1

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

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

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

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

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

AVIS

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

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

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

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

**A Memory Architecture, Test, and Reconfiguration Approach for
Multiple Embedded DRAM**

by
Peter M.J. Diedrich, P.Eng., B.Eng.

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master Of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, CANADA
June 1989

© 1989 Peter M.J. Diedrich

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

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

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

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

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

ISBN 0-310-68851-2

Thesis contains black and white and/or
coloured graphs/tables which when
microfilmed may lose their significance.
The hardcopy of the thesis is available
upon request from Carleton University Library.

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance
of the thesis

**"A Memory Architecture, Test, and Reconfiguration Approach for
Multiple Embedded DRAM"**

submitted by Peter M.J. Diedrich, P.Eng., B.Eng.
in partial fulfillment of the requirements for the degree of
Master of Engineering

Thesis Co-Supervisor

Thesis Co-Supervisor

Chair, Department of Electronics

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, CANADA
19 June 1989

Abstract

Embedded RAM is a highly leveraged macro function that is exploited in Application Specific ICs (ASICs). In particular, dynamic RAM that is embedded in ASICs is characterised by a unique fault model. When the total chip memory may exceed 100 Kilobits distributed among several memories, the problem of test access is exacerbated by the need for a memory redundancy capability to enhance ASIC yields. An analysis and testing of a typical embedded DRAM design is performed to arrive at a fault model and the specification of test algorithms. A memory and test system architecture to support the testing of multiple embedded DRAMs and the reconfiguration for yield enhancement is proposed. The proposed reconfiguration methodology is an alternative to the use of laser or electrically blown fuse technologies. The memory architecture and test access approach are applicable to stand alone RAMs.

Acknowledgements

I would like to thank Professor C.H. Chan for supervising this thesis and my efforts at Carleton University over the past two years. As in my undergraduate thesis work, he instituted a flexible research environment which allowed me to be as productive and creative as I could be. I would like to thank Bell Northern Research Ltd., especially Allan Silburt, Robert Hum, John Hogeboom, and Ian McWalter for providing me with the facilities to conduct the practical aspects of this thesis work and the opportunity to contribute to the memory design programs. The comments and suggestions provided by Mr. Allan Silburt and Benoit Nadeau-Dostie, and the brief discussions that I had with Professor Vinod Agarwal of McGill University and Professor John Hayes of the University of Michigan at Ann Arbor, were invaluable to my understanding and application of knowledge in the areas of embedded memory testing and applications. I would especially like to thank Chuck O'Dale of BNR for his time and efforts in performing the E-Beam testing of the fault monitor devices.

Finally, I would like to thank my wife Linda for her patience and understanding while I spent so much of my time pursuing this research over the last year.

Table Of Contents

	page
Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	vi
List of Tables	ix
List of Symbols and Abbreviations	x
Chapter 1 Introduction	1
1.0 The Case For Testability	1
1.1 Embedded Memory: The Need for a Structured Approach	3
1.2 Fault Tolerance in the ASIC Environment	6
1.3 Research Focus and Organisation of Thesis	8
Chapter 2 A Survey of Testable and Fault Tolerant Memories	11
2.0 Introduction	11
2.1 The TI Megabit DRAM Family	12
2.2 Parallel Column Access Approaches	16
2.3 Self-Testing DRAM	19
2.4 Tree Hierarchy Architecture	23
2.5 Serial Access Architecture for Multiple Embedded RAM	25
2.6 Summary	30
Chapter 3 The One Transistor DRAM	31
3.0 Introduction	31
3.1 Operational Overview of the DRAM	31

	Page
3.2	Fault Modelling and Analysis 46
3.3	DRAM Fault Model Definition and Testing Strategy 53
3.3.1	Fault Model 53
3.3.2	Proposed Test Algorithms 57
3.3.2.1	March Test Sequence 58
3.3.2.2	Walking Row Test Sequence 60
3.3.2.3	Data Retention Test 63
3.3.2.4	Tests for Neighbourhood Pattern Sensitivity 64
3.4	Summary 69
Chapter 4	An Architecture for Testability and Reconfigurability 71
4.0	Introduction 71
4.0.1	Assumptions and Requirements for Memory System 73
4.0.2	Architectural Alternatives 74
4.1	Global Memory System Test Access Architecture 81
4.1.1	Overview of Test Architecture Features 81
4.1.2	Local Memory Architecture 84
4.1.2.1	Configurable Column Decoder 87
4.1.2.2	Parallel Test Access Port 92
4.1.2.3	Shift Register Row Decoder 95
4.1.3	Test Controller Architecture: Definition and Requirements 98
4.1.3.1	Memory Test and Repair Flow Algorithm 98
4.1.3.2	System Test Bus Specification 103
4.1.3.3	Parallel Walking Row Algorithm 107
4.1.3.4	Serialised Test Algorithms 110

	page
4.1.4 Redundancy Implementation Considerations	113
4.2 Related Issues	119
4.2.1 Memory Self Refresh	119
4.2.2 Parallel Array Comparison Testing	121
Chapter 5 Conclusions	123
5.0 Summary of Research and Results	123
5.1 Recommendations for Future Work	128
References	129
Appendix 1 Test and Reconfiguration Control Circuit Overview	135
Appendix 2 Configurable Column Decoder Simulation	164
Appendix 3 Parallel Test Access Port Simulation	171
Appendix 4 Shift Register Decoder Simulation	184

List of Figures

Figure No.		page
1.1.1	Embedded RAM Configuration	2
1.1.2	Embedded RAM Configuration for BIST	4
2.1.1	Megabit Commercial DRAM with DFT Features	13,14
2.2.1	Parallel Access Architectures for High Density RAM	17
2.3.1	Self-Testing DRAM Architecture	20
2.3.2	Circular Shift Register Scheme for Self-Testing DRAM	21,22
2.4.1	Tree Architecture for RAM	24
2.5.1	Serial Access Technique for Embedded Memory	26
3.1.1	DRAM Fault Monitor Pinout Diagram	32
3.1.2	DRAM Fault Monitor Architecture and Block Diagram	33
3.1.3	Timing Diagrams for Internal Control and Datapath	34
3.1.4	Cells, Precharge, and Bitline Sense Amplifiers	35
3.1.5	Dummy Bitlines and Sense Timing Circuit	36
3.1.6	Reference Voltage for Bitlines	37
3.1.7	Row Decoder and Wordline Drivers	38
3.1.8	Static Column Decoders	39
3.1.9	Data I/O Buffers	40
3.2.1	Cell Layout	47
3.2.2	Classical 9-Cell Neighbourhood and 5-Cell Subset	50
3.2.3	DRAM Fault Monitor Cell Decoding and Neighbourhoods	52
3.3.2.1.1	March Test Sequence for DRAM Testing	59
3.3.2.4.1	PSF4 Test for DRAM Static Pattern Sensitive Faults	65

Figure No.		page
3.3.2.4.2	PSF4 Test for DRAM Static Pattern Sensitive Faults Simplified	66
3.3.2.4.3	Modified MOD3 Test for Pattern Sensitive DRAM Faults	68
4.0.1.1	Basic Embedded DRAM Architecture	72
4.0.2.1	Content Addressable Memory Cell Concept	77
4.0.2.2	Bit-Block Oriented DRAM Decoding Architecture	79
4.1.1.1	Memory Test System Access Architecture	82
4.1.2.1	Embedded Memory Test Architecture	85
4.1.2.1.1	Configurable Column Decoder Circuit Design	88
4.1.2.1.2	Configurable Column Decoder Concept	89
4.1.2.1.3	Shared Configurable Decoder for a Dual Cell Array Embedded Memory	90
4.1.2.1.4	EN/DIS Scheme for Inter- and Intracolumn Repair	91
4.1.2.2.1	Parallel Test Access Port Design	93
4.1.2.3.1	Shift Register Decoder Design	96
4.1.3.1.1	Memory Test and Repair Flow	99
4.1.3.3.1	Parallel Walking Row Algorithm Flow Diagram	108
4.1.4.1	Column Redundancy Architectural Overview	114
4.1.4.2	Reconfiguration Concept in the Absence of Mission Mode Y-decoders	117
4.1.4.3	Datapath Modification in Absence of Mission Mode Y-decoders	118
4.2.1.1	Self Refresh Timing Example	120
A.1.1.1	Reconfiguration Control Block Diagram	138
A.1.1.2	EN/DIS Control Circuit Design	139
A.1.1.3	Redundancy Usage Control Circuitry	140

Figure No.		page
A.1.1.4	Test and Reconfiguration Status Logic	141
A.1.1.5	Sample Timing Diagram for Reconfiguration Scheme	142
A.1.2.1	Parallel Access Compare and Write Sample Timing	145
A.1.2.2	Local Parallel Test Signal Generators	146
A.1.2.3	Parallel Test Access Signal Generation Logic	147
A.1.2.4	Parallel Test System Test Bus ERROR Control	148
A.1.3.1	Shift Register Decoder Operational Overview	150
A.1.3.2	Overflow Detector Logic	151
A.1.3.3	Overflow Flag Propagation and Shift Register Shut Down	152,153
A.1.3.4	Shift Register Control Signal Logic Relationships	154
A.1.3.5	Control Signal Multiplexing for Test and Mission Modes	155
A.1.3.6	Sample Timing Diagram for Shift Register Control Signals	156
A.1.4.1	Serial Address and Data Capture Scheme	159
A.1.4.2	Address Buffer Configuration	160
A.1.4.3	Data I/O Buffer Configuration	161
A.1.4.4	Parallel Data Comparator Circuit Concept	162
A.2.1	Configurable Y-decoder Simulation Schematic	165
A.2.2	Configurable Y-decoder Simulation Plot	170
A.3.1	Parallel Test Port Simulation Schematic	172
A.3.2	Parallel Test Port Simulation Plot: Parallel Write	181
A.3.3	Parallel Test Port Simulation Plot: Error Detection	182
A.3.4	Parallel Test Port Simulation Plot: No Error Detection	183
A.4.1	Shift Register Decoder Simulation Schematic	185
A.4.2	Shift Register Decoder Simulation Plot	190

List of Tables

Table No.		page
2.6.1	Summary of Testable RAM Architectures	29
4.1.3.2.1	Test Status Signal Mode Description	105

List of Symbols and Abbreviations

*	denotes an active low signal
[RAS]	denotes an external or off chip input/output signal
A_i	Address input i
AS_i	Serial Address input i
ASIC	Application Specific Integrated Circuit
ATE	Automated Test Equipment
BIST	Built In Self Test
BNR	Ball Northern Research
BWRT	Bulk Write signal
CAM	Content Addressable Memory
CAS	Column Address Strobe
CMOS	Complementary Metal Oxide Semiconductor
CS	Column Select
DFT	Design For Test
DI	Data Input
DO	Data Output
DRAM	Dynamic Random Access Memory
DS_i	Serial Data input i
DSP	Digital Signal Processing
ED	Expected Data
EDB*	Expected Data Bar
EN/DIS*	Enable/disable input to reconfigurable column decoders
ERROR*	System Test Bus error bus
ERROR_{loc}	Local error detection signal

FBL	Float BitLine Left
FBLLOK	Float BitLine Left OK
FBLR	Float BitLine Right
FBLROK	Float BitLine Right OK
FIFO	First In First Out buffer memory
GALPAT	GALloping PATtern for memory test
GALROW	GALloping ROW pattern for memory test
I/O	Input/output signal
IC	Integrated Circuit
IFA	Inductive Fault Analysis
JTAG	Joint Test Action Group
JEDEC	Joint Electronic Devices Engineering Committee
K	Kilobit (2^{10} bits)
lcc	Local signal
$\log_2$	logarithm to base 2
LBL	Low BitLine
LD/CMP*	Load/Compare signal
LS	Left Side select
LSB	Least Significant Bit
M	Memory system size parameter
mV	millivolt
Mb	Megabit (2^{20} bits)
MATS	Modified Algorithmic Test Sequence
MSB	Most Significant Bit
MTC	Memory Test Controller
MTCH	Match output of parallel column data comparator circuit

MTRF	Memory Test and Repair Flow
n	Generic memory size or density symbol
N	Memory system size parameter
NMOS	N-channel Metal Oxide Semiconductor
O()	Order of magnitude symbol for memory test length
OVF	Overflow
OVFR	Overflow Reset
p	Parallelism factor
PC	Personal Computer
PLA	Programmable Logic Array
FMWROW	Parallel Modified Walking Row
PRE, PREB*	Precharge signals
PSF	Pattern Sensitive Fault
PTST	Parallel Test
PWRT	Parallel Write
R	Memory system size parameter
RAM	Random Access Memory
RAS	Row Address Strobe
RASC	Row Address Strobe Complemented
RASB	Row Address Strobe Buffered
RESET	System Test Bus reset signal
RFSH*	Self ReFreSH timing signal
ROM	Read Only Memory
R, RB*	Reconfiguration control signals
RG*	Reconfiguration control signal
REDUSE*	Redundancy Used control signal

RS	Right Side select
R/W*	Read/Write Bar
SADI/PD	Serial Address and Data Input/Parallel Data
SAE	Sense Amplifier Enable
SAN	Source enable sense Amplifier N-channel
SAP	Source enable sense Amplifier P-channel
SH	Shift clock signal
SHA	Shift clock for Address registers
SHAL*	Shift clock Bar for Address registers
SHB*	Shift clock Bar signal
SHC	Shift Clock signal from System Test Bus
SHD	Shift clock for Data registers
SHDB*	Shift clock Bar for Data registers
SHISO*	Shift register Isolate signal
SHRAS	Shift register RAS interlock signal
SL	Sense Line
SRAM	Static Random Access Memory
STATUS*	System Test Bus test status signal
STB	System Test Bus
TST1	Test Status or mode control signal
TST2	Test Status or mode control signal
TY	Test Y-decode signal
VCC	Positive Voltage supply rail
VLSI	Very Large Scale Integration
VREF	Voltage REFerence
VSS	Ground Voltage supply rail

WE	Write Enable
WLB*	Wordline (active low)
WLBI	WordLine Bar input signal
XEN	X-decoder Enable
XEN0	X-decoder Enable reset
Yi	Column select or Y-access signal

1. Introduction

" A circuit is testable if a set of test patterns can be generated, evaluated, and applied in such a way as to satisfy predefined levels of performance defined in terms of fault detection, fault location, and test application criteria within a predefined cost budget and time scale." [Benn84]

1.0 The Case for Testability

This concept of testable digital circuits points to two critical issues in testing ICs: **controllability and observability** of individual circuit inputs and circuit responses, respectively. Coupled closely with these issues are the tasks of test generation, fault coverage evaluation, and test access or delivery. There are a variety of approaches to enhancing the testability of integrated circuits. For example, serial test techniques are used and they employ, for example, Level Sensitive Scan Design (LSSD). Unstructured or ad-hoc testability techniques which include such measures as increasing the "probe-ability" of circuits, or the isolation of circuits with specialised functions, are also used. A highly structured and flexible test access bus approach, an example of which is the P1149 IEEE JTAG proposed standard for handling board level testing requirements, is yet another approach [Turi88].

Improving testability, as described above, and more specifically test access offer many advantages to the ASIC and commodity IC manufacturer. Reduced test time, enhanced fault coverage, high productivity, and reduced development time may all be realised [Park88]. But fundamental to enhancing testability is the establishment and maintenance of accurate fault models for the circuits, macros, and ICs that are made testable. A fault is a behavioral manifestation of a physical, or manufacturing defect that results in improper

Embedded RAM Configuration

System refers to chip level circuits or macros which utilize the embedded RAM.

Figure 1.1.1

operation of a circuit. The design of test algorithms is based upon the fault model, or the set of all possible faults, to which a circuit may be subject. In recent practice, the Stuck-at model [McAd86] for CMOS integrated circuits has been used as the standard for logic circuits. However, with the increasing use of specialised macros which are embedded within logic, such as memories, PLAs, multipliers, and the various configurations that these functional blocks may assume, the requirement for more accurate fault models than the classical Stuck-at model is readily acknowledged [Park88, Been86]. Moreover, the existence of macro-specific fault models and the proliferation of several different macros within the same application specific integrated circuit (ASIC) obviates the need for special attention to the testability - controllability and observability - requirements of the individual macros within the IC.

