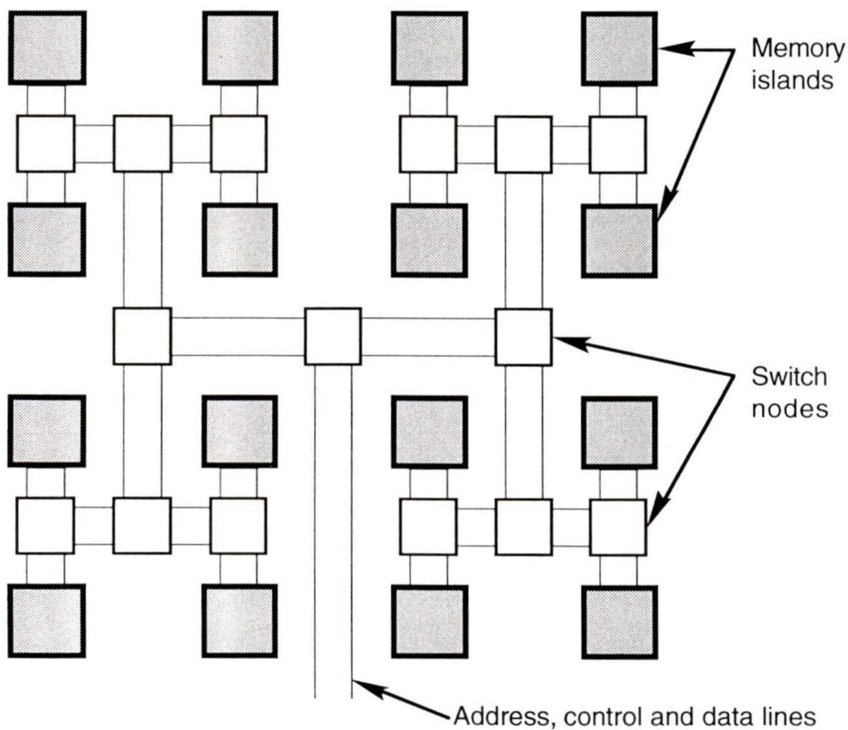


Figure 2.3 Tree organization of memory islands.

As capacities of DRAM chips increase, the potential number of faults occurring on the chip also increases. For multimegabit DRAM chips it is necessary to incorporate redundancy and on-chip error correction in order to realize viable production yields [FS91a, FS91b, Maz88]. The error correction ability is located within each memory island and operates independently and transparently [Ari90, Kal90].

A memory island consists of a rectangular array of memory cells along with some peripheral circuitry. To minimize the size and complexity of the address decoders, the islands have dimensions that have the same order of magnitude. In one new DRAM chip which uses 4 Mbit islands there are 1024 word lines each with 4096 data bit lines and 288 check bit lines (see figure 2.4) [FS91b].

Figure 2.4 shows how a word line may be composed of error correction code (ECC) blocks (a, b, c, d, ...) that are interleaved (see section 2.3.1). A read operation causes all bits on a word line to be accessed, sensed and amplified. From this large set of bits an ECC block is selected, checked and corrected if necessary, and the bits that were addressed are selected and output. To write, most of a read operation is performed, the new bits are written into the data of the checked (and possibly corrected) ECC block, the error correction code is updated, the ECC block is written back into the word line, and the word line is stored in the memory cell array.

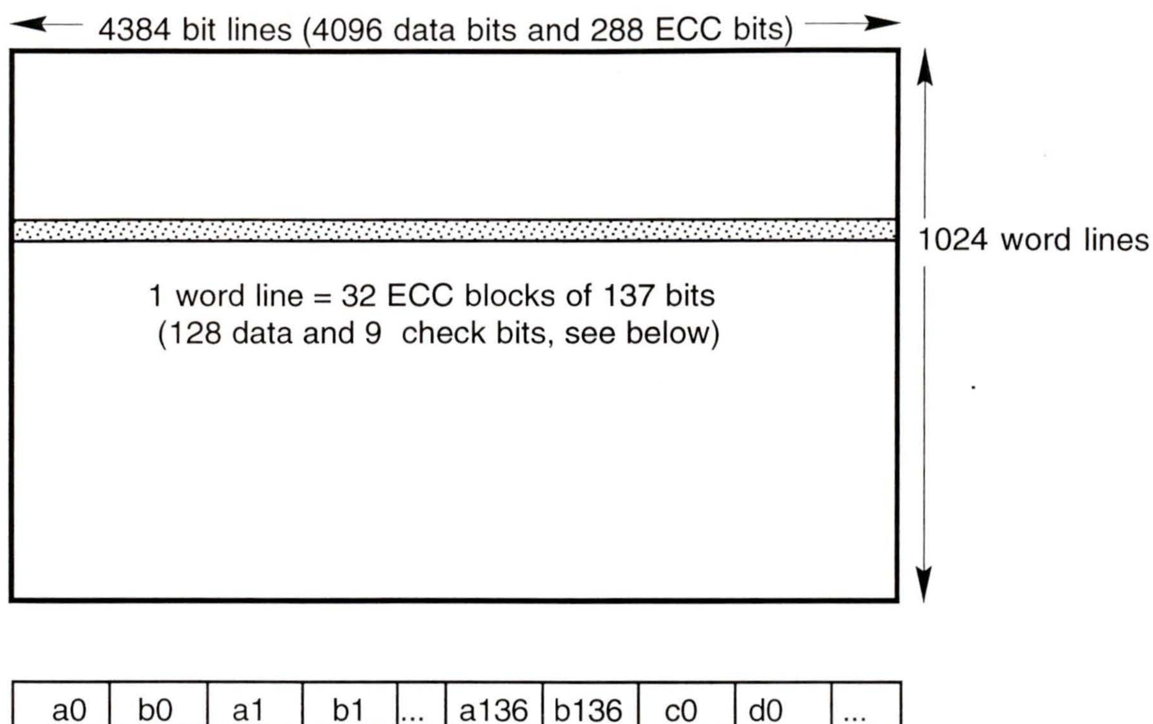


Figure 2.4   A4 Mbit island, with ECC block interleaving.

For the purposes of this thesis we are considering multimegabit DRAM chips that have been designed to allow parallel testing, and include error correction circuitry in each memory island. With these assumptions the additional requirements to implement our new schemes are limited to control circuitry.

2.1.3 Memory System Architecture

The most common forms of memory systems use one or more bits from each of a set of chips to hold a system word [RK91]. The term *symbol* refers to this set of 1, 2, 4, or 8 bits that is stored in a single chip. A system word is accessed by sending the same address to each chip in the set and accessing a symbol from each of the chips in parallel.

To implement a larger memory, a 2D array of chips is used. A few bits are used to select the appropriate set (row) of memory chips and the remaining bits are used to address a symbol from each of the chips in the selected row and the symbols are combined to form a word.

By splitting up the system word among different chips there is a much smaller probability of having more than a single symbol error at the memory system level. With the addition of symbol error correction at the chip level, many memory systems now have two independent levels of fault tolerance to improve their reliability [YF92].

Figure 2.5 shows how a 4MB RAM system might be implemented using 1Mbit DRAM chips organized into 4 rows of 12 chips with a 22 bit address. Two bits are used to select the row or set of chips on the board, and the remaining 20 bits are used by each of the selected chips to access 1 bit. This set of 12 bits, one from each chip, includes four bits which are used to provide a single error correcting code (SEC) capability at the system level. Note that this system level error correction is in addition to, and independent of, any error correction occurring at the chip level.

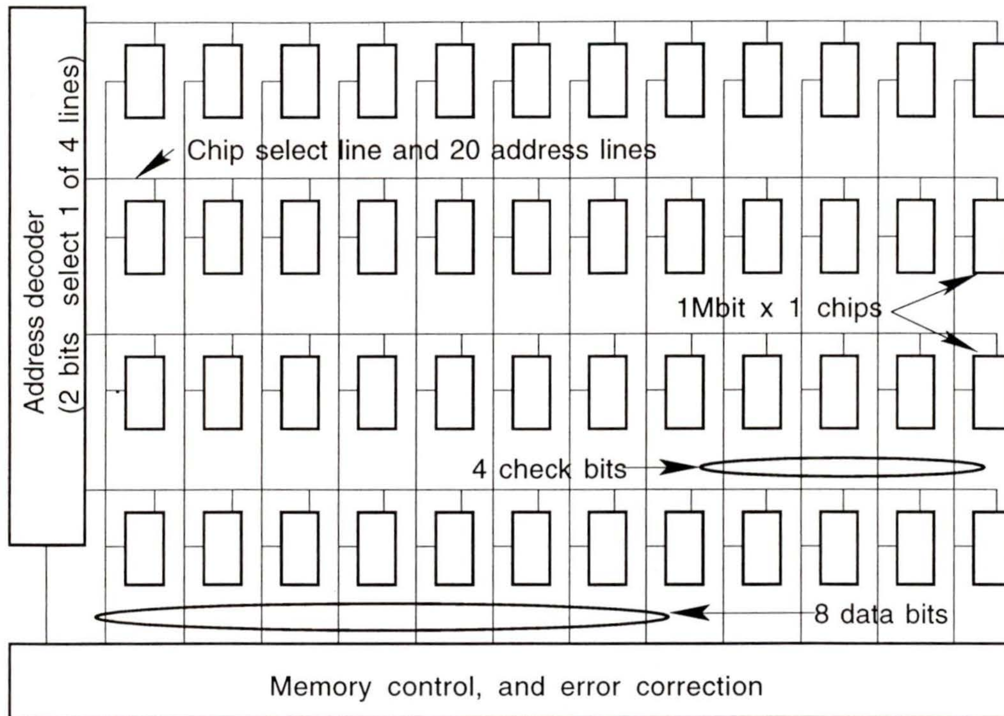


Figure 2.5 Sample configuration of a 4 MB RAM system.

We are not selecting any particular memory system model for this thesis but note that improvements in reliability at the chip level positively affect the reliability of all memory systems in which they are incorporated.

2.2 Faults, Errors and Failures

The reliability of a DRAM chip is directly related to the occurrence of faults. Faults may cause errors and errors may cause failures. A fault is defined as a defect or unexpected occurrence within the chip. An error in a DRAM chip is seen as a difference between the state of the chip (bit values) and the expected state. A failure occurs when the chip returns an incorrect value. When an error exists in the chip that is not correctable by any on-chip error correction, it is called an uncorrectable error (UE).

For our discussion, we consider the existence of an UE to be a chip failure because it has the potential to cause a failure, and it remains as long as the chip is active. Thus, we determine the reliability of a chip by finding the complement of the probability of an UE over time (see chapter 3 for the discussion of probability and reliability). The reliability of a DRAM chip defined in this way provides a convenient base for comparison of different chips and of on-chip error correction schemes and is consistent with other publications [vdG91,CH84].

2.2.1 Permanent Faults

Permanent faults, or hard faults, have many different causes including: environmental conditions, physical defects on the chip, and functional design errors. They are the easiest faults to model and detect because once they occur they remain and behave in a consistent manner [vdG91]. Some examples of permanent or hard faults are:

poor or missing connections,

shorts - caused by defects on the silicon or corrosion,

erroneous functionality - due to a poor design.

Most permanent faults are discovered during initial testing and either remedied or the chip discarded [Int86a]. Field studies have shown that 2%-13% of system failures are caused by permanent faults that either were not detected during the initial screening or, developed due to environmental conditions or wear out [SS82]. In world class memory production facilities, the ratio of permanent faults in DRAM chips is approximately 1 in 1,000,000 bits [SL92].

2.2.2 Intermittent Faults

Intermittent faults are caused by either some form of physical defect or irregularity that still allows the chip to run correctly most of the time (eg: loose connections, resistance and capacitance variations), or some design flaw that causes an effect when a certain sequence of events occurs (critical timing, load). They are more difficult to model and detect, although some intermittent faults can be forced to permanent during testing by stressing the circuit using temperature, pressure, humidity, and power-level variations.

Intermittent faults are expected to recur and last for a finite period of time but the time between occurrences can be significant and dependent on workload. Thus they are not always present during the manufacturing test and must be detected by system diagnostics or handled by error detection and correction circuits at the chip or system level [vdG91].

2.2.3 Transient Faults

Transient faults are caused by environmental factors acting on the chip. These factors include radiation, pollution, temperature, pressure, electromagnetic fields, static discharges, etc. [Sie91, vdG91].

Transient faults are difficult to model in general because their occurrence and effect is uncertain and they are not expected to recur in any particular pattern. A transient fault may cause an erroneous result but does not cause any significant physical change in the chip.

The demand for chips with higher capacities and increased complexities has led to an increased sensitivity of chips to these types of faults. Recent studies indicate that more than 80% of digital system failures are caused by transient faults [vdG91, YS92].

α-particles are a significant source of transient faults in VLSI chips [FS91, SL92]. An α-particle is a helium nuclei that has been emitted from a radioactive substance. In VLSI chips, α-particles causing transient faults are generally emmitted from contaminants within the device (thorium, uranium etc.)[Amer87]. When an α-particle hits a circuit it creates hole-electron pairs as it decelerates and comes to rest in the silicon substrate. The charges generated by this activity are of the same order of magnitude as the charges transferred in a VLSI chip and thus they have the potential to cause erroneous results [Int86a, Lef93].

2.2.4 The Effects of Transient Faults in DRAM

In a well designed DRAM chip it is estimated that over 98% of failures that occur during normal operation are caused by transient faults in the form of α-particle hits [Maz88, Sai82]. It is the nature of these faults that we focus on in this section to build a foundation for our calculations on reliability.

The effects of α-particles in RAM were not of a great concern when cell capacitances were much larger and the refresh cycle could top up the capacitors affected. However, by 1977, α-particle upsets were beginning to become a significant concern as smaller cell capacitances increased the susceptibility to errors caused by these particles [Int86b]. In current DRAM chips an α-particle hit generally causes 1 or 2 cells to be disturbed [Maz88].

As cell capacitances are further reduced, β-particles (electrons) will also become an important factor because there are more of them and they have longer paths and thus have a potential to travel through more cells. The action of β-particles is similar to α-particles but the charge associated with them is much smaller [SRK87, Lef93].

Although the effects of transient faults are hard to model in general; their effects are relatively consistent in DRAM chips. When a particle hits the capacitor the charge stored (if there is one) is reduced or drained. When it hits a data line the voltage drops and a false value is sensed and written back to the cell (if the timing happens to be right) [UMSM88]. Thus, in a DRAM chip the primary effect of an α-particle hit is to change the value stored in a cell.

VLSI devices are coated with a protective film to limit effects of radiation from external sources and from contaminants in the outer packaging. However, some active form of protection, primarily concurrent error correction, must be provided to manage the radiation emitted from the intrinsic contaminants in the silicon, aluminum and doping materials present in the die itself [vdG91].

2.3 Fault Tolerance

As systems become more complex, and the demands on them increase, the potential for failure increases, and the reliability goes down to levels that are not acceptable to the user. In order to improve the reliability of a system, engineers aim to reduce or mask (hide their effect) faults. First, they work toward reducing the potential for failures by improving quality control and spending more effort on design. And second, they incorporate into the systems an ability to correct or mask some of the errors that occur so that the errors caused by the faults are hidden from the system's view. This second phase is referred to as fault tolerance and has become one of the primary tools used for improving the reliability of systems [Joh89].

Implementation of fault tolerance has significant costs attached to it in area, resources and performance. With computer memories, there is a tradeoff between keeping the data clean (or error free) and keeping the access time low.

2.3.1 Fault Tolerant Techniques in RAM

Some of the techniques already used to improve the reliability in memory chips include error correcting codes (ECC's), bit scattering, sparing, consecutive correction, prestorage protection, and extended error correction. Before discussing the application of ECC's we review the other techniques for improved reliability.

The first, bit scattering, is used to reduce the probability of a multiple bit error in a memory word (or symbol). It involves interleaving the data bits of an ECC block on an array word line [Aic84]. Figure 2.4, on page 11, illustrates this almost zero-cost technique which effectively separates multiple faults into single faults (within an ECC block) that are easily handled by error correction schemes.

Sparing involves the use of redundant rows, columns or chips by the memory system to replace faulty memory segments. The effect on the reliability is significant but there is a cost involved at each of the detection, location and recovery levels. Some memory systems are smart enough to detect unreliable chips, replace them with a spare and then rebuild the information in the spare chip [Sie91, RK91].

Consecutive correction operates by maintaining a history of the permanent faults that have been detected. Thus, when a normally uncorrectable error occurs, the last known error is corrected and the word rechecked to see if the remaining error is a correctable one. The cost involved is substantial in terms of both the time to correct the multiple errors and the overhead to keep the failure history, but it is viable for some systems where data integrity is critical [Aic84].

Prestorage protection schemes check the value of the data stored and if a permanent fault affects one of the bits then the data is complemented and stored again to hide the fault. When the data is read the complementation is detected and reversed. The cost for this scheme is incurred mostly during write accesses (25-33% of memory accesses) [MW88].

Extended error correction allows systems that have a single error correction ability to recover from a double error consisting of at least one hard error. When a double error is detected the complement of the data is written to the memory cells, the word is reread, complemented and compared to the first data read. This scheme allows the correction of any hard errors [vdG91]. A single soft error can then be corrected by the primary error correction capability. Here the cost is incurred only when an uncorrectable error is encountered during a read access.

2.3.2 Error Correction Codes

Error correction codes, ECC's, have been used for many years at the memory system level (eg: IBM System/3) to improve fault tolerance as well as at the chip level to deal with transient faults (soft errors) [SL92]. It is now becoming common for multimegabit DRAM chips to incorporate ECC based fault tolerance in the chip to mask permanent faults (to hide them from view) [FS91b] effectively increasing the yield of functionally fault free chips. Typically DRAM chip designers employ single error correction/double error detection (SEC/DED) codes that correct all single bit errors as well as detecting many multibit errors that occur in an error correction code block [CH84].

The ECC schemes used in DRAM today are based on separable codes (meaning the data bits are distinct from the check bits), primarily Hamming codes (odd weight column codes) [CH84, Ari90, SL92]. An ECC block is a fixed length set of bits where most of the bits are data bits and the remainder are check bits.

One common SEC/DED code is (22,16) - where there are 22 bits in the block with 16 of them being data bits, the other 6 being check bits [FS91b]. Other common codes include: (72,64) and (137,128) [Aic84,FS91a].

In error correction codes each check bit corresponds to a parity bit for a unique subset of the data bits. When the data is written to the DRAM chip, a set of check bits are generated and combined with the data into an ECC block. When the data is read, a new set of check bits is generated and compared to the old. With a single error correcting, double error detecting (SEC/DED) code, any difference between the two sets of check bits either uniquely describes the bit that needs to be corrected or indicates that 2 or more errors (an uncorrectable error) have occurred.

At the memory system level, ECCs that can correct or detect 2, 4 or 8 bit errors (single symbol error correction/double symbol error detection, SSEC/DSED) are being proposed and used [Che92]. This further enhances the reliability of the system by providing error correction ability for a symbol (part of a system word) from a bad chip. Combinations of error detection and correction can be used to enhance the system reliability to levels that are necessary for critical applications.

Error correction and detection using codes has two primary costs associated with it. The first is the overhead cost of the code bits which is inversely proportional ((log x)/x) to the length of the code. In a (22/16) code 6 bits are redundant (i.e. there is a 37.5% overhead).

The second cost is the time needed to encode the data when writing to memory and encode, compare and correct the data when reading. This cost is proportional the length of the code (log x). Thus there is a tradeoff to be made between speed (of encoding) and area overhead (redundant bits).

2.3.3 Fault Tolerant RAM Chips

Manufacturers have implemented SEC/DED on most new RAM chips [FS91b] both to increase their effective yields by masking permanent faults and to improve reliability by reducing the effect of soft errors caused by transient faults. A typical 16 Mbit DRAM chip may have only a few permanent faults left for the ECC to handle when it is shipped [SL92]. Over time, errors caused by transient faults may build up in the cells and when two errors 'line up' in one ECC block the information in the chip contains an uncorrectable error (UE). Whenever a piece of data is accessed, the ECC block containing it is checked and corrected if necessary and possible. Thus, any single transient faults are scrubbed out of the ECC block and the probability of two transient faults lining up in any ECC block is reduced.

2.4 Summary

In this chapter we review the architecture of DRAM chips and discuss how data is inserted, accessed and maintained. Faults are classified as permanent, intermittent, and transient, and we focus on the causes and effects of transient faults in DRAM chips caused by α-particles. Finally we consider some of the different forms of fault tolerance that may be incorporated into memory chips and systems with an emphasis on ECCs.

Manufacturers could publish a defect level and indicate the probability of hard or permanent errors and calculate a soft error rate (SER) to indicate the expected level of soft errors (intermittent or transient) that will occur during operation. [Int86a, SS82]. However, the industry generally keeps this information private, and most published information gives only relative improvements and normalized comparisons [SL92].

In RAM chips, most of the permanent, and some of the intermittent, faults are detected and handled by the manufacturer. Transient faults, primarily caused by α-particles, cause 98% of the errors occurring during the normal operation of a DRAM chip in the real world [Maz88, Sai82, vdG91]. For our thesis we assume the distribution of the transient and intermittent faults to be random and to affect only a single cell in a block (see section 2.3.1).

In this thesis we analyze the reliability of RAM chips that use ECCs and determine what gains in reliability are possible if the ECC circuitry available in each memory island is utilized more efficiently. The next chapter provides us with a basis in probability, reliability and reliability modeling on which to develop our analysis.

Chapter 3 Tools for Reliability Analysis.

This chapter outlines the mathematical methods used in our analysis of the reliability of DRAM chips. We begin by reviewing basic probability, and reliability relationships. The next section describes the specific assumptions regarding failure mode and rate which we use in our analysis and the corresponding relationships that follow. Finally we describe the models that are used for determining the reliability of more complex systems including the models we use for analyzing the reliability of DRAM chips.

3.1 Probability and Reliability

3.1.1 General Probability

Probability is the term used to refer to the ratio of the expected number of occurrences of a particular event, divided by the number of possible outcomes. For example, let the particular event be called A, and let P(A) be the probability that the event A occurs. Then:

$$P(A) = \frac{\text{expected number of occurances of A}}{\text{total number of all possible outcomes}}.$$

If the particular event never occurs, the probability is 0 and if it always occurs, the probability is 1. For probability values which are very close to 1 we use the notation 0.9_65 to represent 0.9999995.

3.1.2 Probability Relations.

The complement of A is written as A' and the probability can be given as:

$$P(A') = 1 - P(A).$$

For two events A and B, the probability that both A and B occur is given as:

$$P(A \text{ and } B) = P(A)\,P(B|A) \quad (\text{also} = P(B)\,P(A|B)).$$

This statement indicates that the probability of both event A and event B occurring, P(A and B), is the product of the probability of event A occurring, P(A), and the probability of event B occurring given that event A occurs, P(B|A). When the occurrence of one event does not change the probability of the other the events are described as independent and P(B|A) = P(B) giving us:

$$P(A \text{ and } B) = P(A)\,P(B) \qquad (\text{with A, B independent}).$$

If A and B cannot both occur, the events are called mutually exclusive, and we have:

$$P(A \text{ and } B) = 0 \qquad (\text{with A, B mutually exclusive})$$

The probability that either event A or event B or both occur is expressed generally as:

$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$

This expression states that the probability of either event A or event B occurring is equal to the sum of the individual probabilities of each of the events occurring less the probability of both of the events occurring (because this is counted twice as part of the first two terms).

More specifically we also have:

$$P(A \text{ or } B) = P(A) + P(B) \qquad \text{(with A, B mutually exclusive).}$$

and

$$P(A \text{ or } B) = 1 - P(A' \text{ and } B')$$

The first relation states that if the events A and B are mutually exclusive then the probability of either occurring is simply the sum of the probabilities of the individual events. The second expression shows that the probability of either of two events occurring is 1 minus the probability of neither event occurring, which is an application of DeMorgan's law and complementation.

In this thesis we assume that the reliabilities of the DRAM chip components are independent within the time frame we are discussing. This assumption of independence is based on the estimation that 98% of the faults occurring on DRAM chips during normal operations are due to α-particle hits [Maz88, CH84], (80% for chips in general [YS92]) which occur randomly and almost always affect only 1 cell on a word line (see sections 2.2.4 and 2.3.1).

3.1.3 Stochastic Processes.

A process that depends on random occurrences is described as stochastic. It is the opposite of deterministic because one cannot determine what state a system is in after a certain length of time, only a probability associated with each state. The state of a DRAM chip can be modeled as a stochastic process because we do not know when the chip contains an uncorrectable error (UE), but we can determine the probability that an UE is present over time. With reliability theory we take an assumed distribution of faults and apply the probability relations that we have discussed to analyze the probabilities associated with the system state.

3.1.4 General Reliability

The point in time that an event occurs can be considered a random variable, say X. If we define X to be the time at which a device fails, then the probability that a device fails before a specified time, t, assuming that it is working at time t=0, is called the unreliability , F(t), of a system [TT86]:

$$F(t) = P(X \leq t).$$

The probability that the device continues working through that same time period is called the reliability, R(t), of a system and is given by:

$$R(t) = P(X > t).$$

Because the device must either fail or not fail in that time period, the two probabilities are complementary and must sum to 1:

$$F(t) + R(t) = 1$$

thus: $$F(t) = 1 - R(t).$$

Assuming the device is working at the start of the time period, we have:

$$F(0) = 0, \text{ and } R(0) = 1.$$

And assuming the device is not perfectly reliable, we have:

$$\lim_{t \Rightarrow \infty} F(t) = 1 \text{ and } \lim_{t \Rightarrow \infty} R(t) = 0.$$

The failure probability density function, f(t), which gives the probability of a failure occurring at any instant in time, is expressed as the derivative of the unreliability:

$$f(t) = \frac{dF(t)}{dt}.$$

f(t) is the fraction of the total population that is expected to fail at time t. We also define the portion of working population that is expected to fail, failure rate, z(t), as follows:

$$z(t) = \frac{f(t)}{R(t)}.$$

This is the probability that the component will fail at time t, divided by the probability that the component is still operating at that time.

3.2 Failure Mode and Rate

This section identifies the typical modes of failure in DRAM chips and justifies the use of a constant failure rate in our analysis. We use the Poisson and exponential distributions to relate this constant failure rate to the reliability of DRAM chips. Finally an approximation of the failure rate or soft error rate is found by considering three different sources.

3.2.1 Failure Mode.

Figure 1 is an illustration of a typical failure rate function for DRAM chips which also applies to most mechanical and electronic devices [Int86a, TT86].

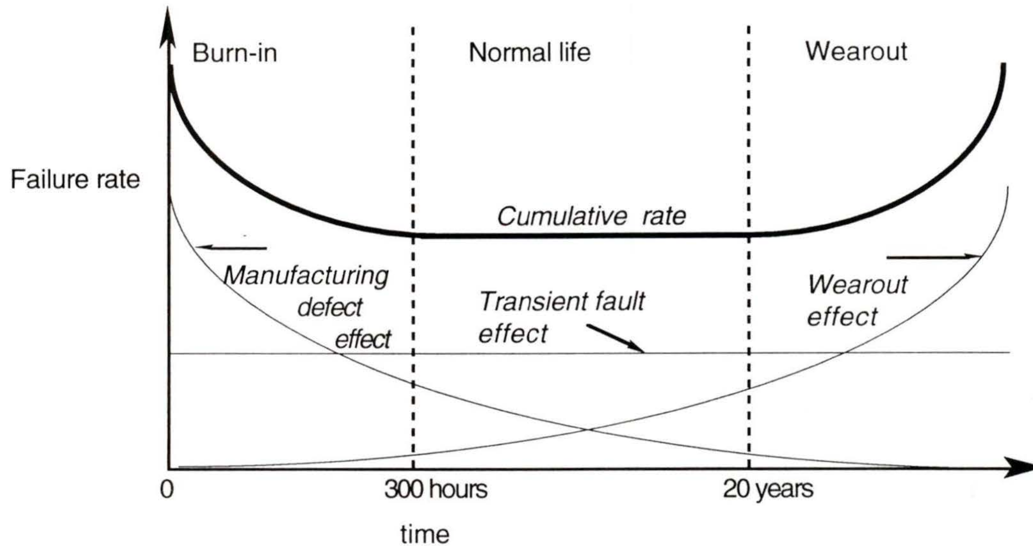


Figure 3.1 An illustration of the failure rate of DRAM chips.

The Burn-in phase is a relatively short period of time (0 - 300 hr) in which the cumulative failure rate is dominated by defects on the chip that cause permanent faults to develop. The Wearout phase is not significant for most computer applications because obsolescence takes the systems out of useful service long before wearout has a significant effect on the failure rate.

The normal life phase of this function illustrates that the cumulative failure rate is essentially constant for most of the useful life of the device [vdG91]. Using λ to refer to this constant failure rate we have:

$$z(t) = \lambda$$

By assuming that the majority of the errors during the normal life of the chip are caused by transient faults, which are primarily caused by α-particles that occur in a random fashion (see section 2.2.4), we can model the failure of a device as a Poisson process [Pag89].

A Poisson distribution describes the number of occurrences of an event within a given time period. Given a random variable Y_t used to indicate the number of occurrences in time (t), and λt to describe the number of occurrences expected, we have:

$$P(Y_t = k) = \frac{1}{k!}(e^{-\lambda t})(\lambda t)^k.$$

The related exponential distribution describes the waiting time to the first occurrence. Using the random variable X again as the time of the first failure, and λ for the expected rate of occurrence, we have:

$$P(X > t) = P(Y_t = 0) = e^{-\lambda t}$$

From this, we see that the waiting time to the first occurrence (or failure) is exponential, with the parameters being the failure rate and time. The reliability, unreliability and probability density functions for a system with a constant failure rate are now defined as follows:

$$R(t) = e^{-\lambda t} \qquad = P(X > t)$$

$$F(t) = 1 - e^{-\lambda t} \qquad = P(X \leq t)$$

$$f(t) = \lambda e^{-\lambda t}.$$

This exponential distribution is commonly used as the basis for analyzing the reliability of computer systems [MW88, Maz88] and in particular to describe the incidence of α-particles [Pag89].

For this thesis we limit our examination of DRAM chip reliability to the Normal life phase of the failure rate function (figure 3.1). We assume that the failure rate is constant and the reliability is exponentially related to the failure rate in time.

3.2.2 Estimating the Failure Rate.

An estimate of the failure rate can be determined empirically and should be cited as part of the specifications of a device. However, most papers that consider the subject of soft errors either hide the rates by normalizing them [SL92, Ari90, Yam92] or exaggerate them to demonstrate the efficacy of their schemes [Maz88, MW88]. One rate we determined from the results of a paper on 16 Kbit RAM chips is 10^{-13} failures per bit per second [CH84]. A more current and complete report on memory cells gives the bit soft error rate (SER) to be 2000 FITs (errors per 10^{9} hours) for 256 Kbit DRAM's which works out to 10^{-14} failures per bit per second [TP89].

The U.S. Department of Defense has published a standard that can be used to calculate the failure rate [Joh89]. The standard, MIL-HNBK-217B [DOD-74], uses a model to predict the failure rate of any IC chip given some basic parameters. The failure rate formula is stated as:

$$\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E) \pi_P.$$

π_L The learning factor. 1 is stable technology and 10 is new .

π_Q The quality of screening. 1 is the highest quality and 150 is typical for commercial components.

C_1 A complexity factor based on the number of bits: $(0.00199)B^{(0.603)}$.

π_T The temperature factor, that accounts for the technology, power dissipation, and operating temperature.

C_2 A second complexity factor based on the number of bits: $(0.00056)B^{(0.644)}$.

π_E Environmental factor. Ranges from 0.2 for a controlled environment, to 10 for a launched missile.

π_P Pin factor. Ranges from 1 to 1.2.

The typical parameter values that describe a 16 Mbit DRAM chip are:

$\pi_L = 10$

$\pi_Q = 150$

$C_1 \approx 45$ for a 16 Mbit chip

$\pi_T = 0.35$

$C_2 \approx 25$ for a 16 Mbit chip.

$\pi_E = 1$

$\pi_P = 1.2$

Using these parameter values we can estimate the failure rate as follows:

$$\lambda_{chip} = 10 \times 150 \times (45 \times 0.35 + 25 \times 1) \times 1.2$$

$$\approx 73800 \text{ bit failures per chip per one million hours.}$$

which gives us:

$$\lambda_{bit} \approx 10^{-12} \text{ failures per bit per second.}$$

For the bit failure rate (per second) we have found a range of 10^{-12} to 10^{-14} failures/s . In chapter 6 we use $\lambda = 10^{-12}$ failures/s or 10^{-21} failures/ns for the bit failure rate in most of our calculations because it gives us a typical value that might be expected in current DRAM chips.

3.3 Reliability Models for Analysis.

The two most commonly used approaches to analyzing the reliability of systems are based on combinatorial modeling and Markov modeling. We use each of these methods in our evaluation of the reliability of DRAM chips.

3.3.1 Combinatorial Modeling.

Combinatorial modeling works by considering the probabilities associated with each component of a system and enumerating each of the possible ways in which a system remains operable. These models are grouped into series models, parallel models and complex models (using both series and parallel subsystems).

Series Model.

In a series model each component must operate properly in order for the system to function properly. Figure 3.2 illustrates the series concept with a system block diagram that shows the flow with each component contributing to the outcome. It intuitively suggests that if one component is broken then the flow would stop at that component and the system fails.

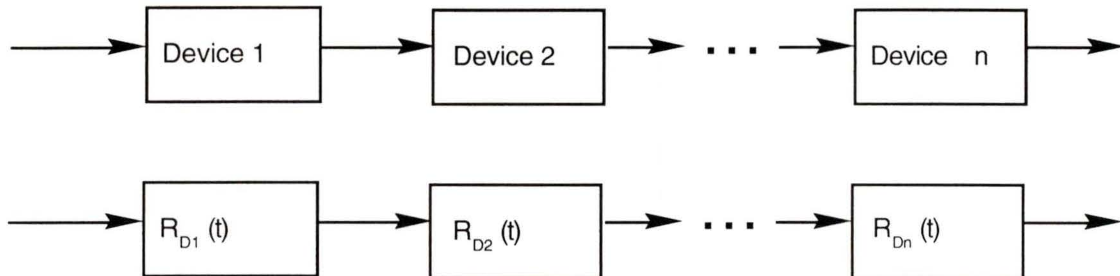


Figure 3.2 A system block diagram for a series model.

By using the basic probability relationship for independent events, P(A and B) = P(A) P(B), we get:

$$R_{sys}(t) = R_{D1}(t)\, R_{D2}(t) \ldots R_{Dn}(t).$$

Thus, the reliability of a series system is the product of the reliabilities of the individual devices that constitute the system.

Parallel Model

In a parallel model (unrelated to parallel processing) only one of the constituent devices needs to be operating correctly for the system to function properly. Figure 3.3 illustrates this concept with the system and reliability block diagrams that show that even just one device operating will allow the flow to pass through the system.

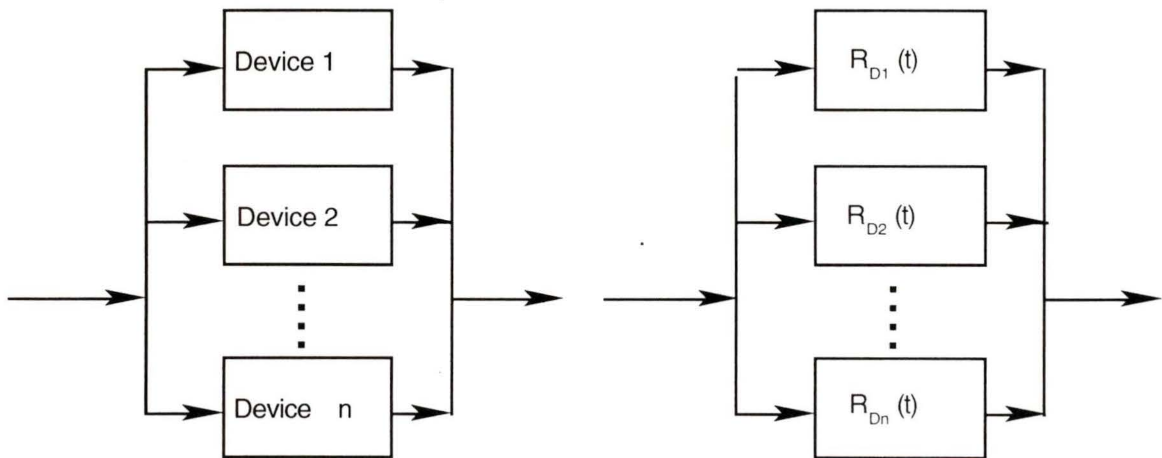


Figure 3.3 A system block diagram for a parallel model.