1.1 Embedded Memory: The Need for a Structured Approach

Memories occupy more circuit sockets at the PC board level than any other commodity integrated circuit. Correspondingly, in ASIC environments, the various species of memories are exploited for their speed and density on-board the actual IC. They may act as buffers, FIFOs, high speed cache memories for processor, or may be uniquely configured for DSP applications. With such specialised internal macro blocks such as memories, and their more complex fault models, there is a need to partition the IC for testing. The complex fault models associated with memories require the application of extensive algorithmic test stimuli, including pseudo-random test patterns, to provide adequate fault coverage. However, there is also a need to simplify the method by which ICs are tested and the specialised embedded macro block, such as a memory depicted in Figure 1.1.1, stands in the path of test and test access simplification. Memories are not easily adaptable to scan test techniques. The use of Built In Self Test (BIST) is therefore the only

Figure 1.1.2
Embedded RAM Configuration
for BIST

alternative to external testing and full test access that can be exploited practically to reduce the complication introduced by embedded macro blocks.

The BIST circuitry required to test a circuit block or macro is located *in situ* as shown in Figure 1.1.2. It may be tailored for any specific fault model and thereby provide high fault coverage, it is cost effective for testing depending upon specific applications, it reduces the requirements of test pattern generation and special access to embedded blocks, and it reduces the excessive serialisation of input test stimuli and output response. Test generation times for ICs are also reduced when BIST is used, and because there is direct access to the embedded block, the circuitry may be tested at mission mode speed depending upon the architecture of the BIST circuit. The need for elaborate test equipment to satisfy access protocols and complex fault models typical of embedded blocks such as RAM is eliminated by using BIST. Self-test schemes which are embedded along with the macros may be easily extended to a board and system level concept to act, for example, as an aid to testing and diagnostic investigation of a circuit, IC, board, or system. However, at the IC level, BIST requires silicon area, may result in performance degradation, and also may increase chip design time and development costs [Park88]. To offset these possible drawbacks of BIST, a well orchestrated approach to testing embedded blocks is required at the IC level.

To focus more precisely on embedded RAM, let us consider the specific problems that are associated with this macro function. Embedded memories may be internally accessed in a variety of ways. For example, the output may be serial, the addressing sequence may be fixed as in the case of a FIFO, or there may be special bits of a memory that are used solely for parity checking and are inaccessible even to the controlling and test circuitry. Embedded memories used as shown in Figure 1.1.1, especially in telecommunications and signal processing applications, can have a wide variety of possible word widths, ranging from one to greater than 40 bits in some cases. This is in direct contrast with commercial

RAMs which have standard x1, x4, and x8 data I/O pinouts for the most commonly available devices. In addition, on a given ASIC, multiple embedded RAMs may be used for any number of special applications.

Exploiting embedded RAMs is a high performance and high density design alternative. Any degradation of the requisite RAM functionality, performance, variety, and flexibility severely limits the on-chip utility and applicability of RAM in ASICs. The testing of multiple embedded blocks of RAM of varying sizes and configurations, including dual port RAM, on the same IC is extremely complex. ATE architectures and capability, targeted for general purpose digital integrated circuits, cannot support such a testing task without excessive data serialisation, special access requirements, and increased IC test time and design effort. Moreover, even dedicated memory testers that are available do not incorporate any special features to test embedded RAM, especially as far as access protocols are concerned, nor do they allow for a user-defined and flexible algorithmic capability to enhance fault coverage for specific fault models.

1.2 Fault Tolerance in the ASIC Environment

These testing logistics problems for embedded memories are further exacerbated by the fact that most manufacturing lines for ASICs are generally not optimised for the densely packed and processing sensitive circuits that characterise dynamic or static RAMs. Coupled with the fact that the use of multiple embedded RAMs can quickly run the number of bits on a given ASIC design up above 64K of memory, the potential exists in these generic manufacturing processes to designate memory as the yield limiting component of an ASIC. In the ASIC product development arena [Butz88], a low volume, quick turnaround mode of operation, product yield and quality are critical to success. Testability is not the only consideration for the embedded RAM problem: maximising yield through

fault tolerance is equally important.

RAM, because of its dense circuitry and finely tuned circuit design, is extremely susceptible to manufacturing defects that produce cell array faults. The probability of a processing defect resulting in a circuit malfunction or fault in the cell arrays is significantly greater when compared to the ability to tolerate similar faults in an equivalent area of, for example, address buffer or control logic circuitry in a given RAM layout design. Almost all approaches to memory fault tolerance involve the use of redundancy of cell array components. Memory, embedded or otherwise, does not typically tolerate circuit faults in the control and peripheral logic. Cell array redundancy does not contribute to yield enhancement in this case.

Memories are the only commercial devices to successfully use redundancy as a fault tolerance technique and have historically used additional rows, columns, or both rows and columns to enhance circuit yield [Moor86]. Most redundancy techniques employ fusible links in circuits. These links may be blasted using laser or electrical stimuli to invoke the redundancy circuit. The inclusion of these fusible links may result in the degradation of circuit performance in critical areas, such as wordline decoders, however, if prudent design practice is maintained, the real penalty is only incurred in silicon area [Died87], not performance. A problem with redundant elements from a testability point of view is that typically they are not testable. An effective fault tolerant scheme must make provision for the testing of the additional and unused circuit elements.

Much of the practical effort in fault tolerance and RAMs has focused upon very high density (64K and above) DRAM and SRAM. New efforts in VLSI fault tolerance research have switched from VLSI to Wafer Scale Integration (WSI). Concepts such as block realignment at the wafer level have been introduced and have been shown to be an effective technique for increasing circuit yield [Stap88]. The fundamental concept that is applied in all of these research efforts is that memories are regular and repeated structures and as such

lend themselves to redundant element substitution techniques. At the WSI level or the 1-4 Megabit DRAM level, blocks may be entire RAMs or several hundred kilobits of storage capacity and control and decoding circuitry.

The problem of realising fault tolerant implementations for embedded RAM, however, is more difficult and has not been addressed in these previous research efforts. As an example of the limitations inherent to the ASIC environment, the use of laser blown fuses can be ruled out almost immediately. In the low volume, fast turnaround ASIC manufacturing process, it is not reasonable to use sophisticated and costly laser equipment, or to incur the time for testing the ASIC design, aligning laser beams to program the fuses, and then repassivating a device after the fuses have been programmed. "Soft" fault tolerance is a practical engineering solution to the problem. A standard test interface for detecting and removing faulty elements in a multiple embedded memory system is required to reduce, or at least limit any potential increase in the testing time or complexity for ASICs.

1.3 Research Focus and Organisation of Thesis

The preceding discussion of the complexity and diversity of issues that are presented by the use of multiple embedded RAM in ASIC design and manufacturing environments illustrates the need for a structured approach to embedded memory design for testability and to fault tolerance. Clearly, it is necessary that the design of the memory and the memory system in an ASIC be closely coupled to the design of test algorithms, test circuitry, and redundancy circuitry. This thesis will study the following issues:

1. the establishment of a DRAM fault model based upon the analysis of a representative embedded RAM design;

2. the specification of representative memory/ test algorithms to provide coverage of the fault model;
3. the design of a system level memory architecture, and the required circuit implementations within the memory macros, to enhance testability and to allow the use of redundant elements for multiple embedded DRAM in the ASIC design environment.

Chapter 2 of this thesis will survey and discuss the various approaches which have been taken in resolving the fault modelling and testing issues specifically as these relate to RAM. Architectures, application environments, tradeoffs, testability improvements, and other relevant performance criteria will be compared for the techniques in the survey and a summary of features potentially applicable to the multiple embedded DRAM environment will be presented.

A realistic approach to solving the combined problem of fault modelling, testing, and reconfiguration in embedded DRAM is necessary. The design analysis, fault modelling analysis, test algorithm proposal, and test results for an embedded DRAM fault monitor device are presented in Chapter 3.

The proposal for a new embedded memory circuit and system architecture for test and reconfigurability is outlined and analysed in Chapter 4. Architectural alternatives are discussed to implement the novel DRAM features in the individual embedded DRAM blocks, including the required circuit design specifications supported by circuit simulations. A global system architecture and embedded memory test bus interface specification is presented and the necessary additions to the local memory control circuitry to implement the memory test and repair algorithm are summarised in detailed circuit descriptions. The chapter concludes with an analysis of redundancy implementation issues and a brief

discussion of the adaptability of the novel embedded memory and system level memory test architecture to the incorporation of self-refresh circuitry and other testability enhancements at the circuit and algorithm level.

2. A Survey of Testable and Fault Tolerant Memories

2.0 Introduction

The problem of enhancing the testability of dynamic RAM is not a new one. As the technology for mass production has increased achievable memory densities from 1 Kilobit to beyond 4 Megabits, test time has become the primary consideration and the need for efficient testing schemes has also increased. Testing times for Megabit memories that are in excess of 10 seconds are typical given current test algorithms and access techniques. The only relief available to the memory producer and user in achieving higher testability and lower testing time is architectural modification to the memory. New approaches to DRAM architectures have been taken to reduce testing time and cost. In the succeeding discussion, five approaches to solving the high density DRAM testing and testability problem will be discussed and evaluated. All incorporate useful design and architectural features that can be utilised and exploited in developing an architecture for embedded DRAM.

The six approaches will be evaluated based upon the following criteria:

1. major architectural innovation or concept;
2. application environment (VLSI or embedded RAM, functional test or characterisation tests);
3. testability improvement or enhancement achieved and flexibility of test algorithm application;
4. ability to test device at speed and emulate normal memory mission mode operation during test application;

5. performance degradation or enhancement incurred due to added circuitry or architectural change;
6. area penalty incurred by additional circuitry;
7. potential fault tolerance of the architecture and the accompanying ability to test any redundant circuits incorporated in the design.

Some of these evaluation criteria are subjective in nature and may only be inferred from the literature available on each memory design or architecture. In addition, in the case of area penalty, the available data on such area penalties incurred in each approach is extremely specific to both layout design rule and to the density of the memory under consideration. The comparison of these various design approaches on the basis of the above six criteria, however, will give a strong indication of the advantages and drawbacks of each architecture and which features will be exploitable in the embedded DRAM environment..

2.1 The TI Megabit DRAM Family

The Texas Instruments Megabit DRAM family [McAd86] is a commercial DRAM product set which utilises high density trench capacitor storage cells. The architecture, shown in the accompanying Figure 2.1.1, follows the classical four quadrant configuration that has characterised DRAMs since their mass production began in the late 1970's. Hierarchical decoding of the storage array addresses results in eight bits of memory, two per quadrant, accessed during each active memory cycle. The standard pinouts for the x1 and x4 versions of the Megabit devices have an additional pin that is unused. This design

Megabit Commercial DRAM with DFT Features

(a) 1 Mb Chip Layout Architecture

Figure 2.1.1

(b) Address and Control Signal
Multiplexing for Test

Figure 2.1.1

uses the extra pin to provide added test functionality and the parallelism of the array decode to reduce total test time.

The device is put into test mode using overvoltage detection on the [RAS*] pin to activate added internal circuitry which reconfigures the memory as eight 128Kx1 equivalent but parallel memories. This also reconfigures certain of the address and data I/O pins to program or select a specific test to be executed, and to provide input data stimuli and allow access to the output data response from the test, respectively. The important feature that this architecture affords is the ability to test the various subarrays of memory in parallel. This reduces test time and allows internal or external comparison of the outputs of the various subarrays under test for any applied test pattern. The test time afforded by the eight bit parallel read and write operation is reportedly five times less than that which would be necessary for the testing the megabit DRAM on a per bit basis.

In addition to the ability to perform parallel read, write, and compare operations, this approach uses several circuits to perform the following tests which may be programmed from selected address bits: static refresh disturb, field plate static and dynamic leakage, sense amplifier margin, and redundancy roll call tests. External control of the sense amplifier timing and cell storage array back bias voltage generator oscillator signals is also possible via a programmed test mode. The extra pin is used as the external input which exercises the various signals used in these tests; for example, the voltage applied to the storage cell field plate is obtained from the voltage applied through the test pin. In all test modes, the device is accessible at mission mode speed. Except for the parallel read and write capability and the redundancy roll call - essentially a test to determine if the activated redundant decoders and address mapping circuitry has been successfully invoked - the remaining tests listed above are design characterisation tests to aid in design development, and are not intended to be functional tests. The redundant rows and columns of cells that are employed in the design are only testable after the normal defective cells in a cell array

have been replaced by the redundant cells. Full retest is required after programming the redundancy circuits.

This family of high density DRAMs utilises circuit innovation to reconfigure the devices for testing, either for the functional or the design characterisation tests. The area penalty incurred by the design for test functions that are included on board is reportedly less than 5% of the total chip area. Little or no deviation from conventional DRAM architectural concepts are found in the memory design and no penalty in performance is incurred due to the added test circuitry or operational modes.

2.2 Parallel Column Access Approaches

There are two approaches that have been documented and which make use of parallel access to all of the columns (bitlines) in a particular DRAM architecture [Mazu87, Inou87]. This architectural approach is aimed at reducing test times in VLSI megabit memories.

Figure 2.2.1 illustrates the approach that is taken. In testing all of the columns, or all of the cells along a given wordline, simultaneously, improvement in test times on the order of 2 to 3 orders of magnitude over non-parallel approaches as DRAM density is increased from 64K to 16 Megabit levels is reported. In both of these architectures, many existing memory test algorithms such as MATS, GALROW, or GALPAT [Abad83], may be easily adapted and applied to the memory in a parallel manner. The verification of the proper functionality of a row of cells through the parallel test approach then allows a particular row to act as a reference for tests to verify the functionality of the peripheral circuits, used in normal mission mode for memory access but which are bypassed during the parallel access test mode (column decoders, I/O datapaths and buffers), and also the functionality of the testing and output response compare circuits. The peripheral and test circuits are tested and valid memory bits are used as reference locations for the testing of these circuits. Normal

Figure 2.2.1

Parallel Access Architectures for High Density RAM

mission mode speed and operating stimuli are applied.

Minimal modifications to the overall circuit design and architecture of the memory is achieved in both of these designs [Mazu87, Inou87] and the column decoder and comparator circuits are the only affected components of the design. The technique proposed in [Inou87] reports a circuit overhead penalty of less than 1% for a 2 Megabit DRAM employing the parallel access circuitry.

The alternative design architecture [Mazu87] also invokes the concept of array segmentation of the memory and gives a circuit overhead figure in terms of the number of bits, N and number of parallel cell arrays, p , and assumes that the cell arrays are organised as $\sqrt{N}/\sqrt{p} \times \sqrt{N}/\sqrt{p}$, or are square cell arrays. The figure quoted for the transistor overhead is $2\sqrt{N}\sqrt{p} + p\log_2 N - p\log_2 p + 12p$, however, the area overhead incurred should be comparable to the 2 Megabit design [Inou87]. No performance degradation in mission mode is reported in either case and any possible loss of performance cannot be inferred from the circuit proposed for the parallel compare and access functions. The parallel compare circuits are not part of the normal mission mode data I/O paths.

This architectural approach, exploiting parallel access to all columns, is very flexible in terms of test stimuli application since full external access to the DRAM is available. Parallel test of segmented arrays each using the parallel access and read-compare circuitry would greatly enhance the two architectures. The parallel access to columns also lends itself to the addition of full self-test circuitry to perform the memory cell, peripheral circuit, and test circuit functional tests. With circuit modification, redundant elements could be tested using this parallel access technique as well. No modification to redundant decoders would be required if only redundant columns existed since they are normally activated by the wordline which is common to normal and redundant cells in each column.

2.3 Self-Testing DRAM

An architecture for a fully self testing dynamic RAM design has been proposed [Haye85] as a means of fully covering the fault model for a high density DRAM and of reducing the test time required to implement the test algorithm to obtain the required fault coverage for very high density DRAMs. The architecture contains four major innovations: the testing of parallel partitions simultaneously, the generation of data stimulus patterns using the cell storage array, the use of on-chip refresh logic as part of the BIST control, and the use of parallel cell storage array response comparison. Figures 2.3.1 and 2.3.2 summarise these innovations. The BIST algorithm employed by the design of the test generation circuitry is very specific to the particular DRAM fault model that is defined for the particular design and no flexibility of algorithm application is possible for the chip without circuit modification. The design was conceived for a full BIST implementation.

The sense amplifier and column decoder are modified to allow the memory to act as a circular shift register and a data stimulus propagator/generator, and to allow for parallel comparison of the responses from duplicated identical cell arrays. The modification of the sense amplifiers and the addition of a flip flop to form a circular shift register results in a loss of performance in the cell access and sense operation due to the addition of a series device between the cell and its sense amplifier. It is reported that the cycle time during test must be increased by 10% due to increased write times during shift operations. Otherwise the memory may be tested at speed. The comparison of the parallel cell array partition responses has the potential drawback that, in a case in which, as an example, two compared cells are defective, one from each partition, if the responses are identical, no error will be detected even though both cells are faulty.