By using the basic probability and reliability relationships developed earlier:

$$P(A \text{ or } B) = 1 - P(A' \text{ and } B'),$$

$$P(A \text{ and } B) = P(A)\,P(B) \qquad (A,B \text{ independent}), \text{ and}$$

$$F(t) + R(t) = 1$$

the reliability of a parallel system is determined to be:

$$R_{sys}(t) = 1 - F_{D1}(t)F_{D2}(t) \ldots F_{Dn}(t)$$

$$= 1 - (1-R_{D1}(t))(1-R_{D2}(t))\ldots(1-R_{Dn}(t)).$$

In words, the reliability of a parallel system is the complement of the probability that all of the components operating in parallel, fail.

Complex Model.

Complex models have both serial and parallel connections or dependencies between devices and subsystems. The calculation of the reliability of a complex system involves recursively evaluating the reliabilities of the least significant subsystem and substituting a single reliability expression in its place.

Figure 3.4 gives an example of a complex system which requires either device 1 and 2 or device 3 and 4 along with device 5 to work for the system to function properly. The subsequent reliability block diagrams demonstrate the recursive reduction that is necessary to find a single reliability expression. Even for this very simple system the expression becomes very complex.

M of N System

An M of N system, where m of the n components or devices must work in order for the system to function properly, is a special kind of complex system. The reliability of this system can be calculated by summing the probability that exactly m components are working properly and the probability that exactly m+1 components are working properly, and so on up to the probability that all n components are working properly.

The calculation of the reliability of a M of N system involves the use of the binomial distribution model which assumes:

a) each component is either working or not,

b) the reliability of each component is the same,

c) there is a fixed number of components,

d) the components operate independently.

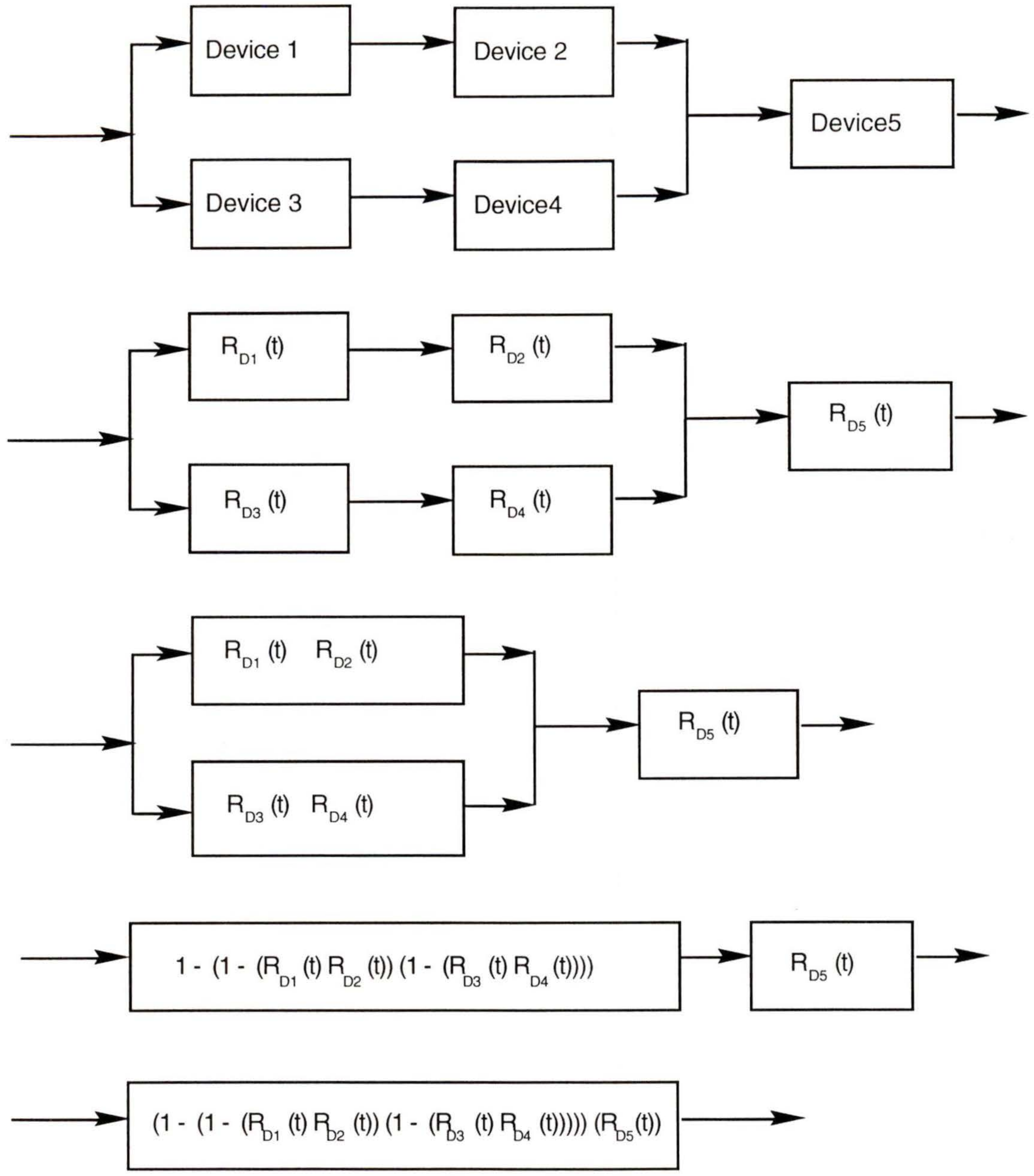


Figure 3.4 A system block diagram for a complex system.

Using X as the variable representing the number of working components, the general expression for the probability of exactly x successes in n trials, with a probability p that a trial is successful is:

$$P(X=x) = \binom{n}{x} p^x (1-p)^{n-x}$$

and thus the probability of m or more components working through a specific time, given that p is the reliability of the component for that time period, is:

$$P(m \text{ of } n) = \sum_{x=m}^{n} \binom{n}{x} p^x (1-p)^{n-x}$$

For example, the reliability of a 4 bit SEC (single error correcting) word after 5 seconds, with a bit failure rate of 0.001 sec^{-1}, is calculated as follows:

$$p = e^{(.001)5} = 0.9950125$$

$$P(3 \text{ of } 4) = \sum_{x=3}^{4} \binom{4}{x} 0.9950125^x (1-0.9950125)^{4-x}$$

$$= 4(\ 0.98511)(.00498752) + 0.9802$$

$$= 0.01965 + 0.9802$$

$$= 0.99985$$

We note that the extreme cases of the M of N system are: the series system which is N of N, and the parallel system which is 1 of N. This method, and variations that assume other distributions is used to calculate the probability of a chip working correctly in the presence of permanent faults at a particular time [CH84, Sta92]. However, we are considering the reliability of a device in which errors are developing in time. To calculate the probability of an UE developing over time would be unmanageable using this method, thus we turn to Markov models.

3.3.2 Markov Modeling

It is often difficult to derive a block diagram that completely describes a complex system and even then the resulting reliability expressions can become very unwieldy. In addition, it is difficult to include many different modes of change in a system such as repair or replacement [Joh89]. A special case of this occurs with the use of ECC's in memory. When an ECC block is accessed, the memory cell array word containing the block is checked, corrected if necessary (and possible) then rewritten. To facilitate the analysis of these more complex systems we use Markov models.

A Markov model consists of a collection of states and transitions. Each condition (or set of conditions) that a system may be in is modeled as a state, and the probability of a change from one particular state to another during the next segment of time is modeled as a transition.

As an example, figure 3.5 shows a Markov model for a 4 bit SEC word. The circles represent states, where the NE state corresponds to no errors, the SE state corresponds to one error and the UE state corresponds to more than one bit in error (an uncorrectable error). It is a simplified model which assumes that there is no mechanism for repair (scrubbing), that a fault does not correct an error caused by a previous fault, that only one fault occurs during an increment of time and that a fault affecting more than one bit does not occur. Chapter 5 contains more accurate models which we use in our analysis.

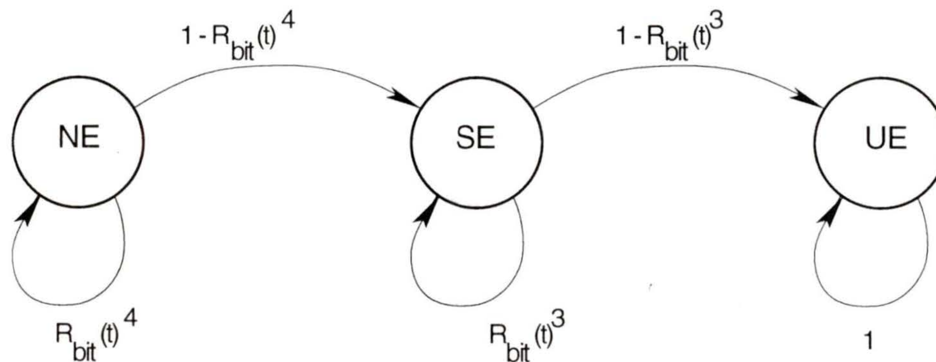


Figure 3.5 A Markov model for a 4 bit SEC word.

To allow for easy calculation of probabilities over time a transition matrix is derived from the Markov model using a time increment of 1. The transition matrix contains the probability of a transition from state X to state Y in T[X,Y].

Using $1 - \lambda t$ as an approximation for the value of $e^{-\lambda t}$ when λt is small, the following approximate transition matrix is generated from the model in figure 3.5:

$$\mathbf{T} = \begin{bmatrix} 1-4\lambda t & 4\lambda t & 0 \\ 0 & 1-3\lambda t & 3\lambda t \\ 0 & 0 & 1 \end{bmatrix}$$

The first row gives the probabilities that the system in the no error state, NE, will be in NE, SE, and UE states, after the next increment of time. Likewise, the second row gives the probabilities for the system in the single error state, SE, and the third row gives the probabilities for the system in the uncorrectable error state, UE.

Given an initial state vector $\mathbf{P_0}$ = [NE, SE, UE], the next state of the system is defined as:

$$\mathbf{P_1} = \mathbf{P_0} \cdot \mathbf{T}$$

Thus with a transition matrix, **T**, and an initial state vector, $\mathbf{P_0}$, we can calculate the probabilities that the system will be in each of the states after a set number of cycles, n, with [TAN90]:

$$\mathbf{P_n} = \mathbf{P_0} \cdot \mathbf{T^n}$$

Given $\mathbf{P_0} = [1\ \ 0\ \ 0]$, a failure rate ($\lambda$) = 0.001, and the transition matrix **T** from above we have:

$$\mathbf{T} \approx \begin{bmatrix} 0.996 & 0.004 & 0 \\ 0 & 0.997 & 0.003 \\ 0 & 0 & 1 \end{bmatrix}$$

we can calculate the new probability state vector after each second as follows:

$$\mathbf{P_1} = \mathbf{P_0} \cdot \mathbf{T^1} \approx [0.996, 0.004, 0]$$

$$\mathbf{P_5} = \mathbf{P_0} \cdot \mathbf{T^5} \approx [0.980159, 0.019722, 0.000119]$$

From this example we see that the probability that the word has failed after 5 seconds is

$$F(5) = 0.000119,$$

and thus the reliability of the 4 bit word after 5 seconds is:

$$R(5) = 0.999881.$$

Generalizing, we see that the probability the system is in the NE state is decreasing. The probability the system is in the SE state is increasing while $P(NE) > \frac{3}{4} P(SE)$, then it starts to decrease. And finally, the probability the system is in the UE state always increases.

Ultimately we have:

$$\lim_{n \Rightarrow \infty} \mathbf{P_n} = \mathbf{P_0} \cdot \mathbf{T^n} = [0, 0, 1].$$

Thus, if we assume that the activities in a DRAM chip occur in a discrete fashion and that only one change occurs during an increment of time we can use this model. Most of the activities in memory chips do follow this assumption if the time increment is small enough. Although α-particles occur in a continuous fashion, we can approximate their occurrence and effect in this discrete model by using a very small time increment. In chapter 4 we introduce the possibility of scrubbing (repair) to improve this model.

3.4 Summary.

This chapter reviews the basic probability and reliability relationships that are used in our analysis of the reliability of DRAM chips. Terms such as stochastic process, independence and failure rate are defined and related to our subject. Different models that are used for analyzing the reliability of systems are discussed and examples are given to demonstrate some of the methods used in reliability analysis.

For this thesis we assume that failures that occur during the normal life phase of a chip are independent and randomly distributed, and that the exponential distribution describes the reliability of a bit over time. A Markov model is used to analyze the reliability of ECC (error correcting code) blocks in a DRAM chip and then a series system model is used to consider the reliability of the chip as a collection of ECC blocks.

In the next chapter we discuss the factors and formulas that are specifically related to the reliability analysis of DRAM chips.

Chapter 4 Reliability of DRAM Chips

We begin this chapter by clarifying our use of the terms reliability and uncorrectable error. This is followed by a justification for our consideration of this topic by relating the buildup of uncorrectable errors to the birthday problem.

In order to proceed with our analysis we first describe all of the terms that we use. Some of these terms are straightforward and have typical values that need little explanation, while others are less obvious and require estimates.

The effects of permanent faults and scrubbing (a form of concurrent error correction) are examined and shown to have a significant effect on the potential reliability of ECC blocks.

We develop expressions for the reliability of DRAM chips that account for the differing types of ECC blocks that may be found on a chip. From this base we are able to calculate (in chapter 6) relative improvements in DRAM reliability using different error correction schemes as described in chapter 5.

Finally we consider other factors or possibilities which we have set aside either because they are not currently used at the DRAM chip level or because their effects are limited.

4.1 Clarification and Justification

4.1.1 Unreliability in a DRAM Chip

In this thesis when we write about the reliability of a bit, or an ECC block we are referring to the condition in which there are no uncorrectable errors (UE's) present. When considering bits and ECC blocks this is reasonable because if an uncorrectable error is present when you consider that element, then a failure has occurred.

We extend the direct relationship between the existence of an UE and failure to chips because it provides a reasonable metric for measuring and comparing unreliability in DRAM chips. Thus, in this thesis we use the terms *unreliability* and *probability of an UE* synonymously.

4.1.2 Justification for Consideration

We have assumed that the soft errors in a DRAM chip are randomly distributed both in time and among the bits. Over time, soft errors accumulate in the DRAM chip and the probability of an uncorrectable error increases.

The probability of two errors occurring in the same 137 bit ECC block of a multimegabit chip seems remote, but if we liken errors to people and ECC blocks to birthdays then we just have a big birthday problem [SL92]. Given X as the number of accumulated errors, and n as the number of ECC blocks, we have:

$$P(UE|\,X) = 1 - \prod_{i=0}^{X-1}\left(\frac{n-i}{n}\right)$$

$$= 1 - \frac{n!}{n^{x}(n-x)!} \,.$$

Using our model chip, where n = 131,072, we find:

X	P(UE|X errors)
10	0.0003
20	0.0014
30	0.0033
40	0.0059
50	0.0093
60	0.0134

Although these probabilities appear low, they are significant when compared to a reliability requirement of $0.9_{8}5$ for a 10 hour period in a critical application [Pra86]. To keep the system reliable then, we need to remove these errors from the blocks while the memory is in use. This process is called scrubbing and is described further in section 4.5 [MW88, RK91].

4.2 Basic DRAM Chip Parameters

4.2.1 Chip Parameters Defined

The DRAM chip that we are discussing is defined by the following parameters:

w	=	the number of bits in an ECC block
n	=	the number of ECC blocks in the chip
N	=	nw = the number of bits in the chip
k	=	the number of memory islands
t	=	time in nanoseconds
λ	=	the failure rate of a bit.

4.2.2 A Typical DRAM Chip

Current papers on DRAM chips are discussing 4,16, 64, and 256 Mbit chips [SL92, Ito90, Ari90, TP89]. All of these chips incorporate error correction schemes to overcome transient faults and allow the chip to be produced at viable yield levels. From these reports we have selected parameter values that reflect the typical architecture of DRAM chips. We select a single DRAM chip to enable us to compare the efficacy of different error correction schemes using a constant base for our analysis.

Thus, for our analysis, we specify a hypothetical 16 Mbit DRAM chip (2^{24} bits). It is divided into 16 - 1 Mbit islands each containing independent error correction circuitry. The single error correcting (and double error detecting) code uses 137 bit blocks consisting of 128 data bits (2^7) and 9 check bits (7% overhead). There are 131,072 (2^{17}) ECC blocks in the chip. This chip model approximates the current leading edge of DRAM chip production.

4.2.3 Parameter Values Assigned

The values assigned to the chip parameters are as follows:

w = 137 bits (128 data + 9 check bits)

n = 131,072 blocks

N = nw = 17,956,864 bits
(16,777,216 data + 1,233,648 check bits)

k = 16 islands

λ = 10^{-21} failures per ns (see section 3.2.2).

4.3 Basic Reliability Relationships in DRAM Chips

Using the definitions of reliability and the assumptions of chapter 3, we summarize some of the reliability relationships in DRAM chips. We use the symbol R(t) extensively in the next few sections to refer to the reliability of different elements (differentiated by subscripts).

4.3.1 Reliability of a Bit

For the reliability of a single bit on the chip we have:

$$R_{bit}(t) = e^{-\lambda t}$$
$$= e^{-10^{-21} t}.$$

As we discussed in section 3.2, the reliability of a bit is exponentially related to the failure rate over time.

4.3.2 Reliability of an ECC Block

We are not able to give succinct definitions for the reliability of ECC blocks because these reliabilities are not easily expressed directly. In chapter 5 we present three schemes which can be used in DRAM chips to maintain or improve reliability. Markov models are used to estimate the values for ECC block reliabilities over time in chapter 6. The following terms for ECC block reliabilities are defined to allow us to develop reliability expressions for DRAM chips:

$R_{ECC0}(t)$ = reliability of ECC block using scheme 0.

$R_{ECC1}(t)$ = reliability of ECC block using scheme 1.

$R_{ECC2}(t)$ = reliability of ECC block using scheme 2.

4.3.3 Reliability of a DRAM Chip

Given the reliability of the ECC blocks we are able to calculate the reliability of the chip by applying the series model of section 3.3.1. For the chip to be considered functioning all blocks must be working and thus the reliability of the chip is the product of the reliabilities of each of the ECC blocks. To refer to the reliability of a chip under each of the schemes we use:

$R_{C0}(t) = (R_{ECC0}(t))^n$ chip using scheme 0,

$R_{C1}(t) = (R_{ECC1}(t))^n$ chip using scheme 1,

$R_{C2}(t) = (R_{ECC2}(t))^n$ chip using scheme 2.

Each of these functions assumes that all of the ECC blocks are functioning correctly. More accurate expressions for chip reliability are developed later in this chapter as we present the necessary background information.

For comparison purposes we consider the reliability of an equivalent 16 Mbit chip that does not implement any error correction scheme (identified by $R_{CX}(t)$). Here we are assuming that the manufacturer is able to produce such a chip, with no faults, in sufficient quantities to make production viable.

For the reliability of a 128 bit block of bits ($R_{128}(t)$) we have:

$$R_{128}(t) = (R_{bit}(t))^{128}$$
$$= e^{-128\lambda t}$$
$$R_{128}(1\text{ s}) \approx 0.9_8872,$$

and for a chip containing 131072 128 bit blocks we have:

$$R_{CX}(t) = (R_{128}(t))^{131072}$$
$$= (e^{-128\lambda t})^{131072}$$
$$= e^{-16777216(10^{-21})t}.$$

Note that the reliability of the chip is not dependent on our arbitrary use of 128 bit blocks because the exponent laws eliminate the distinction between bits and blocks.

Checking the reliability of this chip for different lengths of time we find:

$$R_{CX}(1\text{ second}) \approx 0.999983$$
$$R_{CX}(1\text{ day}) \approx 0.2347$$

Obviously the reliability of a 16 Mbit chip with no mechanism to fix soft errors is unacceptable. This demonstrates the need for on chip error correction even if manufacturers could achieve high enough yields without it.

4.4 DRAM Chips Containing Permanent Faults

4.4.1 Permanent Faults and their Distribution

Error correction in DRAM chips is used to mask the permanent faults that are on the chip. The manufacturer only guarantees that the chips shipped operate in a functionally fault free manner, not that there are no hidden or masked faults. Manufacturers currently average only 1 fault in a 1 Mbit DRAM chip. Thus we estimate the initial number of permanent faults in our model chip to be less than 16. However, most designs incorporate redundancy that enables the manufacturer to clean up a substantial portion of these faults in conjunction with the testing process.

To determine the reliability of a 16 Mbit chip with permanent faults we perform the analysis in parts. We know that there is at most 1 error producing fault in an ECC block because more would cause the chip to be discarded. It is expected that only half of the permanent faults (typically < 8) cause errors while the other permanent faults are hidden by the data (eg: a 1 stored in a cell stuck-at-1).

4.4.2 An ECC Block with a Permanent Fault

We must now use two formulas for the calculation of the reliability of an ECC block, one for a block with no errors, and another for an ECC block with a permanent fault causing an error. We are ignoring the small difference in reliability between an ECC block that has a permanent fault which is hidden and a block with no fault. Although we have segregated the ECC blocks, the expressions for the chip reliabilities still utilize a series model.

The reliability of an ECC block ($R_{pf}(t)$) with one permanent fault that causes an error is defined as follows:

$$R_{pf}(t) = (R_{bit}(t))^{136}$$

$$= e^{-136\lambda t}.$$

Using this expression we calculate some reliabilities:

$$R_{pf}(1\ s) \approx 0.9_{8}864$$

$$R_{pf}(1\ day) \approx 0.9_{4}882$$

$$R_{pf}(1\ year) \approx 0.99572.$$

The reliability of the ECC blocks that contain permanent faults causing errors is even lower than the equivalent 128 bit data block with no ECC because there are more bits that can be affected by transient faults (136 vs. 128). Fortunately there are only a small number of blocks containing permanent faults on the chip.

4.4.3 A DRAM Chip with Permanent Faults

The reliability expressions for the chips under the different schemes each containing 8 permanent faults can be defined as follows:

$$R_{C0}(t) = (R_{ECC0}(t))^{n-8}(R_{pf}(t))^{8} \quad \text{scheme 0}$$

$$R_{C1}(t) = (R_{ECC1}(t))^{n-8}(R_{pf}(t))^{8} \quad \text{scheme 1}$$

$$R_{C2}(t) = (R_{ECC2}(t))^{n-8}(R_{pf}(t))^{8} \quad \text{scheme 2}$$

The exponents are related to the number of permanent faults.

4.5 Error Correction

The reliability of DRAM chips as presented above does not yet accurately present reality. Thus far we have ignored the process that occurs during an access that corrects errors in the selected ECC block. This process is an extended concurrent error correction. It acts both on the portion of data selected and on other unselected bits of data.

4.5.1 Scrubbing

In DRAM chips, when an ECC block is accessed, it is checked before the information is used or updated. If one bit is wrong then the bit is corrected and the block written back. When the ECC facility cannot handle the error then the chip either fails (returns an erroneous value) or reports an error. The process occurs on both read and write accesses, it occurs completely independently and is transparent to the system. The typical ECC's used in DRAM chips are single error correcting and double error detecting, SEC/DED, meaning that anything more than a single error in an ECC block is uncorrectable.

This correction of soft errors using the ECC capability during access of memory blocks is called scrubbing [RK91]. It refers to both the passive form which is a side effect of a regular access and the active form which we are proposing in schemes 1 and 2 described in the next chapter.

Because the memory is generally the system bottleneck, we assume that it is being accessed at its maximum rate (a constant). For our model chip we assume an access rate of 0.02 accesses per ns. If we assume that the accesses are distributed uniformly and randomly, then we find the rate of scrubbing, γ_u, for a block to be:

$$\gamma_u = \frac{0.02}{n}$$

$$\approx 0.000000153 \text{ accesses per ns.}$$

The constant scrubbing rate allows us to use the exponential law to determine the probability that a block is not scrubbed. The probability that an ECC block with one error is scrubbed one or more times is given as:

$$S(t) = 1 - e^{-\gamma t}$$

for example:

$$S(1s) = 1 - e^{-153}$$

$$\approx 0.9_{66}643$$

This shows us that in a chip which is continually uniformly accessed, it is essentially a certainty that any error that occurs is corrected in less than a second. Calculating the probability of a single error occurring in a 16 Mbit DRAM chip with ECC in 1 second we find:

$$1 - R(1s) = 1 - e^{-\lambda tnw}$$

$$= 1 - e^{-(10^{-21})(10^{9})(131072)(137)}$$

$$\approx 1 - e^{-0.000018}$$

$$\approx 1 - 0.999982$$

$$\approx 0.000018.$$

Using the assumptions stated to this point we find that in the space of 1 second it is unlikely that a soft error occurs and almost certain that any existing soft error is corrected. This situation can be related to the birthday problem with the added complications that the probability of a person entering the room is very low and every few seconds the room is cleared. However, as described in the next section, the assumption that accesses are randomly and uniformly distributed is wrong.

4.5.2 Distribution of Scrubbing

In a memory system the access distribution is determined by the processes that are running. A process tends to spend the majority of its run time in very small portions of the code and data space. This phenomenon is referred to as the principle of locality [Hab76].

Using the principle of locality, we may assume that the distribution of memory accesses is nonuniform and that there are favored portions of the memory which have very high probabilities of access and the remainder of the memory has a low probability of access. We use the terms favored and ignored memory to discuss these different portions of memory.

In figure 4.1 we give an illustration of the different rates of access across the memory space. This particular diagram suggests that there are two blocks of memory that are accessed very frequently while the remaining memory cells are generally ignored. We note that this distribution is just an arbitrary example and that in general the distribution is related to the applications running on the system.

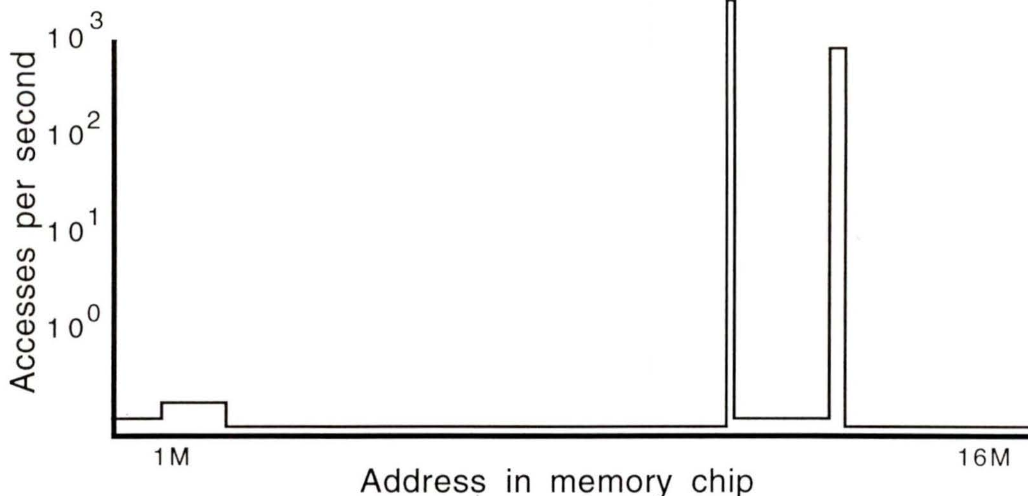


Figure 4.1 Illustration of access rate by memory address.

Figure 4.2 gives the simplified model of the probability of access graph that we assume for our analysis. Our consolidation of the favoured memory is justified because the soft error rate is random and uniformly distributed meaning that the position of the favored and ignored blocks will have no effect. We also average the rates for ignored and favored memory to simplify our analysis.

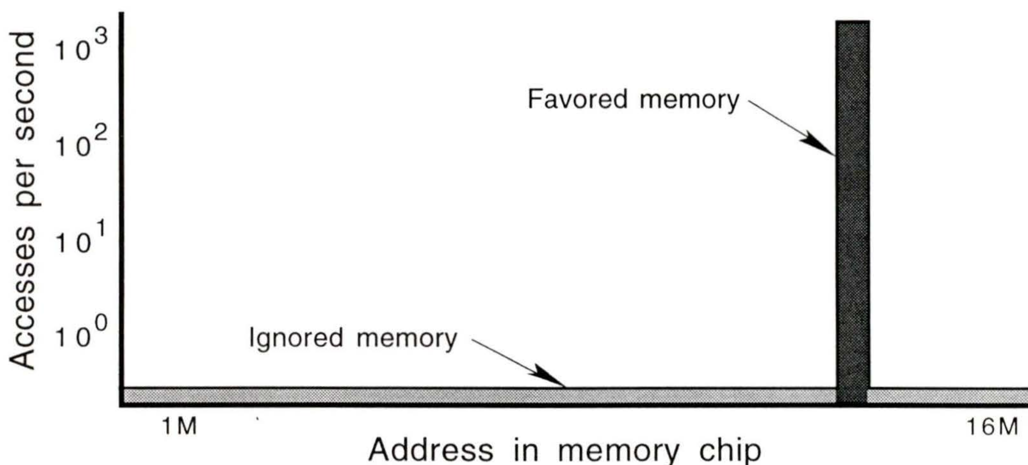


Figure 4.2 Simplified illustration of access rate in memory.

4.5.3 Effect of Scrubbing on Reliability

The ECC blocks in the favored memory have an even better probability of being scrubbed than was calculated when we assumed a uniform distribution of access. The graph illustrating an access rate in the favored memory to be more than 10000 times greater than the access rate of the ignored memory shows that a significant difference exists between the average access rates.

We use an arbitrary selection of the size of the favoured and ignored memory and the access rates to allow us to compare different scrubbing schemes in chapters 5 and 6. For our analysis, we consider a DRAM chip with 512 Kbits of data (4096 ECC blocks) in the favored memory and 15872 Kbits (126976 ECC blocks) in the ignored memory. The rates of access to the favored and ignored ECC blocks are assumed to be:

$$\gamma_f \approx 10^{-6} \text{ accesses per ns}$$

$$\gamma_i \approx 10^{-11} \text{ accesses per ns.}$$

From these rates we calculate the probability of scrubbing occurring in the favored and ignored memory, using S_f and S_i, as follows:

$$S_f(t) = 1 - e^{-\gamma_f t}$$

$$S_i(t) = 1 - e^{-\gamma_i t}$$

$$S_f(1s) \approx 0.9_{4345}$$

$$S_i(1s) \approx 0.00995$$

$$S_i(60s) \approx 0.451$$

$$S_i(3600s) \approx 0.9_{15}232$$

From these results we see that it is very likely that a favored block will be scrubbed in less than a second. However, for the majority of the bits in the memory chip, $(\frac{31}{32})$, the probability of access is not high enough to ensure that an ECC block would typically be scrubbed even once a minute. In our analysis of the reliability of the ECC blocks and chips in chapter 6 we see that the different scrubbing rates have a significant effect on chip reliability.

If a chip has half of its blocks accessed at one rate (γ_1), and the other half accessed at another rate (γ_2), then the block reliabilities must be calculated separately ($R_1(t)$ and $R_2(t)$). Taking these reliabilities along with the reliabilities of the p blocks containing the permanent faults, the formula for the reliability of the chip is estimated by:

$$R_{Chip}(t) = (R_1(t))^{(n-p)/2} (R_2(t))^{(n-p)/2} (R_{pf}(t))^p$$

4.6 Other Factors

There are many other internal and external factors which might have an effect on DRAM chip reliability that are not included in our analysis. Some factors may not be relevant because they are not used in current DRAM chip designs. Other factors are left to one side because their net effect is much less significant than the effect of the schemes we are considering. And ultimately we must restrict our model to a limited number of factors in order to allow the analysis to proceed.

4.6.1 Internal Factors

Permanent faults can develop over the life of the chip. However, because 98% of the DRAM chip failures are attributed to transient faults caused by α-particles [vdG91] the effect of developing permanent faults is relatively insignificant.

Extended fault tolerance (see section 2.3.1) would improve the DRAM chip reliability. The cost in complexity and delay would be too great to justify the use of this technique in most chips. It is more likely that this type of fault tolerance would be implemented at the memory system level to mask the errors that are not correctable at the chip level.

The address decoder scheme in the chip can affect the reliability of the chip. If the purpose of the address decoder is to maximize the number of ECC blocks touched, then the access rate chart would flatten considerably, thereby improving the rate of passive scrubbing. However, this is contrary to the objective of improving the average access time using block reads, which is a primary concern in the development of DRAM chips [Kal90].

4.6.2 External Factors

External factors that may affect the reliability of the DRAM chips include the cache mechanism, and the workload of the memory system. In the cache system we might consider the effects of block size, miss ratio, and the swapping algorithm [PA83]. The memory workload and activity would have an effect on the access or scrubbing rates. For example, if a small PC has many processes halted in memory while only one is being used, the memory used for the ignored processes is not being scrubbed.

By considering the differences between the ECC schemes we are investigating while keeping all other factors equal we show how the reliability of a chip may be improved. These improvements may not be as great if the factors are chosen in an adversarial fashion but they would still be significant.

4.7 Summary

We have defined the parameters that we are using in our models for the analysis that is presented in chapters 5 and 6. The values chosen represent approximations. The specific values are arbitrary but realistic and are kept constant to allow us a basis on which to compare different schemes that are discussed in chapter 5.

Justification for proceeding with the analysis is provided with the interim analysis of the reliability of DRAM chips with no error correction ability. The reliability of a chip is defined in terms of the reliability of the ECC blocks which we discuss further in chapter 5 and analyze in chapter 6.

The access rate (read or write) is shown to have a positive effect on the reliability of the DRAM chip because of the scrubbing (removal of soft errors). We show that if there is a uniform distribution of accesses then the reliability of the ECC blocks, and therefore the chip, would be very high.

The principle of locality is introduced to show the distribution of accesses is very uneven. We assume a simple distribution where a small part of the memory is favoured (accessed very frequently) and the remainder is ignored (accessed relatively infrequently). The ratio of favoured memory to ignored memory as well as the magnitude of the difference between the access rates depend upon the system, the application and the workload. We have arbitrarily selected values that might observed in a typical system. Other values could be used and they would affect the outcome of our analysis of the reliability of the DRAM chips, but they would only affect magnitudes and not the underlying differences between the schemes that are defined and analyzed in chapters 5 and 6.

The expression for the reliability of a DRAM chip developed in this chapter depends upon the access distribution (principle of locality) and the number of permanent faults. To use it we must develop a model for estimating the reliability of the ECC blocks that accounts for differences in the scrubbing rates. In chapter 5 we define three schemes that have different access distributions (and scrubbing rates) which we then analyze in chapter 6.

Chapter 5 Scrubbing Schemes

We begin this chapter by describing the differences between active scrubbing and passive scrubbing. Then we explain the standard application of ECC based scrubbing in DRAM chips (scheme 0) and introduce two new schemes for enhancing the reliability of DRAM chips. This is followed by a summary of the parameters involved in the analysis of each of the three schemes. Finally, we present Markov models and transition matrices for the ECC blocks in each of the schemes we are suggesting. These models are used in chapter 6 to analyze the reliability of both the ECC blocks and the DRAM chips incorporating these schemes.

5.1 Scrubbing in DRAM Chips

The term scrubbing in memory systems or chips is used to refer to the process of checking for errors and correcting them (if possible) when they are found. Passive scrubbing, used to describe the process of checking and possibly correcting an ECC block during a normal access, is now a standard fault tolerance feature on large DRAM chips. This is not just concurrent error correction, because it involves correcting errors in data that has no logical connection to the data being accessed, but simply shares the same ECC block.

Because the accesses to ECC blocks are unevenly distributed, we must do something to improve the access rate (and thus the scrubbing of errors) for the ECC blocks in ignored memory. Active scrubbing (parallel error correction) schemes which check memory cells that are not being directly accessed are commonly suggested and used at the memory system level [MW88, Aic84, RK91]. We believe there is potential for similar schemes to work in DRAM chips.