The area overhead in a designed but reportedly untested 4K version of the self-testing DRAM is approximately 12% of the total die area, and at the one Megabit density level, the

Self Testing DRAM Architecture

Figure 2.3.1

Circular Shift Register Scheme for Self Testing DRAM

Figure 2.3.2 (a) Cell Arrays

(b) Modified Sense Amplifier

Figure 2.3.2

area overhead is projected to be 5%. All of the additional test circuit logic and the circuits not used during self-test are independently testable. Test time for the particular test algorithm, a sliding diagonal, is reduced from $O(n^2)$ to $O((n/p)^{0.5})$, where n is the total number of memory cells and p is the number of parallel, identical cell array partitions. The self-testing DRAM is not reported to incorporate any redundant cell elements.

2.4 Tree Hierarchy Architecture

The decoding hierarchy proposed for a DRAM has classically been at most a two level decode, using 1-of- n array selection and row and column binary decoders. The resulting cell array organisations were typically of the four quadrant variety that characterises the previously discussed TI Megabit DRAM family and most other commercial DRAM designs. The tree decoding concept takes hierarchical array selection significantly further as shown in Figure 2.4.1 [Jarw88]. The memory is segmented into N subarrays, or so called leaf nodes, and $\log_2 N$ switches or branch nodes are used to route data and control signals to the selected leaf node. The most significant $\log_2 N$ addresses are used to steer the selection of a particular node down through the hierarchical tree and the remaining addresses are shared and used to activate the row and column decoders in each replicated tree node, a cell subarray, and are common to all N subarrays.

To test this particular memory the architecture allows all of the leaf nodes to be exercised simultaneously. A distributed response comparator detects the presence of an error. The distributed comparator does not allow location of an error to be done in test mode. Once an error flag is raised, the device may be put into mission mode and each leaf tested individually to locate a faulty cell. As in other architectures discussed, the testing of circuitry that is bypassed in test mode, such as the branching or switching nodes, may be performed when a reference set of functioning cell array elements is known.

Figure 2.4.1
Tree Architecture
for RAM

This tree steering structure for address, data, and control signals results in the following block architecture for memory:

Clearly, the targeted application for this architecture is very high density memories and the design allows for reduced test time and performance enhancement. Limited fault tolerance is recommended using block realignment techniques and laser blown fuses in the decoder tree to generate partially good products. It is indicated that 8 leaf nodes are optimal for a projected 10% performance enhancement. A 14% area penalty is incurred for a one Megabit design based on the tree structure. The results presented also indicate that the area penalty becomes less severe and that the performance enhancement increases with increasing memory density.

Full access to all leaf cells in parallel allows any algorithm to be applied to each and for the output responses to be compared in parallel either internally or externally. No performance degradation due to the added circuitry is indicated and the ability to test the circuit at operating speed is not impeded by the tree architecture. The architecture also will support row and column redundancy at the leaf node level but no analysis of the benefits of this level of redundancy has been carried out. An address mapping circuit that is used to provide maximum memory utilisation in the event of faulty cells or leaf nodes demonstrates that address remapping as opposed to laser programming will result in performance penalties in the memory as well as incur significant area overhead penalties.

2.5 Serial Access Architecture for Multiple Embedded RAM

A BIST architecture for embedded static RAM has been recently proposed [Agar88]. The purpose of this test architecture is to implement a structured approach to testing multiple embedded RAMs of varying sizes and configurations without incurring performance degradation or utilising cumbersome or complex design enhancements, while facilitating a test access method which does not require separate access to each embedded memory to obtain the necessary test fault coverage and which interfaces easily to the

Figure 2.5.1

external environment or to a BIST circuit. Another important consideration in developing the BIST circuit itself is that its circuitry - counters, for example - be easily shared with mission mode circuitry which surrounds the embedded memory blocks. In any BIST circuit implementation, the dedicated BIST controller determines the algorithm to be applied and its finite state machine implementation is constructed based upon a chosen test algorithm.

The fundamental insight that has been exploited in this particular architecture is the use of the memory cells and the data I/O latches which exist in normal embedded blocks as the master and slave elements of a shift register. Figure 2.5.1 demonstrates the basic concept that is implemented. By placing multiplexers on the output of the data I/O latches to allow the bits of the I/O word for each embedded memory to be connected serially, the data write strobe to the memory becomes the master clock phase, and the data read strobe becomes the slave clock phase. Data propagates from a single serial input pin into the data I/O latches through the multiplexers and is written to accessed memory words. When it is read from the memory word, an inherent shift function occurs and the data propagates to the next memory cell or word under test. The serial output data stream is monitored by the BIST circuit to determine whether errors have occurred in the serial writing and reading of the data. Errors in a memory may cancel each other because output response comparison takes place after the serial data bits are shifted from the last I/O bit output buffer and multiplexer circuit.

All of the decoders, control, and I/O circuitry of the memories are tested and the only addition to the memories is a multiplexer for each bit in the word of each memory. The memories may all be tested at speed and few additional control or data lines are required except for the multiplexer interconnections.

There are two proposed methods for adapting the serial access scheme to include several memories. The first approach is to connect the serial output of one memory to the

serial input of the next memory. This creates a serial shift register which consists of all of the data I/O circuits connected through the multiplexers. It requires only the extra line from one memory to another to complete. There is an added restriction that each memory must have the same number of rows (X-address space). No restriction as to I/O word width is required although the serial bit position counters must be modified to have the same number of states as the sum of the I/O word widths of all of the connected memories.

The second possibility for unifying the testing of the multiple embedded blocks of memory is to allow each to have its own serial output bus so that each memory may be tested separately. The serial I/O buses from all memories are multiplexed using additional decoder circuitry and the BIST controller circuit receives only the data on the serial I/O line from the memory block under test. This technique requires both a memory enable selection decoder and a serial I/O line decoder placed in front of the response comparison circuit. The memories are also not tested in parallel as they are using the first interconnection approach. The BIST controller must also be adjusted to account for the differing serial data streams that are expected from each of the individual memory configurations. The advantage to using this approach, however, is that there is absolutely no restriction on the size or configuration of the various memory blocks and no additional circuitry is required. Testing at full mission mode clock speed is maintained.

Reported figures for area overhead vary according to the number of memories and the density of memories utilising this unified test approach. For a one Kilobit static RAM, the BIST circuit to RAM area ratio is approximately 14%, and this is further reduced to 10% at the 2 Kilobit density level. For a two memory system with shared BIST circuitry, the ratio is reported to be 5%. The area overhead ratios improve considerably when the BIST circuit area is compared with the area of the chip in which the embedded memory block(s) reside.

Summary Of Testable Architectures

	Commercial DRAM	Parallel Column Access	Self - Testing DRAM	Tree Hierarchy	Serial Access Architecture
Major Architectural Feature or Innovation	Multiple cell array parallel read/write. Multiple tests	Parallel read/write access to bitlines and/ or sense amplifiers.	Full BIST with Modified Sense Amps and parallelism.	Parallelism and tree structure for data/address.	Serial Data R/W and Master/Slave Operation of cell and I/O latch
Application Intended	1 Mbit + DRAMs	1 Mbit + DRAMs	Medium to High density DRAMs	256K + DRAMs	Embedded RAM:
Testability Enhancement Achieved	Factor of 5 reduction in test times.	$O(n)$ test algorithms reduced to $O((n/p) \cdot 0.5)$	$O(n)$ test algorithms reduced to $O((n/p) \cdot 0.5)$	> 50% decrease in test time for $O(N \log N)$ tests.	Test multiple embedded memories with serial connections of I/Os
Ability to Test at Speed	Yes	Yes, but not during parallel access mode.	Minimal during test cycle	Yes	Yes
Performance Degradation	None	None	Decreased sense amp performance.	None	None
Area Penalty	5% for 1Mbit.	1% for 2Mbit	12% for 4K DRAM 5% for 1Mbit DRAM	Variable, but 14% for 1Mbit	Variable, but < 15% for multiple memories
Fault Tolerance	Ability to test red. decoders and cells	-	-	Block realignment is possible.	-

Table 2.6.1

2.6 Summary

The following table summarises the various DRAM architectures based upon the evaluation criteria. Though all of these architectures are mainly applicable to very high density RAMs, especially DRAM, there are extremely important concepts and innovations embodied in their respective designs and which are applicable to embedded DRAMs:

1. parallel access to and compare of all columns accessed for each wordline;
2. parallel internal comparison of test response between identical memory partitions;
3. serial data capture and test circuit sharing;
4. flexibility of test algorithm specification; and,
5. minimal degradation of mission mode performance.

In succeeding analyses and discussions, the utility of architectural modification to DRAMs targeted for a multiple embedded memory ASIC environment will be demonstrated through the design analysis and fault modelling of a typical embedded DRAM, and through the proposal of a memory and memory test system architecture for ASICs.

3. The One Transistor DRAM

3.0 Introduction

The vehicle for fault modelling and test strategy development is a one transistor DRAM design intended for fault monitoring [Died89]. The chip consists of four quadrants, one of which is devoted to a 2Kx4 DRAM. A chip layout overlay, functional block diagram, and selected circuit schematic descriptions, as well as sample timing diagrams for the DRAM are shown in the accompanying figures, 3.1.1 to 3.1.9.

3.1 Operational Overview of the DRAM

To illustrate the basic principle of the operation of this DRAM, it is useful to proceed through the sequence of events that occur for a row access read cycle as summarised in the sample timing diagram, Figure 3.1.3. Initially, $[RAS^*]$ (* denotes an active low signal and on figures an overscore is used) is held high. To initiate the row access sequence, $[RAS^*]$ is asserted low. The RASC (RAS Control) clock follows an inverter delay later and drives RASB (RAS Buffered) high. RASC generates a LS or RS, left array or right array, selection signal. RASB, duplicated for two sets of row decoders, enables the generation of an active wordline by releasing the dynamic precharge devices in each of the row decoders. First, however, RASC and the left/right cell array selection bit, [LSB], set the Float Bitline signal, either FBLL or FBLR, high to turn off the bitline precharge devices in the selected array driven by FBLLOK or FBLROK and to turn off the bitline access devices in the deselected array. FBLLOK or FBLROK also enable the generation of the active LS' or RS' signal and the selected wordline is driven low. At this point, the P-channel storage cells in the accessed row dump their stored charge onto the active

DRAM Fault Monitor Pinout Diagram

Figure 3.1.1

DRAM Fault Monitor Architecture and Block Diagram

Figure 3.1.2

Read Access Cycle form the Row Address Strobe

Write Access Cycle (all other clocks unchanged)

Figure 3.1.3

DRAM Fault Monitor

Figure 3.1.4

Figure 3.1.5

The VREF generator is used to generate a voltage that is stable between the limits of 3.5V and 3.7V, and only is active if the voltage strays beyond these limits.

DRAM Fault Monitor

Figure 3.1.6

Figure 3.1.7

Figure 3.1.9

bitlines. Every pair of bitlines drives a single sense amplifier and only one of these bitlines is active during initial pre-sense portion of the read cycle. The other remains at its precharged level. The sense amplifiers are shared by columns from both cell arrays. The active low wordline charges a dummy timing wordline high, SAE (Sense Amplifier Enable), to synchronise the cell access operation with the generation of bitline sense control signals. This ensures that the accessed cells have dumped their charge onto the bitlines. Sensing commences following the activation of SAE. The self-timed generation of sensing signals using a dummy bitline configured to emulate an actual bitline is an example of a technique that is widely used to ensure proper operation of different configurations of the DRAM with regard to the wordline cell access portion of the active cycle. In this case, the time between the wordline activation and the sense amplifier activation tracks with increasing numbers of wordlines (cells per column) and bitline capacitance.

SAE generates two signals, SAP (Source enable Amplifier P-channels), and SAN (Source enable Amplifier N-channels), which activate the full CMOS sense amplifier to amplify the initial bitline differential, which is on the order of 200mV and varies with bitline capacitance, and then restore the bitlines to full CMOS voltage levels. The active low signal SAE* is used to keep SAN and SAP equalised to a midrail potential during precharge periods when the DRAM is not accessed. It is also used to gate an N-channel device on the sense amplifier output nodes which precharges the sense lines and bitlines to a midrail level before the P-channel precharge networks are connected to the bitlines at the end of the cycle. Two low threshold inverters are driven by the two sense lines at the bottom of the cell array nearest the row decoders to trigger the column access operation. Only when one of these sense lines, SL1 or SL1*, has reached approximately V_{SS} , will the corresponding Low BitLine signal, LBL1* or LBL2* be driven high. This is another example of the self-timing technique which is used to synchronise two portions of the DRAM cycle which can vary with the size of the DRAM: in this case, bitline restoration

and cell rewrite followed by connection to the data I/O circuitry. The triggering interlock generates CS' which enables the static Y-decoders, provided [CAS], the Column Address Strobe, is in its active high state. The column or Y-decoder outputs, Yi, connect the selected sense amplifier to the databuses which in turn drive the I/O latches. The read data appears at the output pin after flipping the I/O latch. A revision of this DRAM design is to contain fully dynamic column decoders which are identical in design to the row decoders and are adapted to fit in the proper column pitch.

To end the read access sequence, [RAS*] is driven high. RASC and RASB reset the active wordline to a high level. [RAS*] also pulls SAE low to turn off the sense amplifiers and to turn on the equalisation devices for the SAP and SAN signals, and the sense amplifier output nodes. After SAE* is driven high, the bitline precharge devices are then enabled via the Float BitLine signals and their buffered versions, and the bitlines and sense lines in the previously active array are precharged to the reference voltage, V_{REF} . Concurrent with the bitline precharge sequence, the LBL1* or LBL2* signal that was previously high is reset low and the column decoders are turned off to isolate the I/O databuses from the previously activated sense lines and bitlines. The device now remains in a precharging state to bring the bitlines and sense lines to the desired reference potential prior to the assertion of new row and column addresses and the row address strobe. The output data, [DO₀₋₃], is retained in the output data latch.

The write cycle for the device is slightly different. When the row address strobe is asserted low, the [WE] write enable signal is also driven high and the desired input data is presented to the [DI₀₋₃] pins. The [WE] high state automatically enables the column decoding sequence in parallel with the wordline selection and the input data is propagated through the input latches to the databuses and the selected columns. These signals override those produced by the accessed cells so that when the sense signals are enabled the bitline differential is that which is produced by the complementary input data signals and not the

accessed cell. The bitlines and the accessed cells are written. The reset sequence to precharge or standby mode is the same as described above. This write cycle design has the drawback that it is not possible to perform page-mode writes on the device but only page-mode reads.

During a refresh period, the row decoder must cycle through all of the wordlines in the two arrays. No column decode or active [CAS] period is required.

This self-timed design architecture lends itself very well to modular DRAM generation. The precise generation of sense clocks, column decode clocks, and the reset of the device at the end of active cycle are all dependent upon the wordline length, bitline length and their associated varying capacitances with DRAM module configuration. This design approach is contrasted by the circuit design and architecture that are prevalent in commodity DRAM designs used in high density, high performance DRAMs. Commercially available DRAM devices are very different from the designs that are employed in embedded RAM applications. The differences between the two families of related memories lie in essentially three areas: manufacturing process complexity, architectural and circuit design techniques, and application environment.

In terms of manufacturing complexity, DRAMs have been pushing the capability of chip manufacturers in steadily miniaturised photolithography and patterning technology, as well as their ability to build structures deeper into the silicon substrate. Submicron minimum gate lengths and trench storage capacitors in multi-megabit DRAMs are becoming an industry standard [Lu89]. Embedded DRAM, on the other hand, as exemplified by this DRAM that has been analysed employs no memory-specific processing steps in the 1.5 μ m double level metal CMOS process. Packing density is achieved without intentionally "breaking" layout design rules. The resulting flexibility to include a dense memory capability on-board an ASIC chip, typically a low volume IC, that is manufactured in-house is greatly leveraged. In such an environment, the only other alternative memory is

DRAM. This embedded DRAM approach, by virtue of its cell design concept, offers the density increase without the need for exotic manufacturing capability. The use of a one Megabit embedded DRAM alongside a gate array incorporated on one die is an exception [Sawa88]. This commercial DRAM preceded by two years the embedded application, which only became possible through the maturity of manufacturing process and yield enhancement techniques.