In our model chip which is described in section 4.2, there are 16 memory islands. The transfer of all 137 bit blocks to a central circuit would increase the complexity of the chip to an unacceptable level, thus each of these islands have their own ECC circuitry. When a bit is requested from the chip, only the island containing the bit is active, while the other 15 islands are idle. By using the potential error correction capability of these other memory islands, we can improve the scrubbing rate in the DRAM chip and thus reduce the probability of an UE developing.

The primary concerns during chip design involve speed, area overhead, and complexity. Because ECC schemes are already commonly used in current DRAM chips (passive scrubbing), the price of incorporating active scrubbing is substantially covered. We are suggesting two schemes that utilize the existing capabilities in a more extensive manner. Thus the issues of additional area and complexity are limited to controlling circuitry.

5.2 Three Schemes for Scrubbing DRAM Chips

5.2.1 Scheme 0 - Standard 16 Mbit Chip with ECC

We define scheme 0 as a standard on which to base our analysis of the efficacy of schemes 1 and 2. In chapter 4 we describe the basic 16 Mbit chip which we use for our analysis. Thus scheme 0 is our control model that uses only passive scrubbing as is typically found in multimegabit DRAM chips. The ECC circuitry allows the chip to scrub one error in a block of 137 bits (128 data bits and 9 code bits) during each access (read or write).

5.2.2 Scheme 1 - Parallel Scrubbing

In scheme 1 we use the idle ECC circuitry that is contained in each memory island to check and correct 15 other ECC blocks in parallel with the accessing, checking, and correcting of the block containing the requested data. In the DRAM chip we are using for our analysis, 4 of the address bits are used to select the appropriate memory island, 13 of the address bits are used to select the ECC block in the island and the remaining 7 address bits are used to select one of the bits from the 128 data bits contained in the ECC block. The 13 bits that form the address of the ECC block in the selected island are used by each of the 15 other memory islands to sense, check and possibly correct an ECC block in parallel with the regular access.

Figure 5.1 shows the possible effect of this scheme by compounding the access pattern given in figure 4.1 while shifting the address by 1 Mbit each time. The multiple copies of the favored memory peaks indicate that many of the ignored blocks are now checked when the favored memory is accessed. The ECC blocks in the ignored memory which do not coincide with the image of the ECC block in the favored memory are still accessed 16 times more frequently. From this, we see that this scheme improves the access rate on the ignored memory by a factor of at least 16.

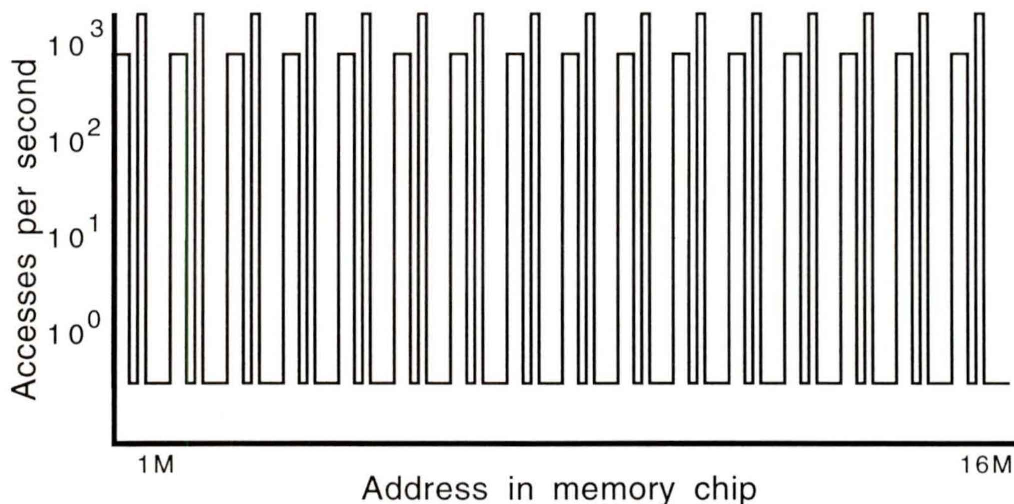


Figure 5.1 Illustration of ECC block access rate using scheme 1.

The favored memory, and those portions of the ignored memory that share the same ECC block addresses in other memory islands, now account for half of the memory. This half is extremely reliable because the potential for correcting soft errors (scrubbing) is high. In addition, the ignored half of the memory has an improved rate of access and thus is also more likely to be scrubbed.

Figure 5.2 illustrates the simplified model of access distribution which we use as a basis to allow us to compare the reliability of this scheme to the reliability of the standard DRAM chip. Our consolidation of the favored and ignored memory is again justified by the fact that transient faults which cause errors are randomly and uniformly distributed.

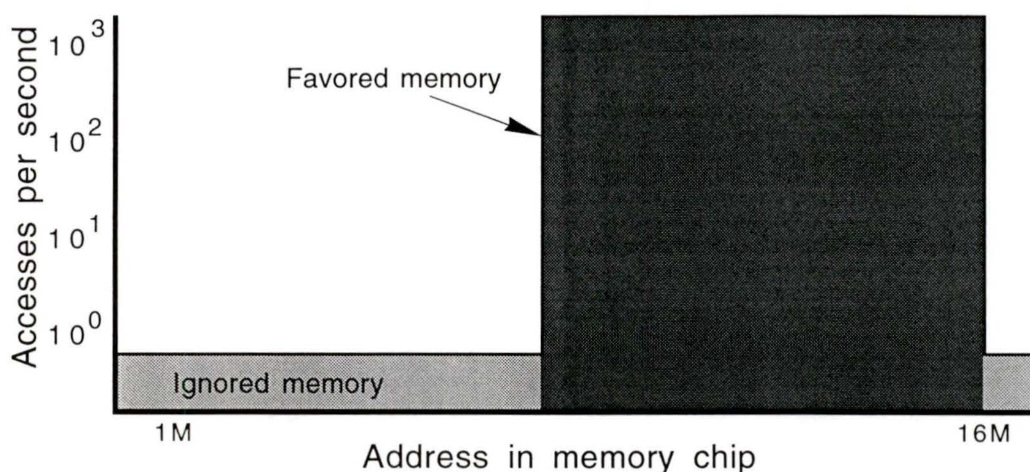


Figure 5.2 Simplified illustration of access rate using scheme 1.

For scheme 1 then, we assume that half of the bits make up the ignored memory. The number of accesses to the ignored memory has been increased by a factor of 16 ($\gamma = 1.6\text{x}10^{-10}$ accesses per ns) giving us an ECC block scrubbing probability of:

$$S(t) = 1 - e^{-\gamma t}$$

$$S(1 \text{ second}) \approx 0.147856$$

$$S(1 \text{ minute}) \approx 0.9_{4}3227$$

$$S(1 \text{ hour}) \approx 0.9_{250}298$$

These figures are significantly better than the scrubbing probability of a block in the ignored memory of a standard DRAM chip as calculated in section 4.5.3.

5.2.3 Scheme 2 - Systematic Scrubbing

Scheme 2 also makes use of the idle ECC circuitry to check and correct ECC blocks in each of the memory islands in parallel with a regular access. In scheme 1 we use the 13 address bits that identify the ECC block in each of the memory islands. Here we suggest a 13-bit counter to systematically and uniformly access all of the blocks in each island.

In this scheme each regular access is also accompanied by 15 extra accesses, one in each of the ignored memory islands. Thus, there are 20,000,000 accesses per second almost uniformly spread over the 8192 ECC blocks in each memory island. Figure 5.3 illustrates that each block (and thus each bit) is typically accessed more than 2000 times per second, essentially assuring us that a soft error will be corrected within 0.0005 seconds. We note that if the majority of the favored memory resides in a single memory island then a portion of the ignored memory will not be accessed as frequently. This is shown in figure 5.3, but we ignore this degenerate case in our analysis.

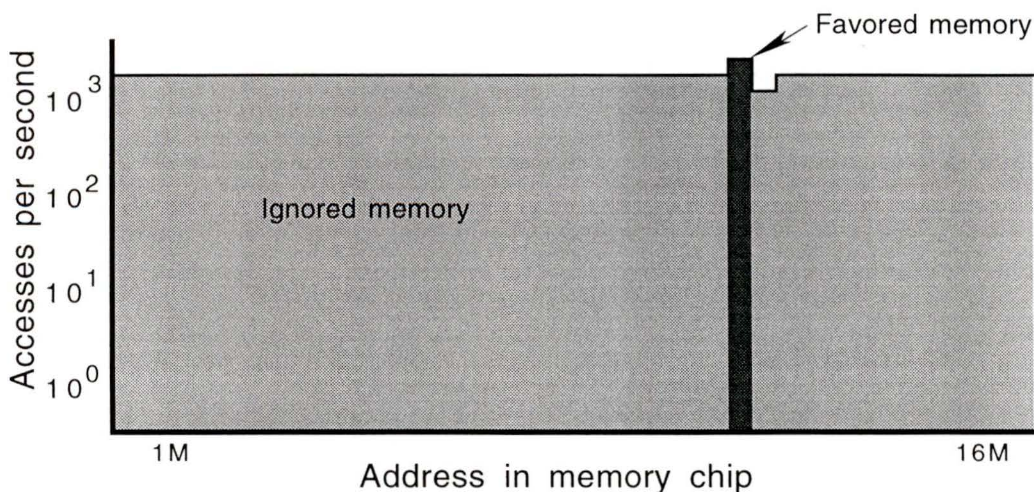


Figure 5.3 Simplified illustration of access rate using scheme 2.

Although this parallel scrubbing is occurring in a systematic fashion instead of randomly, we approximate it as a random distribution with a constant rate. This allows us to use the exponential distribution to describe the probability of a block being scrubbed in the next ns. The random approximation results in a conservative (higher) estimate of the time since the ECC block was last scrubbed, and this conservatively affects (increases) the probability that an ECC block has a single error. We require the random approximation in order to allow us to use a Markov model to estimate the state of the ECC blocks over time.

In scheme 2 there may be some parts of the ignored memory that do not get scrubbed as often because they share the same memory islands as the favored memory (see figure 5.3). However, the effect is minimal and for our analysis we assume that all ignored blocks have similar access frequencies.

5.2.4 Cost of Implementation of Schemes

Our analysis of the cost of implementation of the schemes presented is limited to an intuitive overview. We know that for current DRAM chips the default scheme is scheme 0, meaning that the cost is already acceptable. Scheme 1 does require some extra control signals, but really adds relatively little to the complexity of the chip. In fact, the costs of implementing scheme 1 may be almost entirely paid by the incorporation of parallel testing facilities in the chip to keep the cost of testing reasonable [KKHO89, KS86, PK88]. Scheme 2 costs the most because extra counters as well as control signals are required to implement it.

5.3 Parameters for the Scrubbing Schemes

5.3.1 Scheme 0 Parameters

For scheme 0 we use the following parameters to describe a basic 16 Mbit chip:

ECC blocks:

favored blocks	=	4096
ignored blocks	=	126976

Access rate:

favored blocks (γ_f)	$\approx$	4.8828×10^{-6} accesses/ns
ignored blocks (γ_{i0})	$\approx$	10^{-11} accesses/ns

Soft error rate (λ): $\approx$ 10^{-21} failures/ns

5.3.2 Scheme 1 Parameters

The parameters for a chip incorporating scheme 1 are as follows:

ECC blocks:

favored blocks	=	65536
ignored blocks	=	65536

Access rate:

favored blocks (γ_f)	$\approx$	4.8828×10^{-6} accesses/ns
ignored blocks (γ_{i1})	$\approx$	1.6×10^{-10} accesses/ns

Soft error rate (λ): $\approx$ 10^{-21} failures/ns

5.3.3 Scheme 2 Parameters

Finally for a chip using scheme 2 we use the following parameters:

ECC blocks:

favored blocks	=	4096
ignored blocks	=	126976

Access rate:

favored blocks (γ_f)	$\approx$	4.8828×10^{-6} accesses/ns
ignored blocks (γ_{i2})	$\approx$	2.4414×10^{-6} accesses/ns

Soft error rate (λ): $\approx$ 10^{-21} failures/ns

5.4 Markov Models for the ECC Blocks

To accomplish the analysis of the reliability of a DRAM chip we analyze the unreliability of the different types of ECC blocks using Markov models and then use the series model (see section 3.3.1) to combine the results in chapter 6.

In a DRAM chip, the transient faults occur in a continuous fashion and the accesses occur in a discrete fashion. In our analysis, we approximate the chip behavior with a discrete model using a one nanosecond time step and assume that only one event can occur in a block at a time. For example, an ECC block that is in the NE state must either stay in the NE state or switch to the SE state. The probability of an ECC block, in the NE state, switching to the UE state is 0 because that would require two errors to occur in one time step.

This section describes the basic Markov models that we are using in our analysis of the ECC block reliability. To complete our model, the transition probabilities associated with each transition arc are defined and combined into transition matrices. It is these transition matrices that allow us to calculate the reliability of ECC blocks in chapter 6.

5.4.1 Our Basic Markov Models

In our analysis of the ECC block reliability we use Markov models to describe all of the relevant processes and states. The soft error rate (λ) and the access rates for the favored memory (γ_f) and the three variations of ignored memory (γ_{i0}, γ_{i1}, γ_{i2}) are used to determine the probabilities that the ECC block will change states over time.

Figure 5.4 is a Markov model that we use for an ECC block that tolerates a single error and does not have an error caused by a permanent fault. There are three states which a block may be in: no error (NE), single error (SE) and uncorrectable error (UE). When a block is in the NE state all of the bits contain correct data. A block in the SE state has a single bit containing a soft error. And finally a block in the UE state has more than one bit containing an error and the ECC circuitry is no longer able to correct the errors in that block of data.

The transition arcs identify the possible changes of state that can occur and their corresponding probabilities. These transition probabilities are stated in terms of a fixed time increment, t, and are assumed to remain constant throughout the term of the analysis. For our analysis we use a time increment of 1 nanosecond (ns). To simplify the transition probability expressions in the transition matrix we drop the time increment, t, because it is always 1ns.

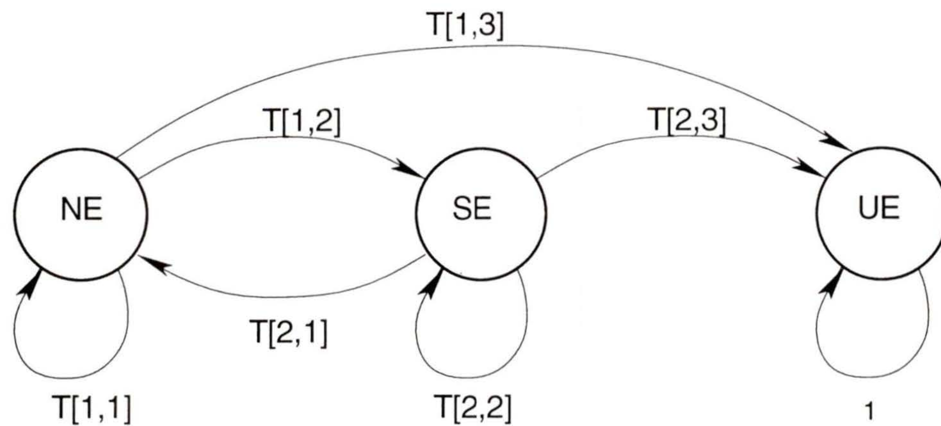


Figure 5.4   Markov model for a Fault Free ECC block .

The labels on the transition arcs (T[1,1], T[1,2], ...) identify addresses of the transition probability within the transition matrix (see chapter 3). Thus, as we explain the transition probabilities that we are using, the transition matrix is defined. The transition arcs that would take the system from the UE state to the SE or NE state are not shown because once the block enters the UE state there is no possibility for it to change states further (unless the system is shut down). Thus, the uncorrectable error (UE) state is called an absorbing state and at some point in time the system will be in that state.

Figure 5.5 is the Markov model that we use to describe an ECC block that contains a permanent fault that causes an error in the data. When the only erroneous bit in the block is the bit affected by the permanent fault then the block is in the PF (permanent fault) state. When any other bit contains a soft error then the block is in the UE (uncorrectable error) state. Here again there is no possibility of a block changing from the UE state (absorbing) to the PF state.

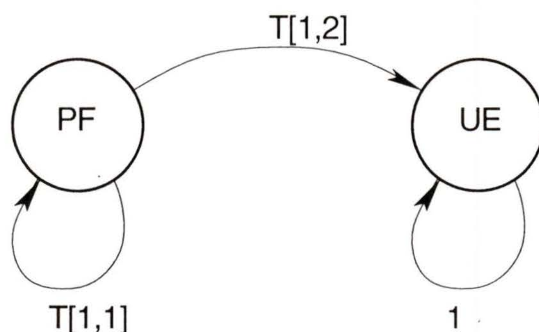


Figure 5.5 Markov model for a SEC block with a permanent fault.

Note that as a simplification we use the model in figure 5.4 for an approximation of an ECC block with a hidden fault even though a hidden fault means that only 136 bits are susceptible to soft errors.

5.4.2 The Transition Probability Matrices

The probabilities defined in the following sections refer to the transition arcs in the various Markov models. We define the following transition matrices which we use for our analysis of the various Markov models:

T_u	Uniform access ECC block
T_f	Favored access ECC block
T_{i0}	Ignored ECC block using scheme 0
T_{i1}	Ignored ECC block using scheme 1
T_{i2}	Ignored ECC block using scheme 2
T_p	ECC block containing a permanent fault.