The application environment for commercial full-chip DRAM is greatly different than that for embedded DRAM. Commercial chips are exploited for their mass storage capability in main memory of computers, processing systems, and other system level applications. High density DRAM with fixed word widths and JEDEC standardised pinout configurations occupies more sockets in such system designs than any other commodity chip. Embedded RAM is utilised on-chip for quite different goals. The use as high speed caches in DSP processors, as FIFOs, as internal scratchpads and stacks, and as a means of eliminating the need for relatively lower density off-chip memory. In typical applications in telecom ICs, word widths up to 40 bits and densities from 16 to 32K bits for an embedded memory are common. In this applications arena, off-chip memories would take up much space at the board level, a situation that would be exacerbated by the requirement of several different smaller but varying memory configurations. The embedded RAM offers flexibility of usage within an ASIC.

The most significant deviations in philosophy of design between embedded and commercial DRAM occur in architectural and circuit design complexity. Commodity DRAMs offer an incredible array of common functions that are standards for almost all devices in the industry. Static column decoding techniques, nibble mode operation, RAS-only and CAS-before-RAS refresh are only some of the added features to DRAM. Internally, the high density achievable results in a highly partitioned cell array architecture and subsequent control circuit design. Hierarchical decoding architectures involving

several levels of address predecoding is a prerequisite. The board level environments to which commodity DRAM devices are exposed necessitate advanced circuit techniques which include external data and address sensing and latching, the use of substrate bias voltage generators, and the generation of signals above or below external power supply levels [MOS86].

In the DRAM designs that the author has seen, including both NMOS and CMOS dynamic RAMs, the circuit design and layout is precisely tuned to optimise speed, power consumption, and die area for each design. This process and the complexity of design is repeated for every commodity DRAM product designed and manufactured. Embedded DRAM is a totally different species of device. Generated using module generation software which can produce layouts and simulation model information from existing building blocks and timing models in a tiny fraction of the time required to fully design a commodity DRAM design, embedded DRAM, or SRAM, is typically very simple in design and architecture. Using minimal partitioning and subdivision of cell storage arrays, a CMOS embedded DRAM typically utilises CMOS logic in addition to customised circuits such as address decoders and sense amplifiers to build the functional block of the memory. Very little design effort is required to convert, for example, TTL input levels to CMOS signal levels, because on-chip, full CMOS signal levels are available. Internal capacitances that the embedded macro must drive are at least ten times less than those loads which must be driven off-chip by commodity DRAMs. As a result, circuitry becomes less complex, the noise environment may be reduced, and the access times and cycle times of the embedded DRAMs become attractive and easily attained. The use of standard CMOS processes for ASICs with embedded RAM requires some advanced circuit techniques, and in the case of the BNR/NT design, the design of a reference voltage generator and a special timing block internal to the macro are necessary but are common to all embedded DRAM configurations.

There seems also to be a paradox with respect to the design of embedded DRAM.

Given the wide variation in density, word width, and application environment, one expects that the circuit design required will be extremely complex, even more complex than the tuned commodity DRAM, to account for this variation and ensure that the design is robust. This is only partly true. Self-timing techniques and some specialised circuitry are both required; however, the internal IC environment that the embedded RAM is exposed to and communicates with is greatly different than the external chip environment. In fact, to make the embedded macros simpler and more modular, timing interfaces within the ASIC operational regime are intentionally simplified. Configuring a DRAM as a pseudo-static RAM with normal chip select, read/write, address, and data I/O points simplifies both the macro designers' task and the macro users' task. The level of design stability and robustness needed in these two operational environments is widely different. The embedded DRAM is easier to design. The purpose of this analysis and research is to make it easier to access, to observe, and to test.

3.2 Fault Modelling Analysis

The CMOS Stuck-at is the most widely used fault model for CMOS digital logic. In respect to memory devices, this model has been applied to develop fault models for RAM peripheral logic, decoders, and the cell storage array [Abad83]. The application of testing algorithms to RAMs is intended to provide fault coverage for the expected manufacturing process defects and their detection as functional faults. To do this effectively requires the precise definition of a fault model for the memory.

Most of the previous work done in this area is based on static RAM fault modelling and algorithmic test application. Typical fault models include cell stuck-at '0' and '1' faults, cell coupling faults either between two cells or multiple cells, and pattern sensitivity faults [Abad83]. Procedures devised have also assumed that faults in the decoders, control logic,

Figure 3.2.1

and external data paths can be visualised as stuck-at faults or coupling faults in the cell array [Nair78, Dekk87]. However, in dynamic RAMs, this 'fault translation' is less clear. Considering the analog, nature of DRAM cell access, datapath midpoint potential precharge techniques, and the nature of folded bitline and shared sense amplifier techniques that are found in typical DRAM architectures [Ride79, MOS86, Died89], the direct translation of surrounding circuit faults to cell array faults is not possible in all cases.

Yield and defectivity data available from a representative silicon manufacturing facility indicate that approximately 85% of the components of yield loss are random in nature. Such manufacturing defects arise from random photolithographic spot defects (40%), thin oxide pinholes (15%), leakage due interlayer dielectric insulator defects (10%), and other miscellaneous yield losses (20%) such as metal breaks, missing contacts, or missing vias [Schv86]. To arrive at a precise fault model for this particular design which takes into account the possibility of such random defects and the prediction of the electrical fault behaviour, a CMOS Stuck-at analysis of the peripheral circuitry and an Inductive Fault Analysis approach [Dekk87, Shen85] to the cell array respectively, have been taken. The fault model is also be extended to include signal shorts in the critical DRAM datapath circuitry.

The cell design for the modular DRAM is shown in Figure 3.2.1. Minimum metal pitch for both the wordlines (Metal 1) and the bitlines (Metal 2) is achieved for the P-channel storage array which resides in the N-well. The repeated pattern characterising the DRAM cell array is the four cell neighbourhood [Haye75] which shares a polysilicon cell plate, two bitlines which are bused into adjacent sense amplifiers, and four wordlines, the two outer wordlines accessing the neighbourhood cells and the two inner wordlines simply passing through to activate the adjacent neighbouring neighbourhoods. In addition to the unrelaxed metal rules, the cell storage capacitor area defined is also at minimum spacing to maximise array packing density. The cell is large in area ($45\mu\text{m}^2$ in the $1.5\mu\text{m}$

CMOS process) to provide a storage capacitance of approximately 61fF per cell. There is no depletion implant or accompanying inversion layer compensation implant [Chat78] to increase the depletion capacitance of the cell, and the cell plate is held at ground potential to use an inverted surface of holes as the primary charge storage mechanism for the device. The thin oxide storage cell areas are at minimum spacing and the separation of the storage nodes is physically achieved via field oxide which underlies the polysilicon gate electrode. The thicker field oxide between the storage nodes makes the parasitic device threshold much more negative than that of a normal device. No N-channel implant is used as a channel stop under the field oxide, however, the field oxidation step results in N-type dopant (Phosphorus) pile-up at the Si-SiO₂ interface, effectively creating a channel stop.

The concept of a cell neighbourhood was first introduced by Hayes [Haye75]. Since that time, the concept and the analysis of cell neighbourhoods to detect pattern sensitivity within the neighbourhood has been advanced [Abad83, Srin77, Srin78, Kino86, Mazu87]. The primary consideration in analysing a cell neighbourhood is determining the susceptibility of a cell in the neighbourhood to having its stored data compromised as a result of reads from or writes to its neighbours and the data pattern stored in its neighbouring cells. The classical 9-cell von Neumann neighbourhood (base cell plus eight neighbours) and its 5-cell Moore neighbourhood subset are shown in Figure 3.2.2. The interaction of the cells in the neighbourhood can be the result of bitline or wordline shorts or leakage due to parasitic devices which exist as a result of the manufacturing process. The concept of pattern sensitivity within a neighbourhood must include the possibility of leakage currents developing between the cells. When such leakage currents are significant enough to cause the data in a cell to be destroyed, a neighbourhood pattern sensitive fault (PSF) is said to occur. For example, in the 9-cell neighbourhood, a worst case for a PSF would find the "base" cell storing a '0' and the remaining eight neighbours storing '1' levels or vice versa. Static Weighted PSF testing [Kino86, Fuji86] of cell neighbourhoods

Classical 9-Cell Neighbourhood and 5-Cell Subset **Figure 3.2.2**

and Sliding Diagonal style tests [Haye85] have been proposed as techniques to test the sensitivity of a cell to the data pattern stored by its neighbours, essentially the number of stored '1's surrounding a stored '0' and vice versa.

In the case of the cell design implementation for the fault monitor chip and the modular DRAM that it is based upon, a slightly different neighbourhood exists. Instead of the classical 9-cell and 5-cell neighbourhoods that have been proposed in the past, the folded bitline architecture of this DRAM design and the subsequent tightly packed cell layout give rise to a 4-cell neighbourhood as shown in the neighbourhood diagram, Figure 3.2 3, and the cell layout. The four cell storage nodes are contained within a large square area of polysilicon cell plate. On the basis of the analysis of the parasitic subthreshold device operation considerations undertaken in [Died89] using MINIMOS, a device simulator, the necessity of considering static PSF faults for the cell is found to have weak physical foundations. However, until PSF sensitivity is proven not to occur under test conditions, and not just simulation and analytic conditions, tests which sensitise PSF faults in the device must be exercised.

Inductive Fault Analysis (IFA) [Shen85] is an effective means of systematically assessing the defect mechanisms and the resulting functional behavioral characteristics of an integrated circuit. Applying it to RAMs has been shown to be effective for fault modelling static RAMs [Dekk87]. Due to the characteristically large portions of the macro which actually store data, and the typically dense geometric structures utilised in the cell array, this analysis technique is a powerful tool for understanding the failure mechanisms which arise due to manufacturing defects. Gross manufacturing defects are not considered in this analysis. Since it is expected that embedded DRAM macros will be the yield limiting factor in many ASIC designs, understanding the mechanisms of failure in the most densely packed areas of the design is critical to developing a test strategy for the macros. The stuck-at analysis and IFA of the fault monitor chip are detailed in [Died89].

DRAM Fault Monitor Cell Decoding and Neighbourhoods

This sequence of sense lines and cell access from wordlines repeats. The four cell neighbourhoods are shown outlined.

Figure 3.2.3

3.3 DRAM Fault Model Definition and Testing Strategy

To use the information obtained from the stuck-at and inductive fault analysis of the DRAM, a fault model for the DRAM may be defined and as such include the fault classes that have been found in the analysis. Tests may then be defined and elaborated as the tools by which the fault model that is proposed may be fully covered. Exercising these tests on the DRAM fault monitor chip will determine the algorithm(s) which provides the maximum fault coverage and is most convenient for the practical implementation of Built In Self Test. As the device matures in design and in fabrication, the simplification of the fault model becomes beneficial in simplifying the complexity of test algorithms that must be applied to the DRAM.

3.3.1 Fault Model

Fault models that have been considered for SRAMs such as in [Dekk87] are not complicated. The process of IFA that has been applied to the cell array design of the fault monitor DRAM and the CMOS Stuck-at model that has been exhaustively applied to the peripheral and control circuitry of the memory [Died89] have illustrated the complex failure modes possible for this DRAM. The resulting fault model for the DRAM is presented below with an example of a defect drawn from the analysis that would exhibit the described faulty behaviour for each fault listed. All of the faults discussed in the Stuck-at and Inductive Fault Analyses of the DRAM may be classified under one of the following fault models [Died89].

1. Cell Stuck-at Faults

An example of this fault is a cell storage node shorted to a wordline resulting in stuck-at

'0'.

2. Wordline Stuck-at Faults

A wordline may become stuck-at '0' causing a row of cells to be continually accessed. A wordline stuck-at '1' will result in no cell access. This fault model also includes the case of no cell access due to a decoder open fault for example.

3. Address Uniqueness Fault

Due to stuck-at '1' faults on address inputs, two or more wordlines or column access signals can be activated simultaneously. In effect, two logical addresses affect the same storage cell.

4. Cell Pair Coupling

This fault model describes the coupling between a pair of cells, say i and j, in an N cell memory. Both write and read operations can cause coupling between cells in the DRAM. There are eight possible combinations of coupling faults for write operations, and four possible combinations for read operations between two cells. Note that read operations in DRAM are the same as write operations except that the cell provides the bitline differential signal for the read operations and this differential is driven from the data I/O drivers for write operations. These combinations are listed below.

Cell i	Cell j	Transition	Operation
'0'	'0'	Cell i '0' --> '1'	Write
'0'	'0'	Cell i '0' --> '0'	Write/Read
'0'	'1'	Cell i '0' --> '1'	Write
'0'	'1'	Cell i '0' --> '0'	Write/Read

'1'	'0'	Cell i '1' --> '0'	Write
'1'	'0'	Cell i '1' --> '1'	Write/Read
'1'	'1'	Cell i '1' --> '0'	Write
'1'	'1'	Cell i '1' --> '1'	Write/Read

A fault in which address uniqueness is violated for two memory locations such that they are addressed simultaneously for every access to one or the other cell will result in coupling faults. A short between two adjacent wordlines or bitlines will also result in coupling faults. A short between two or more cells in different rows or columns. The classical fault model presented above is an abstraction. It states that transitions in one cell may cause a transition from '0' to '1' or '1' to '0'. These two fault examples given above are not examples of transition faults but rather they are state coupling faults. Stuck-at and inductive fault analysis of the DRAM provides no physical basis for cell transition coupling. State coupling, a fault model which implies that two cells are coupled if a change in state in one of the cells results in a corresponding change of state in the coupled cell, and vice versa, is a more simple and realistic model of cell coupling and will be used here. The complexity of the resulting test algorithms can be reduced [Abad83].

5. Cell Neighbourhood Pattern Sensitivity Effect

(a) The four cell neighbourhood that has been discussed is subject to pattern sensitivity faults. In a four cell neighbourhood, there are sixteen possible states and each of the four cells must be verified as capable of storing both a '0' and '1' for the eight states of its surrounding three neighbours. An example of a pattern sensitive fault in the memory would be the corruption of a cell in a four cell neighbourhood from '0' to '1' as a result of its three neighbours each storing a '1' level. Such pattern sensitivity can arise due to inadequate cell plate voltage stability which increases inter-cell leakage current, or leakage

current into the N-well in which the P-channel storage cells reside.

(b) Another neighbourhood exists in DRAM that has not been introduced as a neighbourhood in previous fault analyses and modelling of DRAMs [Abad83]. In the same way as four cell neighbourhoods exist due to layout realisation of the memory array, a neighbourhood of cells attached to a single sense amplifier, a Column Neighbourhood, exists due to the manner in which the cells are accessed. The two bitline column is a neighbourhood. Coupling between the cells in this neighbourhood is more subtle than in the von Neumann, Moore, or the four cell fault monitor neighbourhoods discussed previously. In this neighbourhood, coupling can occur when one cell is connected to the bitline via an active wordline while all other cells are in a store mode. However, just as subthreshold coupling in the other neighbourhoods can conceivably occur, the inactive cells may influence the bitlines as they are driven by the accessed cell and sensed by the differential sense amplifiers in the course of read operation. This type of bitline imbalance fault has been introduced before [Haye85] but conceptualising the cell interactions by a **neighbourhood** interaction is new.

6. Bitline Pair Loss of Signal

An example of this fault is a bitline shorted to V_{REF} . During read cycles, the sense amplifier "sees" no bitline differential and realises in one state or the other based upon the imbalance that, for example, exists on the bitlines due to previous cell accesses or layout induced capacitive imbalance. A cell stuck open fault due to a missing metal 1 wordline to polysilicon access transistor gate contact is also part of this class of faults.

7. Bitline Precharge Degradation Effect

A bitline precharge degradation effect results from a stuck-at V_{CC} condition on an FBLR'

signal, or even an open condition on the V_{REF} precharge voltage. Bitline precharge is achieved only by equalising the bitlines and sense lines at the end of each cycle and depends on accessed data history.

8. Data Retention

Data retention in the DRAM is a fault and can contribute to pattern sensitivity in the cell neighbourhoods if natural processing defects have induced high leakage currents which result in short data storage times in the cell.