5.4.3 ECC Blocks in the NE state

T[1,1]

$T_u[1,1]$ is the probability that a block in the NE state at the beginning of the time increment will be in the NE state at the end of the time increment. Using the definition of reliability for a system with a constant failure rate as discussed in chapter 3, and the probability union axiom (assuming independence) we have:

$$T_u[1,1] = (e^{-\lambda})^{137} = e^{-137\lambda}$$

which is the probability that all bits (137 of them) will remain error free through the time increment. Each of the other transition matrices have the same value in this position because the conditions we are examining do not effect the rate of soft errors. Thus:

$$T_u[1,1] = T_f[1,1] = T_{i0}[1,1] = T_{i1}[1,1] = T_{i2}[1,1]$$

T[1,2]

$T_u[1,2]$ is the probability that a block in the NE state at the beginning of the time increment will switch to the SE state during the time increment (because of a soft error). Because there are only two possible outcomes this probability is the compliment of $T_u[1,1]$. Giving us:

$$T_u[1,2] = 1-T_u[1,1] = 1- e^{-137\lambda},$$

and

$$T_u[1,2] = T_f[1,2] = T_{i0}[1,2] = T_{i1}[1,2] = T_{i2}[1,2].$$

T[1,3]

$T_u[1,3]$ indicates the probability of a transition of an ECC block from the NE state to the UE state. We assume a probability of zero for this because of our assumption that the system is discrete and only one event occurs at a time. The error that is introduced by this assumption is near λ^2 or approximately: 10^{-42}. Thus:

$$T_u[1,3] = 0$$

and

$$T_u[1,3] = T_f[1,3] = T_{i0}[1,3] = T_{i1}[1,3] = T_{i2}[1,3].$$

5.4.4 ECC Blocks in the SE State

When an ECC block has a single error there are three possible states that it may be in after the next time step, NE, SE, and UE. We consider the transition probability from SE to UE first because it is the least significant probability and thus would have the smallest effect on the other probabilities.

T[2,3]

$T_u[2,3]$ contains the probability that a block in the SE state suffers a second soft error in the next time increment. This probability is almost the same as T[1,2], except that there are now only 136 bits that are susceptible to soft errors. Using the definition of unreliability given in chapter 3 we have:

$$T_u[2,3] = 1- e^{-136\lambda}.$$

Again, because the soft error rate is not being modified in our different schemes the other transition probabilities are equal:

$$T_u[2,3] = T_f[2,3] = T_{i0}[2,3] = T_{i1}[2,3] = T_{i2}[2,3].$$

T[2,2]

$T_u[2,2]$ is the probability that an ECC block that has one soft error is not scrubbed and does not suffer another soft error in the next time step. Given the rate of scrubbing for a block with uniform access is γ_u, we have the probability of not scrubbing as $e^{-\gamma_u}$. The probability that no soft error affects the block is now $e^{-136\lambda}$. The union of these two independent probabilities gives us:

$$T_u[2,2] = e^{-\gamma_u} \times e^{-136\lambda}$$

$$= e^{-\gamma_u - 136\lambda}$$

As the scrubbing rate changes in each of the schemes we are examining the probability of remaining in the SE state changes. Thus we have:

$$T_f[2,2] = e^{-\gamma_f - 136\lambda}$$

$$T_{i0}[2,2] = e^{-\gamma_{i0} - 136\lambda}$$

$$T_{i1}[2,2] = e^{-\gamma_{i1} - 136\lambda}$$

$$T_{i2}[2,2] = e^{-\gamma_{i2} - 136\lambda}$$

T[2,1]

$T_u[2,1]$ is the probability that a block in the SE state will be scrubbed and returned to the NE state. The probability of scrubbing is $1-e^{-\gamma_u}$, and the probability that the block did not already switch to the UE state is $e^{-136\lambda}$. Thus we have:

$$T_u[2,1] = (1-e^{-\gamma_u}) \times e^{-136\lambda}$$

Again we consider the variations in the scrubbing rate to determine the transitions in each of the other schemes:

$$T_f[2,1] = (1-e^{-\gamma_f}) \times e^{-136\lambda}$$

$$T_{i0}[2,1] = (1-e^{-\gamma_{i0}}) \times e^{-136\lambda}$$

$$T_{i1}[2,1] = (1-e^{-\gamma_{i1}}) \times e^{-136\lambda}$$

$$T_{i2}[2,1] = (1-e^{-\gamma_{i2}}) \times e^{-136\lambda}$$

5.4.5 ECC Blocks in the UE state

Once a block has an uncorrectable error it will remain in that state until the system is reset. This is called an absorbing state, and the probabilities associated with it are:

$$T_u[3,1] = T_f[3,1] = T_{i0}[3,1] = T_{i1}[3,1] = T_{i2}[3,1] = 0$$

$$T_u[3,2] = T_f[3,2] = T_{i0}[3,2] = T_{i1}[3,2] = T_{i2}[3,2] = 0.$$

$$T_u[3,3] = T_f[3,3] = T_{i0}[3,3] = T_{i1}[3,3] = T_{i2}[3,3] = 1.$$

5.4.6 Transition Matrices for Working ECC Blocks

For the uniform access memory we have the following transition matrix:

$$\mathbf{T_u} = \begin{bmatrix} e^{-137\lambda} & 1-e^{-137\lambda} & 0 \\ (1-e^{-\gamma_u})e^{-136\lambda} & e^{-\gamma_u-136\lambda} & 1-e^{-136\lambda} \\ 0 & 0 & 1 \end{bmatrix}$$

Which approximates to:

$$\mathbf{T_u} \approx \begin{bmatrix} 0.9_{18}863 & 1.37\text{x}10^{-19} & 0 \\ 1.526\text{x}10^{-7} & 0.9_{6}8474 & 1.36\text{x}10^{-19} \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly for the favored memory we have:

$$\mathbf{T_f} = \begin{bmatrix} e^{-137\lambda} & 1-e^{-137\lambda} & 0 \\ (1-e^{-\gamma_f})e^{-136\lambda} & e^{-\gamma_f-136\lambda} & 1-e^{-136\lambda} \\ 0 & 0 & 1 \end{bmatrix}$$

Which approximates to:

$$\mathbf{T_f} \approx \begin{bmatrix} 0.9_{18}863 & 1.37\text{x}10^{-19} & 0 \\ 4.8828\text{x}10^{-6} & 0.9_{5}51172 & 1.36\text{x}10^{-19} \\ 0 & 0 & 1 \end{bmatrix}$$

Finally, for the three models we use for ignored memory we have:

$$\mathbf{T_{i0}} = \begin{bmatrix} e^{-137\lambda} & 1-e^{-137\lambda} & 0 \\ (1-e^{-\gamma_{i0}})e^{-136\lambda} & e^{-\gamma_{i0}-136\lambda} & 1-e^{-136\lambda} \\ 0 & 0 & 1 \end{bmatrix}$$

which approximates to:

$$\mathbf{T_{i0}} \approx \begin{bmatrix} 0.9_{18}863 & 1.37\text{x}10^{-19} & 0 \\ 1\text{x}10^{-11} & 0.9_{11} & 1.36\text{x}10^{-19} \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$\mathbf{T_{i1}} = \begin{bmatrix} e^{-137\lambda} & 1-e^{-137\lambda} & 0 \\ (1-e^{-\gamma_{i1}})e^{-136\lambda} & e^{-\gamma_{i1}-136\lambda} & 1-e^{-136\lambda} \\ 0 & 0 & 1 \end{bmatrix}$$

which approximates to:

$$\mathbf{T_{i1}} \approx \begin{bmatrix} 0.9_{18}863 & 1.37\text{x}10^{-19} & 0 \\ 1.6\text{x}10^{-10} & 0.9_{9}84 & 1.36\text{x}10^{-19} \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{T_{i2}} = \begin{bmatrix} e^{-137\lambda} & 1-e^{-137\lambda} & 0 \\ (1-e^{-\gamma_{i2}})e^{-136\lambda} & e^{-\gamma_{i2}-136\lambda} & 1-e^{-136\lambda} \\ 0 & 0 & 1 \end{bmatrix}$$

which approximates to:

$$\mathbf{T_{i2}} \approx \begin{bmatrix} 0.9_{18}863 & 1.37\text{x}10^{-19} & 0 \\ 2.4414\text{x}10^{-6} & 0.9_{5}75586 & 1.36\text{x}10^{-19} \\ 0 & 0 & 1 \end{bmatrix}$$

5.4.7 Transition Matrix for a Faulty ECC block

The last transition matrix that we define covers the blocks that have a single error caused by a permanent fault (T_p). There are only a small number of these blocks but they are a significant source of UE's because they do not have the error correcting capacity of the other blocks.

$T_p[1,1]$ is the probability that a block in the PF state will stay in the PF state during the next segment of time. This is:

$$T_p[1,1] \quad = \quad (e^{-\lambda})^{136}$$

$T_p[1,2]$ is the probability that a block in the PF state experiences one or more soft errors, causing it to be in the UE state. We recognize that this is the complement of the probability that the block remains in the PF state. This gives us:

$$T_p[1,2] \quad = \quad 1 - e^{-136\lambda}$$

Once a block is in the UE state it remains there, thus we have:

$$T_p[2,1] \quad = \quad 0$$

$$T_p[2,2] \quad = \quad 1$$

For a faulty ECC block then, we have:

$$\mathbf{T_p} = \begin{bmatrix} e^{-136\lambda} & 1-e^{-136\lambda} \\ 0 & 1 \end{bmatrix}$$

Which approximates to:

$$\mathbf{T_p} = \begin{bmatrix} 0.9_{18}864 & 1.36*10^{-19} \\ 0 & 1 \end{bmatrix}.$$

5.5 Summary

In this chapter we describe three scrubbing schemes: scheme 0 which is common in new DRAM chips, and schemes 1 and 2 which are enhancements. We estimate a set of parameters for each scheme which we use as a basis for determining our transition probabilities both in this chapter and for completing our analysis in chapter 6. Markov models are developed for use in our analysis, one for a normal ECC block and another for an ECC block containing a permanent fault causing an error. From these models, we create transition matrices which define the Markov models for each of the types of ECC blocks that we are considering in our analysis.

Chapter 6 Analysis and Results

This chapter brings together all of the work of the previous chapters to analyze the potential of two new schemes for improving the reliability of DRAM chips. We begin by describing the analysis platform, parameter choices, method of analysis, definitions, and output format. The remainder of the chapter describes the analysis process.

Our initial analysis considers individual ECC blocks and the probabilities associated with each state in our model over time. The following section compares the different types of ECC blocks and their reliabilities based on the parameters described in chapter 5. Finally, we calculate and compare the reliabilities of each of the schemes at the chip level.

6.1 Basis of Analysis

6.1.1 Software used for Our Analysis

In order to efficiently analyze the reliability of ECC blocks and DRAM chips over extended periods of time we needed a relatively powerful mathematical tool. We chose the Maple V software from the University of Waterloo because of its flexibility and selectable accuracy [Cha92, Cha91a, Cha91b].

Maple is able to do calculations using exact numbers with no error, until it runs out of memory space or the results exceed Maple's limits. The transition probabilities which we are dealing with exceed these limits, thus we must use floating point operations. Fortunately Maple allows us to select the number of digits we need. We use 100 digits for the matrix multiplications and note that the error did not extend beyond the 88th digit during our analysis.

6.1.2 Time, the Independent Variable

The reliability of DRAM chips over time is obviously going to decrease, and eventually all chips fail (see chapter 5). We are considering the reliability of a DRAM chip over the time range of 2 to 2^{60} nanoseconds because this effectively gives us a finite view of the lifetime of a chip. 2^{60} ns ($\approx 10^{18}$) works out to approximately 36.56 years which, for our purposes, is the end of the useful life of any chip. 2^{50} ns ($\approx 10^{15}$) is approximately 13 days which might be the duration of some critical applications. 2^{30} ns ($\approx 10^{9}$) is approximately 1 second.

For times of less than 1 second, the state probabilities show very low probabilities of UE's, but the results are useful in identifying the differences between the schemes that we are evaluating. Recall that we are using a discrete model to approximate the behavior of a chip, and that the access cycle of our model chip is 50 ns.

Because the probabilities that we are dealing with are generally very small or very close to 1 the vertical axis needs to be plotted with the log of the probability (varying from 10^{-38} to 10^{0}) to bring out the changes that occur. The time range is 2 ns to 36 years, but most of the dramatic changes occur in less than a day. To allow these changes in probability to be observed we plot the log of the time for the horizontal axis.

6.1.3 Calculating the State Probabilities

As described in chapter 3, an ECC block must be in either the no error, the single error, or the uncorrectable error states (NE, SE, or UE). The probabilities associated with these states after the next time increment can be determined by multiplying the current state vector by the transition matrix. To calculate the state probabilities over an arbitrary number of time increments we must multiply the state vector by the transition matrix once for each time step. Thus to find the state vector, **P**, after n time steps we have:

$$\mathbf{P_n} = \mathbf{P_0} \cdot \mathbf{T^n}$$

If we assume that the initial state vector is:

$$\mathbf{P_0} = [\text{NE, SE, UE}] = [\ 1, 0, 0\ ]$$

then we see that the state vector after n time steps is just the first row of the transition matrix after it has been raised to the power n.

6.1.4 Method of Analysis

Multiplying two 3x3 matrices is very simple in Maple, but to do it 10^{18} times is impractical. A test of 1000 3x3 matrix multiplications took more that 65 seconds. Extrapolating this result we find it would take 2×10^{11} years to complete - far too long to stay in grad school. In addition, the error that we have to deal with increases with the number of operations performed.

To overcome these difficulties we choose to multiply the transition matrix by itself repeatedly. The resulting powers of the transition matrix which we calculate are for time step equal to 2, 4, 8, 16 ... 2^{60} ns. This approach limits the calculations required for each of the types of ECC blocks to 60 matrix multiplications. An important side benefit with this set of time values is that we have a clear view of the development of the probabilities which change rapidly in the early stages of analysis and then slow down over the long term. The data points generated are ideal for our log-log graphs.

To collect our data for analysis we save the top row of the resulting transition matrix each time we multiply the transition matrix by itself. This table of probabilities is used to produce plots of the probability of a single error and an uncorrectable error in an ECC block over time (see section 6.2 and 6.3). Finally, these ECC block probabilities are used for calculating and plotting the probabilities of an UE in a DRAM chip over time for each of the different schemes (see section 6.4).

6.1.5 Terms and Definitions

We remind the reader of the discussion in chapter 4 regarding terminology. Specifically, we use the terms *unreliability* and *probability of an UE* synonymously for ECC blocks and chips. We define the following functions for use in our explanations:

$R_{i0}(t)$	reliability of an ignored ECC block in scheme 0
$R_{i1}(t)$	reliability of an ignored ECC block in scheme 1
$R_{i2}(t)$	reliability of an ignored ECC block in scheme 2
$R_{f}(t)$	reliability of a favored ECC block in all schemes
$R_{u}(t)$	reliability of an uniformly accessed ECC block
$R_{pf}(t)$	reliability of an ECC block with a permanent fault
$R_{C0}(t)$	reliability of a chip using scheme 0
$R_{C1}(t)$	reliability of a chip using scheme 1
$R_{C2}(t)$	reliability of a chip using scheme 2

6.2 Analysis

6.2.1 Analysis of an ECC Block in the NE State.

The probability that an ECC block remains in the NE state is very close to 1 throughout the range of time that we are considering. A plot of these probabilities is not very exciting and is essentially redundant as they can easily be derived from the plots of the SE and UE state probabilities. For these reasons we do not consider probabilities associated with the NE state directly.

6.2.2 Analysis of an ECC Block in the UE State

We begin our analysis by considering the general trends in the changing probability of an UE in an ECC block. These values are found in the T[1,3] position of the transition matrix after each successive doubling. We assume that the memory is being continuously randomly accessed with a uniform coverage resulting in a block scrubbing rate of 1.526×10^{-7} accesses per second.

Figure 6.1 shows us three curves indicating the probabilities of an UE existing in an ECC block over time. Each curve corresponds to a different soft error rate (SER). The highest curve represents an ECC block subject to a SER of 10^{-18} failures/ns, a very high error rate that would not be acceptable in commercial products today. The lowest curve shows the probabilities associated with a SER of 10^{-24} failures/ns which is better than any of the devices considered in chapter 2. Finally, the middle curve represents the probabilities found when the SER is 10^{-21} failures/ns, which is a typical value and we use it for the remainder of our analysis (see section 3.2.3).

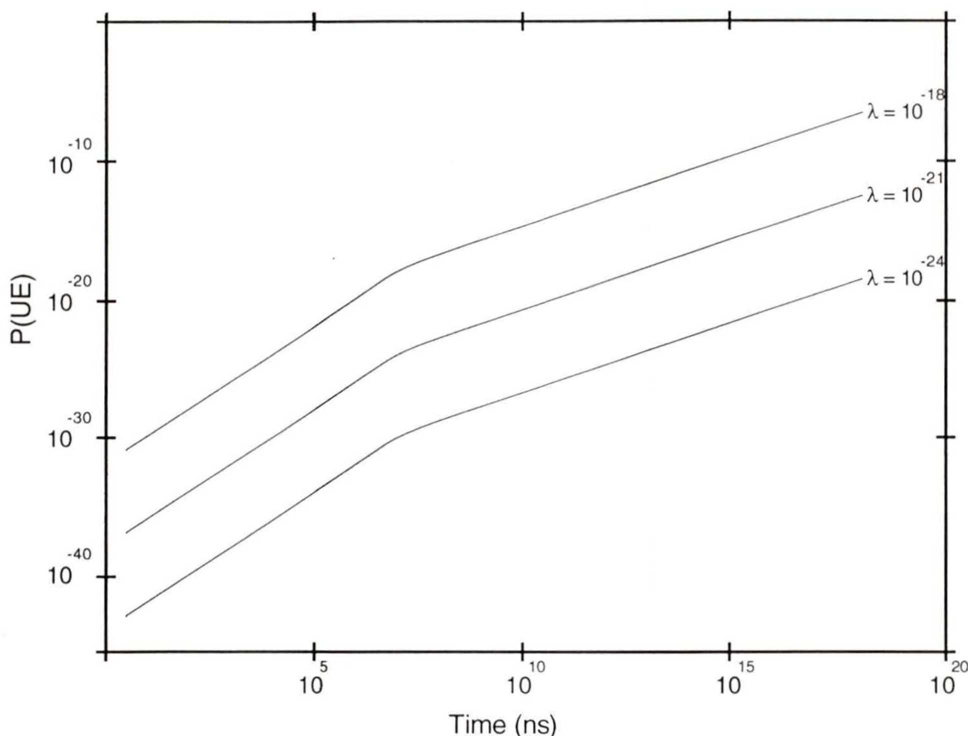


Figure 6.1 Probability of an UE in an ECC block .

As expected, we see that the higher the soft error rate is, the higher the probability of an UE. The probability that an ECC block in either the NE or the SE states experiences a soft error in a time increment is related to the SER and the number of bits. We can't directly calculate the probability that an UE occurs from the SER (λ) because the probability of scrubbing (γ) is also a factor.

When the values of λt are small we see that $1-e^{\lambda t}$ is approximately equal to λt. By ignoring the effect of scrubbing in the early stages, and using this approximation we estimate:

$$P(UE) \approx (1-e^{\lambda t})(1-e^{\lambda t})$$

$$\approx (\lambda t)^2$$

From this the factor between the probability curves is estimated as $(10^3)^2$ or 10^6, this matches the results from our analysis.

Even when the effect of scrubbing causes the occurrence rate of UE's to slow down (see section 6.2.3) we find that the factor between the probability curves remains constant. After 2^{60} nanoseconds, the limit of the time domain we are considering, we have the following probabilities of an UE:

$1.407682409067 \times 10^{-6}$ for $\lambda = 10^{-18}$

$1.407682510661 \times 10^{-12}$ for $\lambda = 10^{-21}$

$1.407682510664 \times 10^{-18}$ for $\lambda = 10^{-24}$

which demonstrates that the initial factor between the curves remains close to constant until the number of error free ECC blocks is exhausted.

Considering our basic ECC block with $\lambda = 10^{-21}$ and $t=2^{50}$ ns (approximately 13 days) we see:

$$P(UE) \approx 1.4\text{x}10^{-16}.$$

Finding the complement gives us the block reliability for this time period:

$$R_u(2^{50}) \approx 0.9_{15}86.$$

From this result a single ECC block seems to have a high reliability, but we must remember that there are 131,072 blocks in our basic 16 Mbit chip and they must all be free of UE's for the chip to remain reliable. Also, we have not yet considered the effects of access distribution.

The slopes of the probability curves change dramatically when the probability of a new soft error occurring in an ECC block with no errors (NE state) equals the probability of an existing soft error (SE state) being scrubbed. We investigate this effect in the next section.

6.2.3 Analysis of an ECC Block in the SE State

Figure 6.2 illustrates the SE state probability curves for ECC blocks that are part of the ignored memory in each of the given schemes. The curves share the same initial probabilities because the overriding factors are the soft error rates acting on the probabilities that a block is in the NE or SE states causing them to change to the SE and UE states respectively.

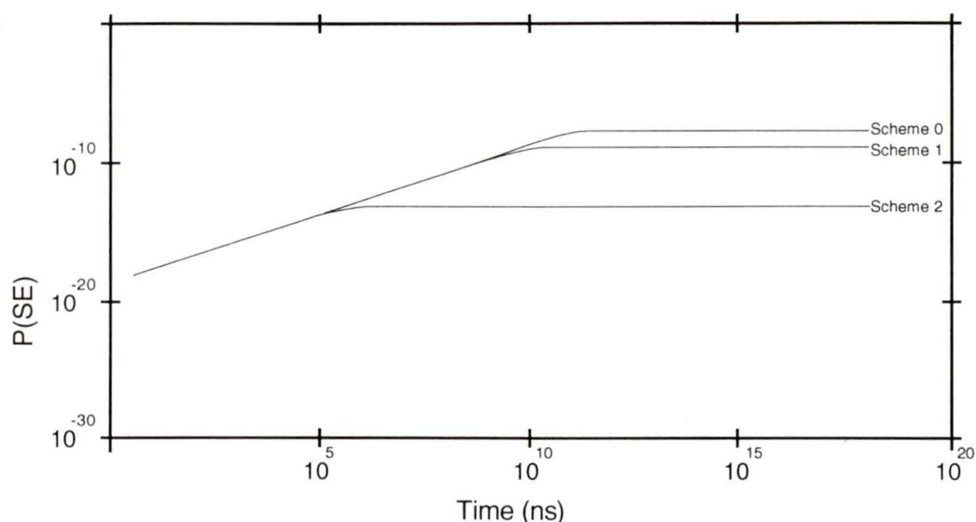


Figure 6.2 Probability of a SE in ignored ECC blocks

When the probability that a block in the SE state is scrubbed, matches the probability of a soft error affecting a block in the NE state, an equilibrium of sorts occurs. Because the SER (λ) is constant we can find the relationship between the scrubbing rate and the equilibrium point by reducing the following equilibrium relationship.

$$T[1,2]*P(NE) = T[2,1]*P(SE)$$

By solving for P(SE) and approximating P(NE) by 1 we get:

$$P(SE) \approx \frac{1-e^{-137\lambda t}}{(1-e^{-\gamma t})^{*}\ e^{-136\lambda t}}$$

which, when γt and λt are small, approximates to:

$$P(SE) \approx \frac{137\lambda t}{(\gamma t)(e^{-136\lambda t})}$$

Finally, by approximating ($e^{-136\lambda t}$) by 1 and cancelling the t's, we see that the equilibrium probability of a single error existing is inversely related to the scrubbing rate (γ):

$$P(SE) \approx \frac{137\lambda}{\gamma}$$

From this point on, both the NE and the SE state probabilities begin to decay very slowly because some blocks in the SE state switch to the UE state as they are affected by transient faults.

6.3 Unreliability of ECC Blocks

Figure 6.3 summarizes the probabilities of UE's developing in ECC blocks (unreliability) as they change over time for each of the types of ECC blocks that are discussed in chapter 5. We have included the curve for an ECC block with an uniform access rate for comparison purposes. From these block probabilities we calculate the probability that a chip contains an UE later in the chapter.

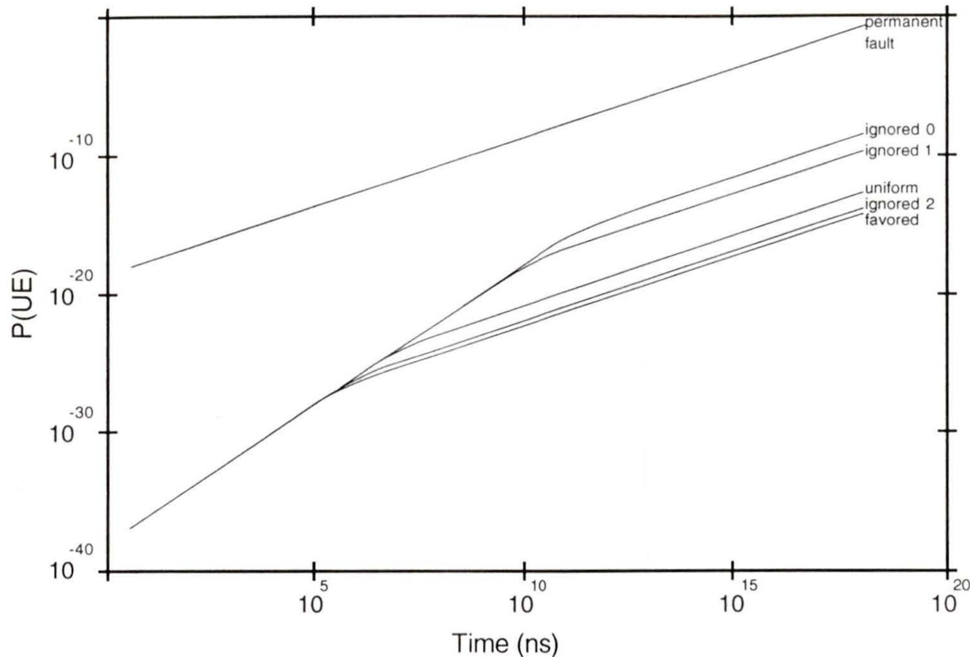


Figure 6.3 Probability of an UE for each ECC block type.

When a block with a permanent fault is affected by a soft error then an UE exists in that block. The probability of an UE in a block containing a permanent fault can thus be determined simply by determining the probability of a soft error occurring:

$$P(\text{UE in block with perm. fault}) = 1-e^{-136\lambda t} \approx 136\lambda t$$

The slopes of all of the other curves representing fault free ECC blocks are consistent. The consistency can be explained by considering that the probability of an UE in an ECC is related to the probability of a SE in an ECC block and the SER. As we discussed in section 6.2.3 the probability of a SE increases (essentially directly related to λ) until an equilibrium point is reached and then it begins to decay very slowly. Thus the slope of the graph of the probability of an UE is related to λ^2 until the equilibrium point for the probability of a SE is reached, then it is related to λ.

At 2^{50} ns (10^{15}) we see the following probabilities of UE's:

$$1-R_{pf}(2^{50}) = 1.5311x10^{-4}$$

$$1-R_{i0}(2^{50}) = 2.0975x10^{-9}$$

$$1-R_{i1}(2^{50}) = 1.3111x10^{-11}$$

$$1-R_{i2}(2^{50}) = 8.5925x10^{-16}$$

$$1-R_{u}(2^{50}) = 1.3747x10^{-14}$$

$$1-R_{f}(2^{50}) = 4.2963x10^{-18}$$

The factor we find between the probability that an UE will be present in the ECC block with a permanent fault ($1.5311x10^{-4}$), and the probability of an UE in the ECC block in the favored memory ($4.2963x10^{-18}$) is $3.56x10^{13}$. This indicates that the permanent faults have an overwhelming effect on the reliability of a chip (see section 6.4).

The lowest curve in figure 6.3 represents the probability of an UE in a favored block of memory. As expected, favored ECC blocks are the least likely to suffer an UE because they are scrubbed more often.

The remaining curves show the probabilities for the ignored ECC blocks in the three different scrubbing schemes (ignored 0, ignored 1 and ignored 2) as well as for an ECC block with uniform access. Here again, the ignored memory in scheme 0 has the highest probability of UE and the ignored memory in scheme 2 has the lowest probability of UE. The only differences between the schemes in this illustration is the access or scrubbing rates that are assumed.

The upper and lower bounds for the probability of an UE in an ECC block could be found by assuming only a single block is accessed all the time, thereby making the scrubbing rate equal to the chip access rate for the one favored block and 0 for the other blocks. We have not included these degenerate cases in figure 6.3 for simplicity purposes.

6.4 Analysis of the Reliability of DRAM Chips

6.4.1 Chips Containing Permanent Faults

From the probabilities for UE's in ECC blocks we are able to determine the probability of an UE in a chip. In order for the chip to be free from UE's each of the ECC blocks within it must be UE free. The series model discussed in section 3.3.1 shows us that the product of the reliabilities of each of the ECC blocks will give us the reliability of the chip.

We begin by considering the probability of an UE in a chip with uniform access, and varying numbers of permanent faults. Using the formula given in section 4.4.3, and recalling the following definitions,

$R_u(t)$ probability of no UE in a chip with uniform access,

$R_{pf}(t)$ as the probability of no UE in an ECC block with a permanent fault, and,

p as the number of permanent faults on the chip,

we have:

$$R_{chip}(t) = R_u(t)^{131072-p} \times R_{pf}(t)^p.$$

Figure 6.4 shows the results of this analysis with 0,1,2 and 8 permanent faults. We see that highest probability of an UE occurs when there are 8 permanent faults and the lowest occurs when there are 0 permanent faults. We notice that even a single permanent fault severely affects the reliability of the chip.

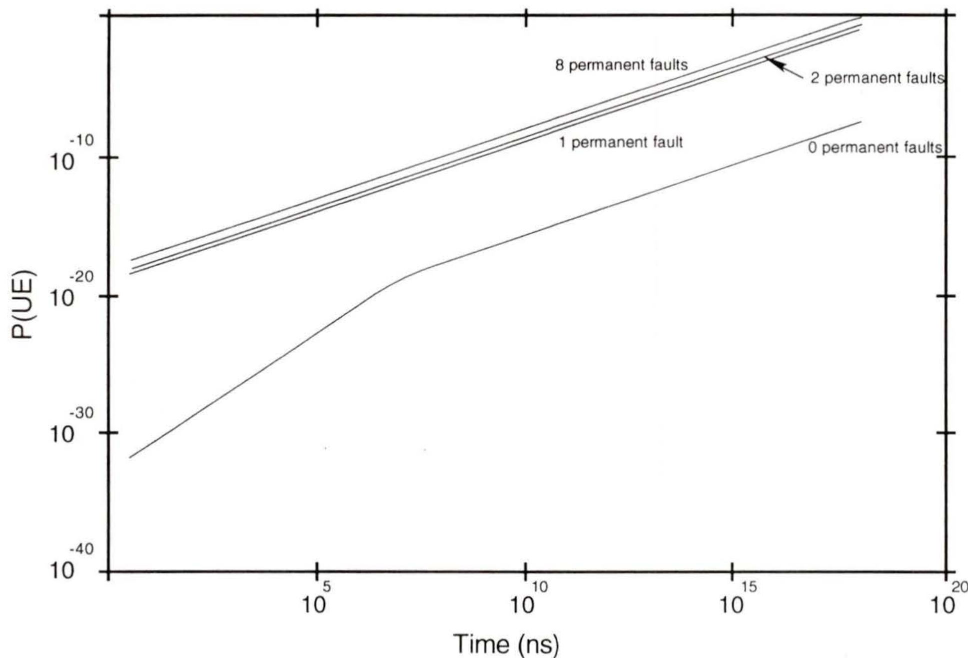


Figure 6.4 Probability of an UE in a DRAM chip.

From this result we recognize that the overwhelming factor in the reliability of chip is the reliability of the ECC blocks containing errors caused by permanent faults. The effects of the different scrubbing schemes are several orders of magnitude less than the effects of the permanent faults.

In order to effectively compare the differences between the different scrubbing schemes proposed we assume that the chips have no permanent faults. This is not the general case, but a significant portion of the DRAM chips produced are error free and we shift our focus to these chips.

6.4.2 Chips with No Permanent Faults

We have estimated the probability of an UE (and thus the reliability) for each of the types of ECC blocks described in chapter 5. By using the basic formula described in section 4.5.3, and assuming 0 permanent faults, the reliabilities of the chips using each of our schemes are defined as follows:

$$R_{C0}(t) = (R_{i0}(t))^{126976} \times (R_f(t))^{4096}$$

$$R_{C1}(t) = (R_{i1}(t))^{65536} \times (R_f(t))^{65536}$$

$$R_{C2}(t) = (R_{i2}(t))^{126976} \times (R_f(t))^{4096}$$

Again we choose to plot the unreliability (probability of an UE) of the chip for each of the schemes because the reliability values are all very close to 1. Figure 6.5 shows the results of these calculations.

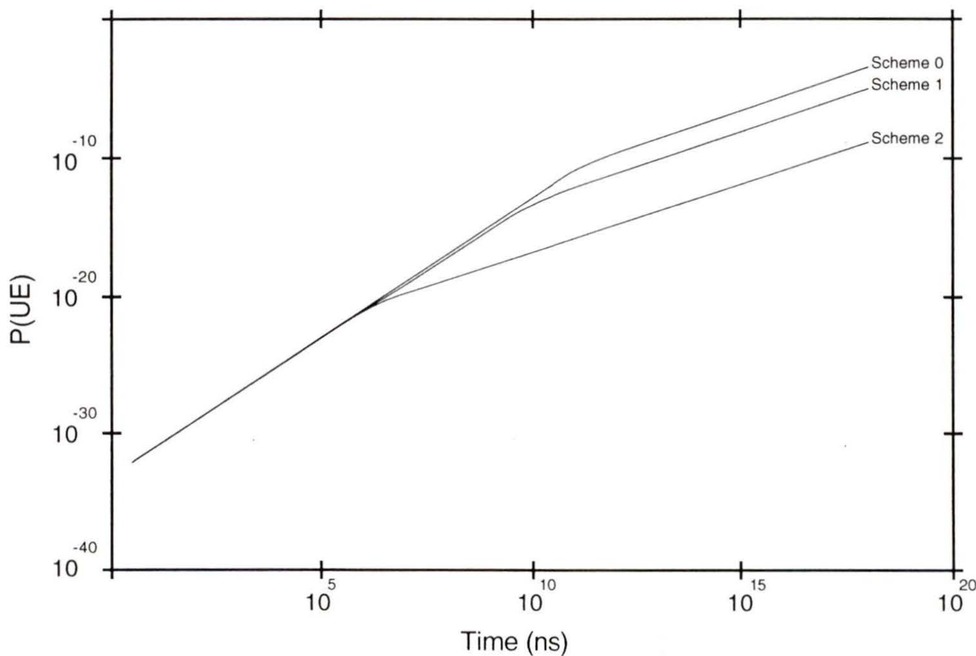


Figure 6.5 P(UE) in a chip containing zero permanent faults

For the chip using scheme 0 we see the highest probability of an UE. This is expected because the reliability of the ECC blocks in the ignored memory in scheme 0 is the poorest. With scheme 1 we can see some improvement because more of the blocks (65536 instead of 8192) are scrubbed at the favored access rate and the scrubbing rate of the ignored blocks has also been increased by a factor of 16. Finally, scheme 2 shows the best results because all of the blocks are being scrubbed at a very high rate.

6.4.3 Differences in Reliability

At $t=2^{30}$ ns (1 second), we have:

scheme 0: $P(UE) = 1.358 \times 10^{-15}$ $R_{C0}(t) = 0.9_{14}8642$

scheme 1: $P(UE) = 6.653 \times 10^{-16}$ $R_{C1}(t) = 0.9_{15}3347$

scheme 2: $P(UE) = 1.057 \times 10^{-18}$ $R_{C2}(t) = 0.9_{17}8943$

At $t=2^{50}$ ns (13 days), we have:

scheme 0: $P(UE) = 2.661 \times 10^{-7}$ $R_{C0}(t) = 0.9_{6}7339$

scheme 1: $P(UE) = 8.590 \times 10^{-9}$ $R_{C1}(t) = 0.9_{8}1410$

scheme 2: $P(UE) = 1.109 \times 10^{-12}$ $R_{C2}(t) = 0.9_{11}8891$

At $t=2^{60}$ ns (36.6 years), we have:

scheme 0: $P(UE) = 2.729 \times 10^{-4}$ $R_{C0}(t) = 0.9_{3}7271$

scheme 1: $P(UE) = 8.790 \times 10^{-6}$ $R_{C1}(t) = 0.9_{5}1210$

scheme 2: $P(UE) = 1.135 \times 10^{-9}$ $R_{C2}(t) = 0.9_{8}8865$

As we can see in figure 6.5 the factor between the probability of an UE in each of the schemes stabilizes after about 10^{12} nanoseconds. From the values above we see that a chip utilizing scheme 1 is approximately 31 times less likely to have an UE than the scheme 0 chip (standard chip). This factor is caused by a 16 times increase in scrubbing rate of the ignored memory and the scrubbing of half of the ECC blocks at the favored memory rate.