3.3.2 Proposed Test Algorithms

In arriving at a test strategy for the DRAM, or for any memory, that is to be relatively simple and which also lends itself to the many requirements of BIST, it is desirable to implement as few as possible test algorithms, or at best a single algorithm that covers the entire fault model for the memory. Because of the very diverse nature of the DRAM fault model that has been defined, to meet these requirements is extremely difficult and challenging. Existing algorithms and techniques for memory test [Abad83, Nair78, Dekk87, Nair79, Nicc85, Mari82, Silb87, Kino86, Haye85] provide excellent examples of test algorithms that provide the requisite fault coverage, but are difficult to integrate into a unified testing algorithm suitable for BIST for embedded DRAM. Tests that must be performed to cover fully the fault monitor DRAM fault model do not diverge greatly from this theme. However, the advantage in this instance is that the DRAM that must be tested is a fault monitor. By default, full external access to the memory exists and the ability to apply any species of test also exists. Through this process, the susceptibility of the DRAM to faults covered in the fault model can be analysed. Faults may then be removed from the fault model or appropriate fabrication or design changes may be made to minimise the

susceptibility of the device to those faults which would not be covered by a chosen algorithm for BIST.

DRAMs, unlike SRAMs or ROMs for example, require periodic refresh due the leakage of charge from the cell storage capacitor. In the specification of test algorithms, the refresh requirement is considered independent from the test. Wait states which suspend refresh cycles would be advantageous from a testing point of view, however, with regards to BIST, this is difficult to achieve without increasing the complexity of the BIST circuitry. The proposed testing strategy and detailed description of the algorithms follows below.

3.3.2.1 March Test Sequence

March tests are powerful tools for fault detection in memories [Abad83]. Depending upon the fault models proposed and the memory circuit design, march tests can detect a variety of faults both contained within the cell array and also present in the surrounding decoder and peripheral control circuitry [Abad83, Dekk87, Nico85, Mari82]. In the case of the fault monitor DRAM, the analysis performed by the author on its circuitry has demonstrated that there are faults which can occur in the periphery which are not directly translated into cell array stuck-at faults, such as pattern sensitive faults and column neighbourhood interactions.

In [Dekk87], it has been proposed that in every march element which performs read-modify-write sequences, a read verify after the write should be performed since a data latch in the data I/O path can act as a memory storage element. The read-verify determines whether a cell has been written or not. If it has not been written, the previous cell storage data will appear at the output. Conversely, if the cell has been written, the output should remain unchanged. This adds to the number of access cycles required to execute the march sequence but verifies proper data propagation to and from the cell array.

March Test Sequence for DRAM Testing

Cell	Sequence			
	0	1	2	3
0	W0	ROW1W0W1	R1W0R0W1R1	R1W0W1W0
-	/	/	/	/
-	W0	ROW1W0W1	R1W0R0W1R1	R1W0W1W0
k	/	/	/	/
-	W0	ROW1W0W1	R1W0R0W1R1	R1W0W1W0
n-1	W0	ROW1W0W1	R1W0R0W1R1	R1W0W1W0

Notes:

1. RC refers to reading the memory location and expecting 0 level. Similar interpretations apply to R1, W0, W1.

Figure 3.3.2.1.1

A single march test exists to cover faults 1 through 4 of the DRAM fault model and is shown in the Figure 3.3.2.1.1 [Nico85]. This test also will cover the classical cell transition coupling model that has been simplified to state coupling for the DRAM fault model. A simpler march test, such as the test proposed in [Dekk87], a test which is a minor embellishment to the MATS marching test sequence [Abad83], will also suffice in covering faults 1 to 4 in the DRAM fault model.

3.3.2.2 Walking Row Test Sequence

The remaining faults in the fault model for the DRAM that are not covered by the march test sequence require more complex test patterns to detect the susceptibility of the device to such faults. The Walking Row sequence is proposed to cover faults 5, 6, and 7 in the fault model.

The bitline common mode fault and the bitline precharge degradation effects in the fault model arise due to various and complex faults in the memory array and the control circuitry. In both cases, the sense amplifier is required to detect a much reduced differential signal than it was designed for, "sense" no differential bitline signal at all, or operate out of its normal design regime. In detecting the possibility of these fault classes affecting the memory operation, it is required to exercise the memory in such a way as to sensitise it to the fault.

The Walking Row test pattern operates as follows. For each column of the memory (in this case two sense lines, two bitlines and one sense amplifier) all logic '0's (or '1's) are stored in each cell. With the folded bitline architecture this translates into the cells on one bitline storing all '0's ('1's) and the other bitline's cells storing all '1's ('0's). For each cell on the column, a '1' ('0') is written. All of the other cells are read in sequence after each write to observe the effect of the write operation on the column, and then the original cell is

read again to verify that it still contains the written '1' ('0'). This is repeated for all cells. For cells on the positive sense logic bitline, a '1' is walked through a background of '0's and for the complementary bitline a '0' through a background of '1's. The length of this test for an n-word memory is $2n^{3/2} + 5n$ memory cycles.

The effectiveness of this test is dependent upon the number of rows per column. The greater the number of rows per column, the greater the number of accesses per column and therefore the greater is the sensitisation of the fault. For varying sizes of DRAMs, a minimum, or maximum, number of accesses per column must be specified. This could take on values of n, the number of cells in the memory, or even be set at the maximum number of wordlines per column. The greater the number of opposite polarity reads done in a column after a write operation is performed, the greater is the sensitisation of the column neighbourhood faults. Examples of faults and the utility of the Walking Row algorithm in sensitising these faults are included in the discussion below to illustrate both the behaviour of the DRAM and the power and complexity of the test.

For the bitline common mode, or loss of bitline differential signal fault, the behaviour of the sense amplifier in each column is indeterminate. When presented with no differential signal to sense, the sense amplifier will move in a direction depending upon bitline precharge voltage imbalance or even layout imbalance. When the Walking Row pattern is applied to each column in a folded bitline DRAM, a 5b fault model neighbourhood, the actual cell storage levels on the paired bitlines are opposite in polarity, as previously described. This sets up the worst case conditions for cell subthreshold leakage onto the bitlines since on one bitline, stored '0's will pull the potential slightly below the precharge voltage level after the precharge networks are released and on the paired bitline, stored '1's will exhibit the opposite effect. As a '1' or '0' is walked through a background cell pattern of opposite polarity, the robustness of each cell and sense amplifier in the column neighbourhood in counteracting the resulting imbalance is tested. When a stuck open cell is

tested for example, the induced bitline imbalance from all other cells connected to the column bitlines will be at a maximum: the sense amplifier will realise in the state opposite to the expected state. This same behaviour is expected for many of the defects that can induce the bitline common mode voltage class of fault. Testing the device at full speed and with close to minimum [RAS*] high periods for bitline and datapath precharge sensitises the DRAM columns to this type of fault. Devices with many rows and long bitlines will be especially indicative of the susceptibility of the DRAM to faults of this class.

The precharge degradation expected in a cell array due to, say, a stuck-at '1' on FBLR signal [Died89] is another example of a slightly different but related fault to the bitline pair loss of differential fault. One entire array of cells will not be accessed and the other array of cells will be subject to degraded precharge levels. At the end of each row cycle in a fully working device, the bitlines and sense lines are first equalised by N-channel devices in each column and then are precharged to V_{REF} via a network of three P-channel devices located in each column. Operating the device at close to minimum cycle times and with a Walking Row test pattern will again sensitise each column to the imbalance in bitline precharge voltage levels. Consider for example, the case of a single defect which causes the P-channel precharge network in a column to be always off, due to a missing intermetal via to connect FBLROK [Died89] to the devices. The result for this column is that at the end of each active row cycle, during normal read/write access or even during refresh, the bitlines will be equalised only from V_{CC} and V_{SS} full rail levels to approximately $V_{CC}/2$. The two complementary sense lines will be equalised to within an N-channel V_{TN} of each other with no assurance that the voltage on each will be identical. This residue of previously accessed data will be carried over into the next row cycle when the succeeding cell access occurs. It is clear that the Walking Row test with its series of repetitive reads of one data polarity followed by the occasional read of the opposite data polarity will sensitise the column to the degradation of precharge voltage levels. Minimum RAS cycle times will

be optimum in sensitising the fault. Long precharge times will also be optimum, at least in this case, because the bitlines and sense lines then have the opportunity to leak away towards one supply voltage rail or the other.

The Walking Row algorithm is very powerful for sensitising and detecting the above faults. It may be modified and simplified to reduce the number of operations that must be performed on the entire memory by selecting one particular column in the DRAM macro in each cell array and performing the following operations on only two cells in that column. With the entire memory initialised to all '0's (or '1's) write a '1' in cell *i* on this column. Read cell *j*, which contains a '0', for a period of time, say *n* accesses where *n* is the number of bits in the entire memory. Return to cell *i* and verify that a '1' can be read. This clearly sensitises the bitline precharge degradation fault due to the repeated accesses on the single column. However, if a localised defect such as a missing contact from FBLROK to a precharge network [Died89] exists in one column only (a spot defect), depending upon which column is selected, one may or may not detect the fault in the defective column. If parallel access to the columns existed, this algorithm would be able to test all columns in each array simultaneously to prevent the loss of fault coverage. Similarly, a cell open fault which results in a bitline pair loss of signal will not be detected by such a reduced algorithmic sequence.

3.3.2.3 Data Retention Test

The required stimulus for data retention in a DRAM is to insert a WAIT state or states in the normal refresh interval. This can be done on any algorithmic memory test setup and the WAIT state duration may be varied to characterise the leakage effects in the DRAM cell arrays.

3.3.2.4 Tests for Neighbourhood Pattern Sensitivity

Neighbourhood pattern sensitive faults, fault models 5a and 5b, are included in the fault model for the DRAM. The problem that exists in sensitising the cells of the memory to neighbourhood PSFs has been the number of memory reads and writes required. In the case of the folded bitline DRAM architectures, the topology of the array and the required row and column decoding sequences exacerbates the problem. To obtain precise coverage of PSFs, a relatively complex deterministic addressing sequence is required given the DRAM decoding scheme and bit organisation. This sequence is expanded or shortened depending upon the organisation. A neighbourhood PSF analysis algorithm to test static PSF sensitivity has been conceived and is shown in Figures 3.3.2.4.1 and 3.3.2.4.2. PSF4 is in part based partly upon the concepts embodied in the algorithm proposed in [Kino86].

An analysis of the Walking Row Test indicates that it can be used successfully to test the worst case four cell neighbourhood sensitivities in addition to the column neighbourhood fault class. Because cell arrays in this DRAM design are organised in bit-blocks or subarrays such that each bit of the stored memory word is column decoded independently of each other bit, and during the Walking Row test, only one column is accessed and all others are quiescent (that is, they are not written but only read), the worst case static PSF patterns for four cell neighbourhoods are exercised. For example, if we are walking a '1' through a background of '0's, a '1' is walked through '0's on the positive logic bitline and vice versa for the negative logic bitline. With adjacent columns quiescent, the written cell may or may not be the worst case cell (that is, it stores a '0' ('1') while its three neighbours store '1' ('0')) but its neighbour on the same bitline in the four cell neighbourhood may be the worst case cells for a given write operation. Figure 3.2.3 clarifies this neighbourhood analysis. This is not a problem because the Walking 1/0

PSF4 Test for DRAM Static Pattern Sensitive Faults

Memory neighbourhood configuration

0,1,2,3 refer to four cells and test modes for the algorithm.

1	2
0	3

1. Initialize memory to all 0's.
2. (a) Set mode $i=0$
 - (b) For all cells $j=i$, exercise minimum write sequence to take the three cells through the 8 possible states and verify mode cell contents after each write.
 - (c) $i=i+1$
3. Repeat 1 and 2 by initializing memory to 1's and beginning minimum write sequence with neighbouring cells all in a 1 state.

Algorithm Length is $34n$ where n is the number of cells in the memory.

The minimum write sequence refers to the fact that there are three neighbouring cells for each base cell. To progress from a 000 state to a 111 state can be accomplished via a minimum of 8 write operations as is obvious from a Karnaugh map in three variables.

Figure 3.3.2.4.1

PSF4 Test for DRAM Static Pattern Sensitive Faults Simplified

Memory neighbourhood
configuration

0,1,2,3 refer to four cells
and test modes for the
algorithm.

1	2
0	3

The PSF4 algorithm may be simplified and reduced in length. By performing an Eulerian tour minimum write sequence, conditions for the testing of other neighbourhood static PSFs arise. The chart below summarizes the sequence of writes and reads required to complete the algorithm for a given Eulerian tour.

States of Cells (modes) in Neighbourhood	Write Operations	Read Operations
0 1 2 3	0 1 2 3	0 1 2 3
0 0 0 0	W0 W0 W0 W0	- - - -
0 0 0 1	- - - W1	R0 R0 R0 R1
0 0 1 1	- - W1 -	R0 R0 R1 R1
0 0 1 0	- - - W0	R0 R0 R1 R0
0 1 1 0	- W1 - -	R0 R1 R1 R0
0 1 0 0	- - W0 -	R0 R1 R0 R0
0 1 0 1	- - - W1	R0 R1 R0 R1
0 1 1 1	- - W1 -	R0 R1 R1 R1
1 0 0 0	W1 W0 W0 W0	- R0 R0 R0
1 0 0 1	- - - W1	R1 R0 R0 R1
1 0 1 1	- - W1 -	R1 R0 R1 R1
1 0 1 0	- - - W0	R1 R0 R1 R0
1 1 1 0	- W1 - -	R1 R1 R1 R0
1 1 0 0	- - W0 -	R1 R1 R0 R0
1 1 0 1	- - - W1	R1 R1 R0 R1
1 1 1 1	- - W1 -	R1 R1 R1 R1

By executing the algorithm in this sequence, all Static PSF patterns for each neighbourhood are sensitized and tested. The model assumes no effect from read accessing cells in a neighbourhood.

The algorithm length is $81N/4$ memory accesses for an N bit memory.

This algorithm also contains coverage of the faults normally detectable using march algorithms. Each "mode" is actually a series of addresses in the memory which correspond to the neighbourhood cell designation. By performing writes on selected portions of the memory and then cycling through the addresses to read all cells, the sequence is simple. A memory neighbourhood map is all that is required to determine which addresses correspond to which memory write "mode".

Figure 3.3.2.4.2

algorithm will read the worst case cell, whichever it may be, and detect a fault if one occurs. With the column decoder replicated for each bit of the memory word and with the Y-addresses acting as the most significant bits, every column and every neighbourhood cell is tested independently for the worst case neighbourhood patterns. There is no loss of fault coverage at the logical boundary between the bit arrays because the adjacent columns are column Y_{i-1} for one bit and column 0 for the adjacent bit in a replicated 1-of-i Y-decoder circuit. One will be quiescent while the other is being exercised.

A test which will allow the worst case neighbourhood PSF sensitisation to be exercised is again the Walking 1/0's Test. This test is similar to the Walking Row Test and operates as follows. For each cell given a background pattern stored in the entire memory, the cell is read and then written with the opposite polarity of data. The entire memory is read to determine the effect of the write operation on all other cells, and the cell is read to verify that the memory read had no effect on it. The cell under test is then rewritten with the background data, the succeeding cell selected, and the sequence is repeated. This test is also an excellent reference test for the detection of other faults in the fault model including those covered by march tests [Abad83]. However, the approximate length of $2n^2+6n$ is prohibitive and BIST circuitry implementations will dwarf small embedded memories in many instances depending upon the availability of counters and logic surrounding the embedded DFAM macro which can be reused to implement the required algorithm.

An algorithm has been proposed which is heuristic in nature and which is designed to cover many of the pattern sensitivity fault detection sequences, but not all in its present form [Hoge86]. Based upon the modulo-3 addressing scheme described in [Abad83], this so-called MOD3 sequence performs a pseudo-march sequence between the three logical memory partitions which exercises all of the required neighbourhood transitions based upon the classical 5-cell Moore neighbourhood adjacency model. As an example applying the modulo-3 designations to the neighbourhoods of the DRAM fault monitor design given

Modified MOD3 Test for Pattern Sensitive DRAM Faults 68

Sep	Partition 0	Partition 1	Partition 2
1		ROW1R1	
2			ROW1R1
3	ROW1R1		
4		R1W0R0	
5			R1W0R0
6	R1W0R0		
7		ROW1R1	
8			ROW1R1
9	ROW1R1		
10		R1	
11			R1W0R0
12	R1W0R0		
13		R1W0R0	
14			ROW1R1
15	ROW1R1		
16		ROW1R1	
17			R1
18	R1W0R0		
19		R1W0R0	
20			R1W0R0
21	ROW1R1		
22		ROW1R1	
23			ROW1R1
24		R1W0R0	
25	R1W0R0		
26		ROW1R1	
27			R1W0R0
28	ROW1R1		
29		R1W0R0	
30			ROW1R1
31	R1		
32		R0	

Notes:

1. The test sequence is repeated twice. The address sequence within each logical partition is ascending on the first pass and descending on the second pass.
2. R0 refers to reading the memory location and expecting 0 level. Similar interpretations apply to R1, W0, and W1.
3. Memory is initialized to all stored 0's.