The chip using scheme 2 has an even more impressive improvement factor of 240,232 over the scheme 0 chip. This improvement is caused by increasing the rate of access of the ignored memory by 244,140 times.

From these results we see a very strong relationship (direct) between the improvement in scrubbing and the reduction in the probability of an UE in a chip.

6.5 Summary

This chapter presents our analysis of the reliability of DRAM chips. We analyze the reliability of individual ECC blocks using the Markov model that is developed in chapter 5. Then, the resulting probabilities are combined using the series model developed in chapter 4 to determine the reliability of the DRAM chips utilizing our different scrubbing schemes..

We first consider the effects of the soft error rate on the probability of an UE in an ECC block. Here we find that the improvement in the unreliability is related to the square of the improvement to the soft error rate within the time domain we are considering.

The change in slope of the probability curves is explained by examining the probability that an ECC block is in the SE state. We show that the probability of a single error stabilized (in the time range we are considering) and that there is an inverse relationship between the scrubbing rate (γ) and this stabilization probability.

The probabilities of UE's in each of the types of ECC blocks discussed are plotted for comparison. As expected the ECC blocks that are scrubbed more frequently have a lower probability of being in the UE state (the higher reliability). An ECC block with a permanent fault is 10^{13} times more likely to suffer an UE than an ECC block in favored memory.

Next we calculate and plot the probability of an UE in a chip with varying numbers of permanent faults and find that even a single permanent fault would have an overriding effect. From this result we recognize that our schemes would have a negligible effect in chips containing permanent faults.

Finally we compare the probability curves for each of our schemes on DRAM chips with no permanent faults. We find that a chip using scheme 1 is 31 times more reliable than a similar chip using scheme 0, and a chip using scheme 2 is 240232 times more reliable than the scheme 0 chip.

In summary, if a chip has a permanent fault then the schemes proposed will not have a significant affect on the reliability of the chip. However, a significant portion of DRAM chips produced will not have any permanent faults. These select chips can have their reliabilities significantly improved with the implementation of either scheme 1 or scheme 2.

Chapter 7 Conclusions

7.1 Overview of Development

In chapter 1 we discover that up to 98% of failures that occur in DRAM chips during normal operation are caused by transient faults [Maz88, CH84, Sai82]. We discuss the use of error correction schemes in DRAM chips to improve yield and reliability and note that research efforts have generally been focussed on analyzing the effect on yield. From this research we determined that a method of estimating the reliability of DRAM chips is needed and that enhancements to the standard error correction schemes used in these chips is possible.

Chapter 2 reviews the architecture of memory, and describes permanent, intermittent and transient faults. We focus on the causes of transient faults in DRAM chips and some of the techniques used to provide fault tolerance. Finally we explain the use and limitations of error correction codes (ECC's) that are built into DRAM chips and adopt the industry standard single error correction-double error detection scheme (SEC/DED) for use in our analysis.

The basic mathematics of probability and reliability used in our analysis is covered in chapter 3. We describe our assumptions about the random uniform distribution of α-particles (which cause transient faults) and estimate a constant failure rate (or soft error rate) for use in our analysis. Models for determining the reliability of complex systems are examined. We choose to use Markov models to estimate the reliability of ECC blocks and a series system model to derive the reliability of DRAM chips from the block results.

In chapter 4 we point out that the term, *probability of an uncorrectable error*, can be used synonymously with *unreliability* when discussing bits and ECC blocks, and extend this meaning to DRAM chips. We demonstrate (with the birthday problem) that a DRAM chip will relatively quickly become unreliable over time if the errors caused by transient faults are not removed (or scrubbed) as they develop. A typical DRAM chip is chosen to use as the basis for our analysis and the necessary parameters are identified.

The reliability of a bit is derived using our assumption that transient faults are randomly distributed. For ECC blocks we explain that the factors affecting the reliability make analysis difficult, and point to our use of Markov models in chapter 5 and 6 to estimate the reliability. Our model for estimating the reliability of a DRAM chip is expressed in a simple way and then adjusted to accommodate the effects of permanent faults, scrubbing, and access distribution.

We show that a permanent fault in an ECC block may use up the single error correction coverage for that block and that the reliability of such affected blocks is even lower than an equivalent block of data with no error correction. We adopt a distribution of access for use in our analysis that assumes a small portion of memory is heavily used, and the remainder is seldom used. Although this distribution is very simple, it is sufficient to demonstrate that our model can be adjusted to accommodate many other distributions. We also show that the assumption of uniform access would significantly reduce the accuracy of any estimation of reliability.

In Chapter 5 we define the parameters for the standard scrubbing scheme and our two enhanced schemes. We define scheme 0 as the standard scheme common in many current DRAM chips. Scheme 1 involves checking the same ECC block in each of the 15 other memory islands in the chip in parallel with the concurrent checking and correcting of the ECC block containing the data that is being accessed. Scheme 2 adds a counter to each memory island to provide systematic checking and correcting in each of the 15 other memory islands when they are idle.

Markov models are described in terms of the states that ECC blocks can be in and the transition probabilities between these states. Finally the transition probabilities are explicitly defined in transition matrices for use in our analysis.

Chapter 6 begins by describing the procedures we used to perform our analysis, the domain of the time variable, and the form of the output, then proceeds with the analysis.

7.2 Summary of Results

The Markov models that we developed for analyzing the reliability of ECC blocks proved to be effective in giving us a picture of the changing probabilities over time. By using our series model to combine the reliabilities of ECC blocks we are able to estimate the reliability of the three types of DRAM chips using the schemes which we identified.

The first results in section 6.2.2 show us that the probability of an uncorrectable error (UE) is almost directly related to the square of the soft error rate (SER) and demonstrate that our estimation of the SER only has an effect on the magnitude of the outcome. In section 6.2.3 we see that the probability of a single error (SE) climbs steadily until the first scrubbing cycle (the inverse of the scrubbing rate) is complete and then it decays very slowly for the remainder of our analysis.

When we consider the different types of ECC blocks we find that a block which is scrubbed is between 10^5 and 10^{14} times less likely to suffer an uncorrectable error than an ECC block containing a permanent fault or an ECC block without the benefit of scrubbing (see section 6.3). The variation in the values found becomes constant after approximately 10^{12} ns (see figure 6.3) and is related to the variation in the access rate.

By combining the results of our ECC block analysis using the series model we find that a chip containing just one permanent fault is 10^8 times more likely to develop an UE than the same chip with no permanent faults (see figure 6.4). In order to avoid the overwhelming effect of permanent faults we chose to compare the efficacy of our schemes using chips with no permanent faults.

Section 6.4.2 analyzes the probability of an UE developing in DRAM chips using each of our three schemes. We find that the chip using our scheme 1 is 31 times less likely to experience an UE than the chip using scheme 0 and that the chip using scheme 2 was 240,232 times less likely to suffer an UE than the chip using scheme 0. Although these results are specific for the access distribution defined in chapter 4 any other distribution would also experience similar improvements.

7.3 Future Work

There is room for improvements in our analysis model, but they are dependent on the availability of additional information. One improvement would be the incorporation of states and transition probabilities to account for developing permanent faults. Even though only 2% of the DRAM chip failures can be attributed to new permanent faults, they might have a significant impact on the effectiveness of our schemes. A second improvement would be the addition of transition probabilities which reflect the effects of β-particles as they become a significant factor.

More accurate results on the reliability of specific chips are obtainable if the manufacturers are willing to provide the following information:

What portion of the chips contain no permanent faults?

What is the SER?

At what rate do permanent faults appear in DRAM chips?

Our model determines the probability of an UE developing in a DRAM chip over time, which, for comparison purposes, we have linked directly to the reliability of a chip. A second model could be developed to estimate the probability of accessing a block, or a bit, containing an UE using the same access distribution that is adopted for use in our model.

We have only considered the effects of our schemes at the DRAM chip level. A model to identify the probability of an UE in a memory system that uses our chip model and accounts for the effects of cache and block accesses would allow system designers to more accurately estimate the reliability of systems.

SEC/DED schemes are used by manufacturers primarily because they are very effective at increasing yields and reliability. Questions that need to be resolved by the manufacturers include:

> Would the use of more extensive error correction schemes (DEC or DEC/TEC) be viable solely for the purpose of increasing reliability?
>
> Would it be more cost effective to incorporate some of the other schemes mentioned in section 2.3.1 such as complementation or extended error correction instead of enhancing the error correction schemes?

Whether manufacturers can incorporate additional schemes to manage permanent faults so that the effects of the schemes we propose could be available in all of the chips produced will be determined by market demands. The tasks of estimating the cost of implementation and analyzing the potential payback of using one of our schemes is left to others.

7.4 Concluding Remarks

Our results lead us to the conclusion that the model that we developed for our analysis is useful for estimating the reliability of DRAM chips. We found that both scheme 1 and scheme 2 are effective at significantly reducing the probability of an UE developing in a DRAM chip provided there are no permanent faults.

As the number of bits in DRAM chips continues to grow, techniques to assess and improve chip reliability will become more important. We believe that our contribution in these areas may be of benefit to chip and system designers.

Bibliography

[Aic84] Aichelmann, F.J. (1984). Fault-Tolerant Design Techniques for Semiconductor Memory Applications. *IBM Journal of Research and Development,* 28 (2), pp. 177-83.

[Amer87] Amerasekera, E.A. and Campbell, D.S. (1987). *Failure Mechanisms in Semiconductor Devices.* John Wiley and Sons, Chichester, UK.

[Ari90] Arimoto, K. *et al.* (1990). A Speed Enhanced DRAM Array Architecture with Embedded ECC. *IEEE Journal of Solid State Circuits,* 25 (1), pp. 11-16.

[Cen79] Cenker, R.P. et al. (1979). A Fault Tolerant 64K Dynamic Random Access Memory, *IEEE Trans. on Elect. Devices.,* 26 (6), pp. 853-60.

[CH84] Chen, C.L. and Hsiao, M.Y. (1984). Error-Correcting Codes for Semiconductor Memory Applications: A State of the Art Review. *IBM Journal of Research and Development,* 28 (2), pp. 124-134.

[Cha92] Char, B.W. *et al.* (1992). *First Leaves: a Tutorial Introduction to Maple V.* Springer-Verlag, NY.

[Cha91a] Char, B.W. *et al.* (1991). *Maple V Library Reference Manual.* Springer-Verlag, NY.

[Cha91b] Char, B.W. *et al.* (1991). *Maple V Language Reference Manual.* Springer-Verlag, NY.

[Che69] Chen, A. (1969). Redundancy in LSI Memory Array. *IEEE Journal of Solid State Circuits,* 4 (10), pp. 291-3.

[Che92] Chen, C.L. (1992). Symbol Error-Correcting Codes for Computer Memory Systems. *IEEE Trans. on Computers.* 41 (2), pp. 252-256.

[Coc93] Cockburn, B. (1993). Private correspondence, February 8, 1993.

[DOD74] Department of Defense (1974). Military Standardization Handbook: Reliability Prediction of Electronic Equipment. United States MIL-HNBK-217B.

[FS91a] Fifield, J.A. and Stapper, C.H. (1991). High-Speed On-Chip ECC for Synergistic Fault-Tolerant Memory Chips. *IEEE Journal of Solid-State Circuits,* 26 (10), pp. 1449-52.

[FS91b] Franklin, M. and Saluja, K. (1991). Pattern Sensitive Fault Testing of RAM's with Built-In ECC. In *Proc. Fault Tolerant Computing International Conference*, pp. 385-93.

[Hab76] Habermann, A. N. (1976). *Introduction to Operating System Design.* Science Research Associates, Inc., Chicago.

[Ham50] Hamming, R.W. (1950). Error Detecting and Correcting Codes. *Bell Systems Tech Journal*, 26 (4), pp. 147-60.

[Int86a] Intel (1986). *Components Quality/Reliability Handbook.* Intel Corporation, CA.

[Int86b] Intel (1986). *Memory Components Handbook.* Intel Corporation, CA.

[Ito90] Itoh, K. (1990). Trends in Megabit DRAM Circuit Design. *IEEE Journal of Solid State Circuits,* 25(3), pp. 778-89.

[Joh89] Johnson, B.W. (1989). *Design and Analysis of Fault-Tolerant Digital Systems.* Addison-Wesley, Reading, MA.

[Kal90] Kalter, H., et al (1990). A 50-ns 16-Mb DRAM with a 10-ns Data Rate and On Chip ECC. *IEEE Journal of Solid State Circuits,* 25(5), pp. 1118-1127.

[KKHO89] Kraus, R., Kowarik, O., Hoffmann, K. and Oberle, D. (1989). Design For Test of Mbit DRAM's. In *Proc IEEE Int. Test Conference,* pp. 316-321.

[KS86] Kinoshita, K. and Saluja, K.K. (1986). Built-in Testing of Memory Using On-Chip Compact Testing Scheme. *IEEE Trans. on Computers,* 35 (10), pp. 862-870.

[Lal85] Lala, P.K. (1985). *Fault Tolerant and Fault Testable Hardware Design.* Prentice-Hall International, London.

[Lef93] Lefebvre, M. (1993). Private communication on a-particles and their effects, June 13, 1993.

[Maz88] Mazumder, P. (1988). An On-Chip Double-Bit Error Correcting Code for Three Dimensional Dynamic RAM. In *Proc. IEEE Int. Test Conf.,* pp. 279-88.

[MW88] Meyer, J.F. and Wei, L. (1988). Influence of Workload on Error Recovery in Random Access Memories. *IEEE Transactions on Computers,* 37 (4), pp. 500-507.

[PA83] Pohm, A.V. and Agrawal, O.P. (1983). *High-Speed Memory Systems.* Reston Publishing Company, Reston, Virginia.

[Pag89] Page, L.B. (1989). *Probability for Engineering with Applications to Reliability.* Computer Science Press Inc., New York.

[Pra86] Pradhan, D.K. (1986). *Fault-Tolerant Computing, Theory and Techniques.* Prentice-Hall, Englewood Cliffs, NJ.

[PK88] Pradhan, D.K. and Kamath, N.R. (1988). RTRAM: Reconfigurable and Testable Multi-bit RAM Design. In *Proc. IEEE Int. Test Conference*, pp. 263-77.

[RK91] Rennels, D.A. and Kim, H. (1991). VLSI implementation of A Self-Checking Self-Exercising Memory System. In *Proc. Fault Tolerant Computing International Conference*, pp. 170-77.

[Sai82] Sai-Halasz, G.A. *et al* (1982). Alpha-Particle-Induced Soft-Error Rate in VLSI Circuits. *IEEE Journal of Solid State Circuits,* 17 (3), pp. 355-362.

[Sie91] Siewiorek, D.P. (1991). Architecture of Fault -Tolerant Computers: An Historical Perspective. *Proceedings of the IEEE*, 79 (12), pp. 1710- 1733.

[SL92] Stapper, C.H. and Lee, H. (1992). Synergistic Fault-Tolerance for Memory Chips. *IEEE Trans. on Computers.,* 41(9), pp. 1078-87.

[SRK87] Sayers, I.L., Russel, G. and Kinniment, D.J. (1987). Concurrent Checking Techniques - A DFT Alternative. In *Developments in Integrated Circuit Testing,* (M. Miller Ed.), Academic Press, London, 1987.

[SS82] Siewiorek, D.P. and Schwartz, R.S. (1982). *The Theory and Practice of Reliable System Design.* Digital Press, Bedford, MA.

[Sta92] Stapper, C.H. (1992). A New Statistical Approach for Fault-Tolerant VLSI Systems. *IEEE Int. Symp. on Fault Tolerant Computing,* pp. 356-65.

[TAN90] Tan, S.T. (1990). *Applied Finite Mathematics.* Kent Publishing Company, Boston.

[TK84] Taylor, H.M. and Karlin, S. (1984). *An Introduction to Stochastic Modeling.* Academic Press, London.

[TP89} Tasch, A.F. and Parker, L.H. (1989). Memory Cell and Technology Issues for 64 and 256 Mbit One-Transistor Cell MOS DRAM's. *Proceedings of the IEEE,* 77 (3), pp. 374-88.

[TT86] Tobias, P.A. and Trindade, D.C. (1986). *Applied Reliability.* Van Nostrand Reinhold Company, New York.

[UMSM92] Umemoto, Y., Masuda, N., Shigeta, J. and Mitsusada, K. (1992). Improvement of Alpha-Particle-Induced Soft-Error Immunity in a GaAs SRAM by a Buried p-Layer. *IEEE Trans. on Electron Devices,* 35(3), pp. 268-273.

[vdG91] van de Goor, A.J. (1991). *Testing Semiconductor Memories.* Wiley, London.

[Yam92] Yamaguchi, K., et al (1992). A 1.5 ns Access Time, 78μm^2 Memory-Cell size, 64-Kb ECL-CMOS SRAM. *IEEE Journal of Solid State Circuits,* 27 (2), pp. 167-173.

[YF92] Yang, G. and Fuja, T. (1992). The Reliability of Systems with Two Levels of Fault Tolerance: The Return of the "Birthday Surprise". *IEEE Transactions on Computers,* 41 (11), pp. 1490-96.

[YS92] Yang, F.L. and Saleh, R.A. (1992). Simulation and Analysis of Transient Faults in Digital Circuits. *IEEE Journal of Solid State Circuits,* 27 (3), pp. 258-64.

VITA

Surname: Pelton Given Names: Timothy Ward

Place of Birth: BC, Canada Date of Birth: Sept 18, 1962

Educational Institutions Attended:

University of Victoria	1992-1993
	1989-1991
	1980-1982

Degrees Awarded:

B.Ed.	University of Victoria	1991
B.Sc.	University of Victoria	1983

Honours and Awards:

University of Victoria Graduate Teaching Fellowship 1992-1993

Partial Copyright License

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis: The Reliability of Dynamic Random Access Memory Chips

Author: ____________________
(Signature)

Timothy Ward Pelton
(Name)

Dec. 8, 1993
(Date)