Figure 3.3.2.4.3

the row and column decoding scheme, results in some cases in which the four cell neighbourhoods containing all three designated logical partitions with one partition represented twice, and other neighbourhoods containing two groups of two cells that are in different logical modulo-3 partitions. However, analysis of the algorithm as applied to the classical 5-cell neighbourhood shows that if it is executed twice, once with addressing within the partitions done in ascending order, and the second time done in descending order, all transitions in the neighbourhood and states of the neighbourhood are exercised. Because the access to the memory is not parallel but sequential, and changes to cells in the same modulo-3 group occur in sequence depending upon the addressing, the same will be true of the 4-cell neighbourhood, even though cells of the same modulo-3 group are in each neighbourhood. Adding the read verify to every march element will alleviate the data I/O latch problem that was previously discussed, and will result in an algorithm that is $182n/3$ in length. Moreover, this algorithm covers faults 1-4 fully, and partially covers fault 5a [Silb87] in the DRAM fault model when it is embellished with the added read-verify operations and reversed addressing sequences. The detailed code for the MOD3 March Test algorithm is found in Figure 3.3.2.4.3.

3.4 Summary

Testing of the DRAM devices on the NIOSAID MS2200 tester, with the aid of E-beam voltage contrast probing of internal signals, has resulted in no experimentally verified conclusion concerning the DRAM fault model [Died89].

The following tests have been proposed to cover the fault models and expected failure mechanisms in the DRAM:

1. March Test;

2. Walking Row;
3. Walking 1/0's;
4. MOD3 Modified March Test; and,
5. PSF4 Simplified Test for Pattern Sensitive Faults.

All of the algorithms are suitable for separate BIST implementations of varying complexity. The most attractive candidates for the embedded DRAM test system implementation are the march and Walking Row tests. These two tests, as will be seen in the following chapter, are especially adaptable to the modified memory architecture, memory system test architecture, and memory reconfiguration scheme that have been developed.

4. An Architecture for Testability and Reconfigurability

4.1 Introduction

In the previous three chapters, the need for a test and reconfiguration approach for embedded memories, specifically DRAMs, has been simplified by developing a realistic fault model through the detailed analysis and testing of an example DRAM. Test algorithms have been designed and proposed to cover the DRAM fault model. Several architectural approaches have been analysed for the testing of high density memories, either externally or using BIST, and one approach to the testing of embedded SRAMs. What needs to be addressed next is a means, either through circuit, test algorithm, or memory system architectural innovation, or all three of these alternatives, of combining the test requirements dictated by the DRAM fault model with a scheme for reconfiguring defective memories in the multiple embedded memory ASIC application environment. There is no single solution to this problem, however, the test system and memory configuration that has been arrived at, and which will be elaborated in the course of the remaining discussion, will serve to illustrate the use of innovative circuit concepts as well as the application of the ideas that have been embodied in previous RAM test approaches and in standard RAM design.

A general discussion of several alternatives in modifying the DRAM will be followed by specific circuit alterations and design descriptions for the embedded memory macro. Based on the new architecture and functionality of the embedded DRAM, the global memory system will be specified. An analysis of the implementation of the memory test and repair flow, memory test algorithms, and redundancy will be presented.

Figure 4.0.1.1 Basic Embedded DRAM Architecture

4.0.1 Assumptions and Requirements for the Memory System

The analysis of the embedded DRAM used as the vehicle for developing the fault model and test approach indicates that it consists of the following essential functional blocks (Figure 4.0.1.1):

1. Cell array of variable size, single or multiple cell subarrays;
2. Row and Column Decoders;
3. Address Buffers;
4. Data Sense and Driver Circuits for reading and writing; and,
5. a standard Control Block for timing signal generation.

These functional components will not be modified so as to diminish mission mode functionality or performance, but they will be augmented to support test and reconfiguration functions (redundant cells must be added to the cell array along with a redundant decoder, for example) and will incorporate the necessary selection control circuitry to enable full mission mode and full test mode functionality and flexibility, and memory cell redundancy. To provide only a single feature to support a single test method is not an optimal solution. Rather, to provide a host of features which allow the user to select a global chip test integration of the memory, a local BIST or test access method, as well as any requirement for implementing redundancy is the goal. Redundant cell elements will remain an option for each and every embedded memory configuration.

The requirements that must be set out for the mission mode and test mode operation of the embedded memory and the surrounding test circuitry are:

1. testing is limited to RAM functionality and the detection and correction

of spot defects in the cell arrays. Parametric tests are not to be performed, however, operating the device at speed is preferable during test;

2. no laser or electrically programmable fuses are implemented for reconfiguring the DRAM redundant elements after test;
3. shared test access to all embedded DRAMs is required;
4. minimal loss of performance is incurred due to circuit additions or modifications;
5. no restriction exists on memory usage or row/column/word dimensions;
6. reasonable test circuit area overhead, both for the individual memory and the chip is expected to be incurred;
7. the test and reconfiguration system "intelligence" must be embodied primarily in the embedded block to make the system easily expandable and exploitable for different embedded memory distributions and configurations.

Based upon these global memory test system requirements, the possibilities which exist for implementing the requirements of test and reconfiguration of multiple embedded memories may now be elaborated.

4.0.2 Architectural Alternatives

Many viable options exist for enhancing the architecture of the memory to facilitate test and reconfiguration. Consideration must also be given simultaneously to the features or functionality that would be imposed upon the embedded memory test system and the test architecture necessitated for such features and desired functions.

First, consider the system level aspect of this problem. Clearly, there is a need for a

central Memory Test Controller, either external or internal to a given ASIC device which incorporates the multiple embedded memories. This MTC, analogous to an individualised memory BIST controller or finite state machine, will be required to select memories, generate test addresses and data, and control signals to activate all memories simultaneously or to activate the memories in parallel. If different modes of test are to be used, the MTC will be required to provide test status or mode lines that may either be decoded by the test control circuitry that is added to each embedded memory or bused in decoded form to each memory individually. The MTC may also be used to invoke a memory repair sequence following the application of the test algorithms to the embedded memories. Depending upon the circuitry or built-in "intelligence" which exists in each embedded memory macro, this may be done on a memory by memory basis or may be incorporated into each memory and performed simultaneously for all memories. With this array of options and alternatives, the MTC can become very complex and specific to only one ASIC embedded memory system very easily. In addition, multiple bit address and data buses, as well as memory selection decoding, which must be input to each embedded block will incur a large ASIC chip area overhead, solely devoted to signal busing. Reducing this bus routing overhead, in a memory system in which individual blocks are usually far removed from each other, using serial address and data generation by the MTC and serial address and data capture by the embedded memories is a viable alternative.

Given an architecture for the memory test control requirements of an embedded memory system, the specification for testing and reconfiguring the embedded memories becomes important. The test stimuli can override the peripheral circuitry of each memory to consider the chip memory organisation in whatever global memory configuration is necessary to simplify the unified testing of all memories. The memory organisation can be modified for testing, but there still remains the requirement to test all decoders, address buffers, control circuitry, and data I/O paths. These circuits may be, for example, tested

with a known functioning set of cells in each particular memory [Inou87]. It is more efficient to test the memory in a mode which resembles as closely as possible its mission mode operation. Additions of extra circuitry, the provision of different test stimuli, or utilising a particular test sequence may be simplified through this approach. Moreover, when redundancy is factored into this equation, the problem of testing a reconfigured memory (that is, memory with active redundant cells) falls naturally out when it is tested in a mode resembling its mission mode of operation.

In Chapter 3, a detailed discussion of test algorithms such as March, Walking Row, and others were put forth as candidates for providing test coverage of the DRAM fault model. Based upon the fault modelling analysis of the DRAM, these sequences were proposed to cover the fault model. It is effective to design a memory and a memory test system to implement tests more easily and effectively while preserving mission mode performance criteria. In such algorithmic sequences, parallel access to all of the columns (bitlines) of a memory will significantly speed up the application of tests and simplify the address or data input sequence requirements to execute these tests. By providing a second parallel test access port to the memory in this way, transforming the DRAM into a dual port memory for testing purposes, the test application time of typically lengthy algorithms, such as Walking Row, may be reduced while maintaining close to optimal coverage of the fault models.

A companion circuit innovation to this parallel access port is a CAM-style comparison circuit, similar in function to that shown in Figure 4.0.2.1, incorporated directly in the circuitry for this second port of access. The memory would thus include the read/write access circuitry and the error detection circuitry normally embodied in a memory test controller [Agar88].

Another possibility for parallel comparison techniques applied to each embedded memory block is the comparison of output responses of parallel identical arrays.

When the data on the complementary bitlines matches the data stored in the SRAM cell, the match signal bus is driven low. Otherwise, the match bus remains in its precharged high state and the next memory location is polled.

Figure 4.0.2.1 Content Addressable Memory Cell Concept

Depending upon the particular configuration of an embedded memory, this could take the form of two arrays being fully compared [Haye85], several identical array outputs compared simultaneously [Jarw88], or in the case of bit-block oriented architectures such as that in the modular embedded DRAM design indicated in Figure 4.0.2.2, the comparison of all data output bits. Single cell array memories must be compatible with the test access mode that is to be utilised. Noting the sequential nature of the test algorithms that have been proposed, the memory may also be modified so that no decoding or addressing is required at all to implement a March test or Walking Row test, for example, if parallel access to all columns via a second port is implemented. The row decoder may be converted to a shift register, which instead of requiring several binary address inputs, uses only shift control signals. This will simplify the test controller functional requirements and reduce its physical size and complexity as well as the need for wide address buses connected to all the memories.

Some of the memories in the embedded memory system will be capable of reconfiguration due to the inclusion of redundant cell elements based upon the memory yield relationships pertaining to a given IC manufacturing process. These elements may be organised as rows of cells, columns of cells, or even entire arrays of cells with the associated decoding and separate datapaths. The addition of rows of cells is attractive because it solves the problems of redundant datapaths and large area overhead in memories which have very wide word widths. However, the row access portion of a RAM, in general, is the critical access time component. Any delay which is added to a memory due to the deselection of a group of cells and the invocation of the redundant cell elements will add to this critical delay and will adversely affect the memory performance [Died87]. Column redundancy does not suffer from such a penalising restriction and added delay is tolerable, mainly because the long wordline attached to the large capacitance cell wordline access devices is replaced by a single pair of column access devices which are physically

2

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No 2)

Figure 4.0.2.2 Bit-Block Oriented DRAM Decoding Architecture

adjacent to each column decoder driver. Self-timed techniques for DRAM operation will mask the column decode delay and any addition to it due to the use of a redundancy scheme. Array redundancy, that is a redundancy technique which instead of having, for example, one extra column per bit-block array, combines all of the columns into a single array of cells with its own datapath and data I/O buffer, is perhaps the least restricting of all of the possible redundancy techniques. It also is the least effective because a defect and associated fault which affects two or more bits may be corrected if each bit-block cell subarray has column redundancy. However, if only one full array is invoked to correct a fault, all of the faults will not be corrected and the device will have to be rejected. Ease of adding redundancy and controlling the programming circuitry can be both a memory and MTC requirement. Moreover, detecting a fault, locating it, and reprogramming the decoder to use redundant elements when the faulty address is invoked is another shared functional requirement of the embedded block and the MTC. To simplify the MTC, this functionality, or "intelligence", may be embodied in each memory block so that the MTC simply applies test stimuli and expected responses, and each memory tests and reconfigures itself based upon the stimuli and response information which is processed locally.

It is conceivable in a distributed memory system that a central redundant group of cells, configurable to any desired row, column, or word organisation could be used. Each time a memory encountered an address which during test was found to be defective, it would be steered to this central bank of redundant elements. The problem with an approach such as this is that it would be shared by several memories, each of which may require a different word organisation of the cells. Moreover, it may be situated remotely with respect to some or all of the memories and this will definitely induce performance degradation. The physical signal busing, handshaking, and multiple access arbitration protocols required to interface each memory to the central redundancy unit would make such a scheme extremely undesirable, especially when compared to the area and speed performance achieved in a

distributed redundancy scheme in which each memory retains and controls its own redundant elements.

4.1 Global Memory System Test Access Architecture

The preceding discussion highlights the many possibilities for designing the optimum memory test controller, memory test and repair flow, embedded memory architecture, and the redundancy/reconfiguration scheme for a distributed embedded memory system. However, it is clear, based upon the need for flexibility and efficiency that such a system should be designed around the following concepts. The Memory Test Controller must be as simple as possible to make it portable to any ASIC with distributed embedded memories with minor modifications. The embedded blocks must incorporate the required circuitry to simplify test algorithm application, test response verification, and redundant element invocation. The redundancy and reconfiguration implementation must be easily added and removed from embedded blocks, and thus ignored by those embedded memories which do not use such circuitry. As noted above, these concepts determine the architecture of the memory, the memory test controller, the test/repair flow and test generation sequence.

4.1.1 Overview of Test Architecture Features

In the accompanying figure 4.1.1.1, an overview of the embedded memory test access system architecture is shown. The system is comprised of the three elements alluded to earlier, the **Memory Test Controller**, a **System Test Bus**, and a series of "**n**" **Memories** of differing configurations. All memories are connected to the shared test bus. All of the signals that make up the test bus are generated (memory input stimulus) or monitored (memory response) by the MTC. The test bus is used to step the memory test

An n-memory embedded DRAM system with shared test bus and system test controller.

System parameters N, M, R defined by one or more of the n memories.

Figure 4.1.1.1 Memory Test System Access Architecture

system through a predetermined **Memory Test and Repair Flow**, Figure 4.1.3.1.1, which is embodied in the MTC. In the course of this flow, the individual memories are tested and reconfigured in parallel according to the test flow states or modes, and their status is reported via the system test bus to the MTC.

The MTC has three major states or test modes. Parallel test mode, in which all of the DRAMs are tested in parallel via the parallel column access port, is the first mode. The remaining two states are serial test stimuli generation modes in which first, a test algorithm is applied and the embedded memories respond by reconfiguring themselves where possible when errors are detected, and second, the memories are retested using the identical test sequence to confirm the successful use of redundancy. In these two serialised test modes, all of the embedded memories are again tested in parallel using a serial addressing technique in which each memory block captures the addresses pertaining to its own address space and ignores addresses which select locations outside its logical address space. Following the second serial test sequence, the accept/reject status of the memory system is reported to the MTC.

The memory test system architecture is designed so that the MTC "sees" only a single memory whose characteristics are defined based upon the parameters of the individual memories which are connected to the test bus:

1. N: the largest address space of all of the embedded memory blocks;
2. M: the largest memory word width (number of bits of I/O) of all of the embedded memory blocks;
3. R: the largest number of rows (wordlines) in a memory of all of the embedded memory blocks.

The N parameter is used to determine the address space that must be addressed using the

serial address generation technique employed by the System Test Bus. Similarly, the M parameter is used to determine the length of the generated serial data pattern. R , the largest number of rows, is used to specify the length of the algorithmic control sequence for the parallel column access mode stimuli generated by the MTC. It should be noted that due to the wide variety and possibility of memory configurations in the embedded ASIC environment, the three parameters may be specified by three different memories in the n-memory system.

The following sections will illustrate in detail the operation of the Embedded Memory blocks, the MTC, and System Test Bus. Modifications and circuit additions required to bring about the desired functionality in the Memory Test System Architecture will also be presented in detail.

4.1.2 Local Memory Architecture

The accompanying block diagram, Figure 4.1.2.1, of the embedded memory architecture illustrates the additions made to the memory macro. Circuit modifications have been made to facilitate parallel read/write access to all columns for Parallel Test Access Mode. A shift register decoder is used to eliminate the addressing requirement for parallel test mode by bypassing the decoder trees of the wordline selection circuitry. A configurable column decoder which implements column redundancy and is soft programmable during the memory test sequence is also added.

The embedded memory macro also contains additional logic which is used to interface address and data buffers to the test controller via the memory system test bus. This logic consists of a serial address capture circuit which configures the address buffers during the test mode as a serially loaded shift register. A similar circuit is added to the data I/O buffers so that data may also be captured serially from the test bus. Address or data bits which are

Figure 4.1.2.1 Embedded Memory Test Architecture

out of the address space or greater than the data word width of an individual memory are logically shifted out of the respective shift registers and thus ignored. A parallel data I/O comparator is added at the output of the data I/O circuits of each memory so that direct comparison of response and input stimuli may be accomplished by the local memory. Local error information is then used to activate the memory reconfiguration control circuitry so that an error that has been detected will trigger the memory reconfiguration circuitry to correct the error.

The Local Test/Repair Control Block is the control circuit which interfaces directly with the test bus control and status signals. This block of control logic, which is active only during test mode, generates all of the local memory control signals to step the embedded memory block through the parallel test (data write and compare, shift register row decoder control, and error and status signal generation), generates the serial address and data capture control signals, and invokes the configurable column decoder when an error is detected. It inhibits reconfiguration of a memory if all of its redundant elements have been exhausted. An error after this condition naturally results in failure of the test.

The actual operation of the additional circuitry will be expanded upon when the Memory Test and Repair Flow is explained in later sections. Typical test control and status generation circuitry interfacing the embedded memory to the system test bus, and sample timing waveforms to illustrate the required operation of these typical control circuits are detailed in Appendix 1. The next sections will cover in detail the circuit innovations arrived at to equip the embedded memory with a configurable column decoder, a parallel test access, and a shift register row decoder.

4.1.2.1 Configurable Column Decoder

The design of the configurable column decoder is shown in Figure 4.1.2.1.1. A dynamic NAND decoder structure is augmented by two additions to the circuit: an SRAM cell or latch which stores the enable/disable status of each decoder element, and a pass gate and pulldown chain controlled by the decoder NAND gate to do the programming and reconfiguration of the column decoder. The column decoders require only two additional signal inputs, EN/DIS* and RESET. The operation of this circuit closely resembles the write and read accessing of static RAMs.

At the beginning of memory testing, the local decoder RESET signal is pulsed high to set all of the latches and enable all normal column decoders for the embedded memory for the succeeding test. During testing if a cell is found to be defective in a certain column that is currently accessed during the test sequence, that column is already selected via the address lines. The EN/DIS* line is driven low to write the latch and disable the column access device NOR driver. During a normal access cycle, EN/DIS* is precharged high. When the faulty column is selected for access, the NAND decoder opens the pass gate and the written latch pulls down the EN/DIS* line, like the read access of a static RAM cell [Dieu87]. This line then activates the redundant column access driver, essentially an inverter, or a NOR gate enabled by the EN/DIS* line and a column decoder enable line shared by all of the normal column decoders as illustrated in Figure 4.1.2.1.2, and the redundant column is activated. All other column decoders are deselected and unaffected by the EN/DIS* line.

If multiple redundant columns are required as shown in Figure 4.1.2.1.4, two or more separate EN/DIS* lines may be used. All EN/DIS* lines would be activated during the reconfiguration, but only the accessed decoder, the one controlled by only one of the EN/DIS* lines, would actually be programmed. If an EN/DIS* line had previously been

Figure 4.1.2.1.1 Configurable Column Decoder Circuit Design

Figure 4.1.2.1.2 Configurable Column Decoder Concept

Interlocked with Left/Right Cell Array Selection Address

Figure 4.1.2.1.3 Shared Configurable Decoder
for a Dual Cell Array Embedded Memory

Figure 4.1.2.1.4 EN/DIS Scheme for Inter- and Intracolumn Repair

used to program a redundant location, it must be externally inhibited by the reconfiguration control circuitry.

The redundant columns may also be similarly programmed for each parallel array of cells in a dual lobe memory architecture, for example, with right/left array selection and separate NOP decoder drivers. These concepts are demonstrated in Figure 4.1.2.1.3. Some reduction in column access speed may occur here depending upon the number of column decoders and the load on the EN/DIS* line, but this is not the critical access path in the DRAM [Died87].

EN/DIS* is driven by a tristatable driver during test/reconfigure mode so as to allow the dual test/mission mode function of the EN/DIS* signal. Typical circuit realisations to control EN/DIS* circuitry are detailed in Appendix 1. Full simulation of the configurable column decoder has verified its concept and functionality and the simulation is documented in Appendix 2.

4.1.2.2 Parallel Test Access Port

The parallel test access port is an additional access technique which allows all columns, or bitline pairs, in a cell array to be accessed simultaneously for reading and for writing. The basic read operation is not a true read operation, but a parallel compare operation not unlike the process which is performed by a Content Addressable Memory (CAM).

The parallel test port circuitry operates as follows. After the sense amplifiers have driven all of the bitline pairs in a cell array row access cycle to full CMOS levels and the bitline levels have been restored, the parallel port access devices, shown in Figure 4.1.2.2.1 to be driven by a signal named PTST, are turned on. After this occurs, the complementary bitline data signals are steered to two transistor pulldown chains. Each chain consists of another transistor driven by the complementary Expected Data signals, ED

Notes: BL1 and BL2 are bitlines from the cell array accessed by the signal PTST
 ERROR is a shared bus for all parallel access comparators
 ED and EDB are complementary data used to write to the columns or to perform a data response comparison

Figure 4.1.2.2.1 Parallel Test Access Port Design

and EDB^* , which are normally low. One of these two signals is driven high. If the output and expected data match, the precharged MTCH node remains high. If the output and expected data do not agree, this MTCH line is discharged through the pulldown chain which has both transistors turned on. This activates a driver which pulls a precharged low ERROR line high to indicate that a read error has been detected in one of the columns. For the next read cycle, PTST is disabled, the expected data signals are reset to low logic levels, MTCH is precharged high, ERROR is precharged low, and the transferred bitline buses which drive the CAM-style comparator circuit are precharged high.

A parallel test write operation is accomplished in a similar manner except that the signal PWRT is driven high with the PTST signal. ED and EDB^* are activated to develop the necessary bitline differential which overrides the wordline accessed cell differential and is sensed and amplified by the sense amplifiers as a normal write cycle would take place in this particular DRAM design [Died89].

The bitlines "see" only the added capacitance of the pass gates of these circuits during mission mode. No series devices between the sense amplifiers and the bitlines is tolerable in order not to reduce mission mode performance of the sense amplifiers. The circuit requires six additional signals for read and write control, data, and pre-charge. None of these signals is used in mission mode and are controlled solely by the test control logic. This is documented in Appendix 1. Simulations of this circuit for read comparisons of matching and non-matching data, as well as for parallel write operations, have been performed and they have verified this circuit. Appendix 3 contains the simulation analysis results.

The layout technique that is required to implement such a circuit is not particularly difficult. By combining this parallel port circuitry with the sense amplifiers and normal access devices and laying each out in the layout pitch of four bitlines and not the normal two bitline column pitch, the task is simplified [Hida88]. The sense amplifiers and parallel

port circuitry is placed on either side of the cell array, as shown in the block diagram of the local embedded memory architecture, Figure 4.1.2.1. The bitlines for each column feed through the sense amplifiers, column access, and parallel port circuitry to the main databuses. With the additional six signals running through the parallel port circuitry, the layout will be widened to accommodate these buses and the additional bitline access, comparator, write, and precharge devices within the pitch of four, instead of two bitlines.

4.1.2.3 Shift Register Row Decoder

The row decoder/shift register is designed to provide wordline access either using normal addressing via addresses or using a cyclic shift register arrangement. The shift register is designed for use in conjunction with the parallel compare operation to reduce and simplify the algorithms that are applicable in a parallel column test mode. As illustrated in the circuit schematic of Figure 4.1.2.3.1, the NAND decoder structure remains unchanged.

The wordline driver is modified to include a master/slave D-type latch and additional gating transistors to allow the input of the adjacent wordline signal, WLBI, RAS, and shift register and wordline driver control signals.

Consider first the normal or mission mode operation of this circuit. SH is set low and SHB* is set high. During inactive or precharge periods when no wordline is selected, XEN is low, XEN0 is low, and the NAND decoder is reset. SHISO* is set low to isolate the wordline driver inverter and SHRAS is high to hold the active low wordlines high. When an active row access cycle begins, the addresses are asserted at the NAND decoder inputs, SHISO*, XEN, and XEN0 are driven high, and SHRAS is driven low. With the signals set up in these levels, the low selected NAND output propagates to the output driver, and similarly, the unselected NAND decoder nodes remain in their precharged high states keeping the associated wordlines high. At the end of the active row cycle, SHISO*,

Figure 4.1.2.3.1 Shift Register Decoder Design

XEN, and XEN0 are driven low, SHRAS is driven high resetting the wordline to its inactive high state, and the NAND decoder is precharged. These waveforms, and the control circuitry that is used to generate the signals, are detailed in Appendix 1 and the simulation results of Appendix 4.

When the row decoder is to be operated as a shift register, the inputs are asserted in a different sequence while the row addresses are ignored. To initialise the shift register, XEN is set low and remains low. XEN0 is pulsed low and, with SHB* in active high mode to enable the feedback path in the D-type latch, all of the wordlines are held high. To shift in the adjacent wordline signal, WLBI, SHISO* is held low, SHRAS is held high, and SHB* is pulsed low while SH is pulsed high to sample WLBI. The high SH signal gates the adjacent wordline input, WLBI, into the master D-type latch, while the low SHB* inhibits any changes in the slave latch. When SH returns low and SHB* returns high the data is clocked into the slave latch and the shift operation is completed. To implement a row access from the currently active shift register element, the element in which the slave latch currently stores a logic '0', SHRAS is driven low and SHISO* driven high to allow the slave latch to drive the wordline driver stage. The wordline is reset simply by setting SHISO* low and SHRAS high.

Another feature is added to the shift register decoder to perform bulk initialisation of the memory which is normally required at the beginning of memory test algorithms. The BWRT signal is used to drive all wordlines low so that a bulk write of one data polarity may be done to an entire cell array. This requires that SHISO* be held in a low state to isolate the slave D-type latch of the shift register element and that SHRAS be set low to allow the inverter gate to be driven high by BWRT. The control logic and external circuitry that is necessary to realise a circular shift register and these modes of operation is detailed in Appendix 1. The simulation of Appendix 4 illustrates all of the functions of this shift register row decoder circuit and verifies the circuit concept.

The circuit contains six additional control signals to fulfil its function: XEN0, the complementary shift signals, SH and SHB*, SHISO*, BWRT, and SHRAS signals. The adjacent wordline input signal is connected directly from the physically adjacent wordline shift register decoder element.

4.1.3 Test Controller Architecture: Definition and Requirements

A brief overview of the Memory Test System Architecture has been presented. This architecture was seen to consist of a Memory Test Controller, a System Test Bus, and the embedded memory macros. After the presentation of the enhancements to the memory architecture, more detail on how these memory circuit enhancements - the test controller and test bus, and the required DRAM test and repair sequences - will act in unison to arrive at a memory test and repair flow algorithm will be presented.

4.1.3.1 Memory Test and Repair: Flow Algorithm

As a critical system design criterion, the transfer or incorporation of test and repair "intelligence" in the local embedded memories was mentioned. This approach has many advantages with respect to the MTC design.

With the test and repair control logic incorporated into the embedded memories, the MTC tests all of the embedded memories in parallel using simplified shift register clocking techniques, as opposed to direct memory addressing, and serial address generation for memory test algorithms. The embedded memory blocks respond to the test stimuli by using only the address and data bits needed locally and by reconfiguring themselves based upon local test response verification. Should this not be possible, the memories notify the MTC via error and status signals.

Figure 4.1.3.1.1

**Memory Test
and Repair
Flow Algorithm
(MTRF)**

If these functions were instead to be embodied in the MTC, its complexity would greatly increase. For example, the MTC would have to monitor all test results from each memory. It would have to be designed to generate addresses and data separately for each memory, nullifying the advantage of serial address and data capture. It would have to generate enabling signals to test each memory and also signals to control the reconfiguration of each memory. It would have to be designed to "know" which of the multiple embedded memories has redundant elements. In addition to all of these disjoint tasks, all of which would change from ASIC to ASIC because the embedded memory systems necessarily change, the test algorithm must still be embodied in the MTC. Test time will also increase due to the sequential test and repair flow required for each memory, and any opportunity to test all memories simultaneously in parallel is forgone.

In direct contrast to this MTC design requirement scenario, the proposed test architecture requires that the MTC be concerned only with the memory system parameters N , M , and R , and the generation of the test and repair algorithm flow and requisite test algorithms based on those three parameters. Only counters embodied in the MTC need to be modified according to these three memory system parameters, making it easy to port the MTC from one ASIC to another for reuse.

When the local memories indicate via a shared test bus that no errors have occurred, or that errors have occurred that cannot be repaired due to lack of, or the exhaustion of redundant elements, the MTC decides upon acceptance or rejection of the ASIC. The MTC does not need to know the exact status of each memory, but rather the status of the memories as an interconnected system during test. In addition, the MTC does not need to differentiate between DRAMs and SRAMs which may be included in the test system scheme. If one particular memory test in the test flow is DRAM specific, such as the parallel port access test, the SRAMs are modified to ignore this test mode. Such advantages as these make the test architecture and MTC concept modular.

The resulting simplification of the Memory Test and Repair Flow (MTRF) algorithm is shown in Figure 4.1.3.1.1. At the beginning of the MTRF, the MTC initialises all of the embedded memories and precharges the shared status and error lines which will be driven by the memories. The Parallel Port Access Test is invoked. This is a simplified version of the Walking Row memory test algorithm based upon the memory system parameter R, dictated by the embedded block with the largest number of rows, or wordlines. During this test mode evaluation of the memories, the MTC monitors the shared error bus line to determine whether an error has occurred in one of the memories. If an error is detected, the MTC will reject the ASIC if no memories include redundancy. If one or more of the embedded memories includes redundant elements, the MTC will receive the error information indicated by the memories, but will continue the memory test and repair flow algorithm.

The next step in the MTRF algorithm is to proceed to the serial march test and reconfiguration. The march test to be used has been discussed in a previous section on test algorithms (see section 3.3.2). The MTC initialises all embedded memories and ensures that any of the memories which contain redundant elements and reconfigured decoders are in their reset mode in which the normal decoders are used. The MTC generates the address, data, and enable and read/write stimuli based upon the memory system parameter N, the maximum address space, and M the largest data word width. The local memories capture the pertinent address and data bits and execute the test algorithm locally. After each read cycle, each memory performs its own response verification. The test continues to its end if no errors are found. If an error is indicated and a memory has no redundant elements, the memory indicates to the MTC that it should be rejected via the system test bus accept/reject status line. If redundancy is available, faulty columns are replaced by the available redundant columns. If all of the redundancy is used up and an error is detected, the memory indicates this to the MTC via the system test bus

After the completion of this test, the second serial test sequence is invoked. The second test pass ensures that any redundant elements that have been substituted for defective ones in the first pass of this algorithm will be verified as functional during this second pass. If faults are detected during the second serial test pass, the MTC is notified that it should be rejected. Otherwise, the memory system test result is a pass.

The recommendation of two different test algorithms for the serial test and repair phase of the MTRF is not optimal, but is a necessary tradeoff between test time, MTC complexity, test algorithm fault coverage and redundancy effectiveness. The primary reason for two test algorithms is to provide the required coverage of the DRAM fault model. The $O(n)$ march test covers faults 1-4 of the DRAM fault model, while the $O(n^{3/2})$ Walking Row Test covers faults 5-7 in particular in the DRAM fault model and faults 1-4. An evaluation of the serial test algorithm test lengths for a model embedded memory system is included in a later discussion of the test algorithms to be used by the MTC based upon the system parameters N , M , and R .

A problem encountered is that the parallel test algorithm (PMWROW) fault coverage is maximised when the reconfigured elements have been invoked in a later test, due to the unknown nature of the errors detected. If errors found are due to cell spot defects only, a subsequent test sequence will find that insufficient redundancy exists to repair the faulty memory and it will be rejected. On the other hand, if the errors were due to PSF or sense amplifier sensitivity faults in the memory, and if a simpler memory test is specified for the first and second serial test passes, it will not sensitise or detect these PSF faults. Therefore, a more rigorous test, such as a serialised Walking Row test, is suggested for the serialised test under these conditions. Moving the parallel test to the end of the test flow is not an option because in the parallel port access, due to its design and required simplicity, normal and redundant columns are accessed before and after any column addressing reconfiguration occurs. If a defective column is replaced by invoking the

configurable column decoder, it will still be accessed and demonstrate faulty behaviour during the parallel test access mode. The parallel test is an efficient means of rejecting devices which contain memories with no redundancy. It may not be argued that for the sake of simplicity of design of the MTC that a failure during parallel test is cause enough to reject a device regardless of the availability of redundant elements. However, because of the indeterminate nature of the error or fault - a stuck-at cell, a faulty decoder, or cell pattern sensitivity - this will eliminate the advantage of the parallel test mode for memory blocks which incorporate redundant elements.

Some means of modifying the MTRF to utilise the PMWROW test to full advantage for memories that include redundant elements requires further investigation. Another important factor is the ability to perform full speed tests on the embedded blocks. In its present form, the MTRF and test system architecture do not provide for full speed test of the DRAM blocks. The addition of another test step in the test flow - which will likely require local address generators interfaced to the System Test Bus - should be considered in applying this test architecture to a multiple embedded DRAM system. A subset of a full memory test which is performed at speed to exercise the decoders and other peripheral circuitry has been done in a previous testable DRAM design effort [Inou87].

4.1.3.2 System Test Bus Specification

To control the Memory Test and Repair Flow algorithm sequence, the Memory Test Controller must necessarily generate and monitor a number of signals. The signals are classified into 5 groups:

1. memory control inputs;
2. test mode indicator signals;

3. test stimulus signals;
4. initialisation signals; and,
5. test status signals.

The memory control inputs consist of two signals, an enable signal designated to be RAS since we are primarily concerned with DRAMs, and a read/write signal designated as R/W* to control the type of I/O operation being performed on the embedded memories. These signals are functionally equivalent to the mission mode signals of each memory, however, a local multiplexer is included in each memory to select the system test bus signals during the test and repair flow. The use of only two signals for memory control is predicated upon a standardised embedded memory mission mode control interface, which, if extended to SRAMs, will allow their inclusion in the embedded memory test system. This makes the overall test architecture very flexible and useful for the ASIC design environment.

The test mode indicator signals, TST1, TST2, and LD/CMP*, are three signals which are used to indicate to the embedded memories whether they are in test or mission mode, and if in test mode, in what state in the test and repair flow sequence. The truth table shown in Table 4.1.3.2.1 illustrates the various modes and the logic values of these signals. With only four actual states in the memory test and repair flow algorithm, the use of the additional three states of the eight possible states eliminates the need for certain initialisation signals and test control signals by using the LD/CMP* bus. It is used in parallel test access mode to initialise all memories when LD/CMP* is '0' and to proceed with the test when set to '1'. The initialisation of the memories includes a bulk write operation to all cells and an internal reset of all configurable column decoder latch cells. Similarly, in the serial test modes, LD/CMP* is used to control the loading and comparing of address and data in the registers of each of the embedded memories.

Test Status Signals			Memory Test System Functions
TST1	TST2	LD/CMP	
L	L	L	Initialise memories for Memory Test and Repair Flow
L	L	H	Execute Parallel Column Test
L	H	H	Serialised Test (1), address and data load and memory access mode
L	H	L	Serialised Test (1) output response compare mode
H	L	H	Serialised Test (2), address and data load and memory access
H	L	L	Serialised Test (2) output response compare mode
H	H	X	Mission Mode default settings

Table 4.1.3.2.1 Test Status Signal Mode Description

The test stimulus signals consist of two buses, SADI/PD, Serial Address and Data Input/Parallel Data, and SHC, a Shift Clock. During parallel test mode, SADI/PD is used to provide the memories with input data and expected response data according to the state of the R/W* signal. Because all columns in all memories are tested in parallel with the same data sequence, only one data signal is required. The memories use the test mode indicator signals, control input signals, and data signals to perform the parallel writing and comparison required in this test. Circuit implementations to control the various signals needed locally by each memory are included in Appendix 1. During serial test modes, SADI/PD buses the serial address and data information to the embedded memories. The shift clock signal, SHC, is used by the MTC in parallel mode to execute the Walking Row algorithm by driving the shift register row decoders. In serial test mode, it is used as strobe that is activated when an address or data bit is asserted on the SADI/PD bus line. The local memories generate from SHC the necessary local shift register control signals to capture the current address or data bit from the serial bit stream.

The initialisation signal, RESET, also performs a dual function for the various test modes. In parallel test mode, RESET is used in the wrap-around operation of the shift register decoder and its generation in the parallel access test algorithm depends upon the memory system parameter R. With all memories tested simultaneously during parallel column access testing, a series of shift pulses is required to cycle through the largest row addressing space embedded block. Other smaller blocks disable themselves when their shift register decoders overflow. The RESET bus resets all overflow conditions in the shift registers and sets them up for the next cycle through the rows of the memories. During the two serial test modes, the RESET signal is used by the MTC to indicate to all of the embedded memories that new address and data is going to be presented on the SADI/PD line and to reset the local capture circuits.

The last group of signals to make up the test bus is the error and status signals. The

two signals, $ERROR^*$ and $STATUS^*$, are monitored by the MTC and are actually driven and shared by the embedded memories. During either parallel or serial testing, after the local memories perform a read compare operation, the $ERROR^*$ line is driven low if an error is detected or remains in its precharged high state if no errors occur. The memories use a local error signal to drive the reconfiguration control logic during serial testing. The $ERROR^*$ bus is used only as a diagnostic indicator of error status in both parallel and serial test modes. The $STATUS^*$ signal is used as an accept/reject signal. Local memory logic determines for each memory during the test and repair algorithm whether the memory system should be accepted or rejected as a result of the tests. During parallel test mode, when an error is encountered in a memory, the MTC is notified via the $STATUS^*$ signal. During serial test mode, if a memory has an error that cannot be repaired, $STATUS^*$ is driven low and in this case the MTC indicates a rejection of the device is required.

4.1.3.3 Parallel Walking Row Algorithm

The MTC engine for the embedded memory system embodies the memory test algorithms for the memory system. This task has been simplified by using a shared system test bus and by specifying a set of memory system parameters, N , M , and R , which describe the physical dimensions of the embedded memories as a system. The memory test algorithms are constructed around these three embedded memory system parameters.

The Walking Row memory test algorithm has been previously described in section 3.3.2. A Parallel Modified Walking Row is designed to use the parallel test access memory port and System Test Bus. The critical parameter for this PMWROW is R , which is determined by the individual memory block having the largest number of rows. The basic operation of this algorithm in its parallel implementation is shown in Figure 4.1.3.3.1.

After initialising the memory to all '0' ('1'), select a current row using the shift register

Figure 4.1.3.3.1

Parallel Walking Row
Algorithm Flow Diagram

decoder. Write '1' ('0') to all cells on that row in parallel. Read verify all '0' ('1') from all other rows, then read verify '1' ('0') in current row. Rewrite '0' ('1') in the current row and move to the next row. Repeat this process F times. To obtain the necessary worst case PSF coverage for the column as well as the four cell neighbourhoods in the DRAM, this process is done first for the even columns of the DRAM, and then for the odd columns of the DRAM. This retains the static PSF fault coverage, discussed previously in section 3.3.2.4, of the normal Walking Row algorithm which performs this test for each column in a series manner. Additional parallel test monitor circuitry is necessary (driven by the test mode status signals, TST1, TST2, and LD/CMP*), but results in the retention of the full fault coverage of the algorithm.

The normal Walking Row algorithm, as demonstrated in section 3.3.2, requires a total of $2N^{3/2}+5N$ steps to complete, including memory initialisation, where "N" is the number of memory locations in, say, an $N \times 1$ memory.

The Parallel Modified Walking Row algorithm requires a total of $4R^2+12R+4$ steps, for an equivalent memory with R rows and a parallel test access port. To make a comparison of the advantage of applying this test algorithm through the parallel test access port, consider the following memory parameters. A multiple embedded memory system consisting of " p " $N \times 1$ memories organised as $\sqrt{N} \times \sqrt{N}$ ($R=\sqrt{N}$, in this case) is to be compared with an equivalent single memory organised as $\sqrt{pN} \times \sqrt{pN}$. The test length for the parallel test is given by substitution as $4N+12\sqrt{N}+4$ and for the normal Walking Row algorithm on an equivalent memory as $2(pN)^{3/2}+5pN$. The speed-up factor for this algorithm implementation is approximately $0.5p^{3/2}\sqrt{N}$.

This sample case illustrates that with the increased parallelism afforded by the embedded memory test system architecture, $O(n^{3/2})$ algorithms are sped up by a factor of on the order $p^{3/2}$, and the precise speed up applied to the system is dependent upon the largest address space, N , and the total number of bits of memory in the memory system. A

factor of $\sqrt{N}$ improvement in test length is achieved when this is applied to only a single memory. This factor of improvement in test algorithm length, transforming $O(n^{3/2})$ test algorithms into $O(n)$ algorithms, makes parallelism a very useful feature. Because of additions to the embedded memories which allow them to be tested in parallel simultaneously, the test algorithm application complexity and test time is also further reduced in proportion to the added memory test system parallelism.

4.1.3.4 Serialised Test Algorithms

The MTRF for the multiple embedded memory system shows that a memory march test algorithm, shown in Figure 3.3.2.1.1, and the Walking Row algorithm must be implemented for the serialised test sequences during which all memories are tested in parallel, and in the case where a memory block has redundant cells available, it reconfigures based upon the test results monitored locally by each memory. The MTC generates the address, data, and control stimuli as if there is only one single memory with the dimensions $N \times M$, defined by the embedded memory system parameters.

With the system test bus architecture, the address and data stimuli are transmitted serially to the embedded memory blocks using the SADI/PD signal bus. The RESET bus is used to initialise the local memory address and data capture circuitry, the SHC clock is used to clock the serial bit stream into the local memory address and data registers, and the TST1 and TST2, and LD/CMP* signals are used to indicate which serial test mode is current. LD/CMP* initiates the serial bit stream load sequence, and the memory response compare operation, respectively. The MTC generates the addresses and data as per the test algorithms and serialises them into packets of bits as shown below. The memory control signals are asserted following the serial bit capture sequence:

$$\langle AB_{a-1}, AB_{a-2}, \dots, AB_1, AB_0, D_{b-1}, D_{b-2}, \dots, D_1, D_0 \rangle$$

The Address Bits, AB_i , are serialised first with the MSB first followed by the lesser significant bits. The Data bits, Di , are serialised following the address bits, again from MSB to LSB. The relationships of the serial data packets sent to the memories using SADI/PD are $a=\log_2 N$ and $b=M$. Each addressing and memory cycle during the serial test algorithm execution sequences requires $a+b=\log_2 N+M+1$ bit capture cycles plus the one additional cycle to initialise the local address and data capture circuitry. Following these cycles, the memory control signals, RAS and R/W^* , are activated and at the end of this cycle the response verification is initiated by the LD/CMP^* signal. Sample timing waveforms for this are included in Appendix 1.

It is evident that the serialisation of the memory address and data stimuli for test algorithms results in an extended test time. However, this is balanced by the fact that all of the memories are tested in parallel during this serial test sequence. A conservative measure of the added test length imposed by the test and repair bus architecture makes the following assumptions: each serial bit capture cycle requires the equivalent cycle time of a memory read or write operation and the RESET cycle requires the equivalent of a memory access cycle time. The fault monitor DRAM cycle times are approximately 50ns and depend upon the size of the embedded block of memory, so that it is evident that with bus driver circuitry, the assumptions made are within worst case upper bound conditions. $O(n)$ algorithms such as the march test of Figure 3.3.2.1.1 instead become $O(N(\log_2 N+M+1))$, or approximately $O(N\log_2 N+NM)$, where N and M are the embedded memory system physical parameters. However, this does not accurately reflect the true test length increase because the memories are tested simultaneously. To make a realistic comparison, consider "p" $N \times M$ memories tested using an $O(n)$ test algorithm and also using M -bit words as data stimuli. The test length comparison between applying the test algorithm sequentially to all

"p" memories and applying it using the test architecture for the "p" NxM memories in an equivalent embedded memory system is $\lceil p/(\log_2 N + M) \rceil$. As the number of memories in parallel increases and the density of the memories decreases, an application environment that is typical for ASICs, the ratio approaches unity. With the large word widths (M), however, this ratio is not expected to ever exceed unity. However, the penalty of the serial accessing technique is not severe and is more than offset by the reduced number of access lines required to apply the tests to the memory.

Using $O(n^{3/2})$ algorithms, such as Walking Row results in a test length increase of approximately $\lceil p/(\log_2 N + M) \rceil^{3/2}$.

4.1.4 Redundancy Implementation Considerations

The design of a configurable column decoder has been described and verified through simulation. Realisations of control logic blocks to interface an embedded memory to a test bus that have been proposed and the use of local test response verification to drive the memory reconfiguration has been shown as part of a larger Memory Test and Repair Flow algorithm. In the ensuing discussion, methods of incorporating redundant elements into an embedded block will be discussed. The efficiency of this implementation, comparison to row redundancy and a discussion of why it was not proposed for the embedded PAM reconfigurability scheme, and a means of implementing reconfigurability for very wide word RAMs which have no column decoders will be discussed.

Figure 4.1.4.1 shows how the redundancy is incorporated into the Y-decoder and cell array in a bit-block oriented cell array architecture. For each of the identical Y-decoders, one additional redundant column access driver, controlled by EN/DIS*, is added to the 1-of-i Y-decoder. A column of cells - two bitlines and a sense amplifier in the folded bitline architecture - is added to the cell array for each bit-block Y-decoder to provide the redundant memory elements. Regardless of the word width, and the number of bit-blocks, each decoder is in parallel to all other Y-decoders. When an error is detected during testing and reconfiguration is required, the EN/DIS* line which is shared by all of the Y-decoders is activated to reconfigure them. During a subsequent memory access to a faulty address which has been reallocated by the reconfiguration mechanism, all of the redundant columns in all of the Y-decoders are now used to replace the faulty column or columns. More complex reconfiguration schemes with multiple EN/DIS* control lines devoted to each bit-block Y-decoder may be used, however, the area overhead with respect to the embedded memory block as a whole would greatly increase at the expense of this use of the redundancy circuitry. The above simpler approach proposed, indicated in Figure 4.1.4.1,

Figure 4.1.4.1 Column Redundancy Architectural Overview

is a trade-off between ease of implementation of the reconfiguration control logic, and the flexibility for various sizes of embedded memories typical of modular RAM macro generation in an ASIC design environment.

Consider, for example, a 2Kx8 DRAM that is organised as 128 rows by 16 columns by 8 bits. If the redundancy scheme is implemented for this RAM, adding a seventeenth column to the sixteen column bit-blocks, each of which is 128 rows wide, the overhead in redundant cells alone is approximately 5.4%. This is not an unreasonable figure when considered alone. However, consider instead this 16K block of memory organised as a 512x32 memory. With the same approach only using 128 rows by 4 columns by 32 bits, there are 32 bit-blocks each with an added 128 cells. The cell overhead alone is approximately 40%. This is not acceptable, however, the need for redundancy may be just as critical in one memory configuration as in the other. There is obviously a need for an alternative means to implementing redundancy in wider word RAMs. It is necessary to note that the precise number of additional redundant cell elements in a particular embedded memory will be dictated by its density and word width organisation. In addition, it is not expected that every embedded block in an ASIC will require a redundancy capability, although with the proposed reconfiguration approach, this is entirely feasible.

Row redundancy is extremely advantageous from primarily an incurred overhead perspective. To continue the previous example, for a memory organised as 128 rows by 4 columns by 32 bits which is given one extra redundant row, the overhead in memory cells is less than 1% of the total number of memory cells. Similar comparisons may be made for many different embedded memory organisations.

However, implementing row redundancy with a circuit scheme as proposed, with EN/DIS* lines and special control circuitry, will not be optimal. The row access operation of the memory cycle is typically the longest part of memory access time. It is the critical path for speed performance. The present soft programmable redundancy scheme could be

adapted for row redundancy, but at the expense of increased access time due to the EN/DIS* discharge scheme for redundant element activation. Repairable memories would increase physical yield, but the resulting performance of the ASIC, designed for a certain memory speed or access time, will in most cases not meet specification, effectively reducing yield. In addition, with the shift register row decoder circuitry already added to the row decoder, the added reconfigurability circuitry would complicate the control of the decoder and its physical layout, which is already very cell pitch sensitive. For these reasons, the reconfigurable column decoder was found to be a reasonable compromise for implementing a simple and flexible redundancy scheme for embedded RAM.

Wide word and relatively small address space (N) memories, or those which use no column decoding at all, are prevalent in the embedded DRAM ASIC applications environment. Many of these RAMs are at density levels such that they may require the use of redundant elements. The problem that must be solved is how to do so efficiently within the context of the proposed memory and system architecture for test access and reconfigurability. Figure 4.1.4.2 illustrates an approach that is slightly more complex than the bit-block Y-decoder redundancy realisations, but provides an elegant solution to reduce redundant cell population for wide word RAMs. The solution is to add a single redundant column and to treat the memory as if it had a Y-decoder during test only. The columns of a DRAM with no column decoding will be equal to the number of data I/O bits, M. Figure 4.1.4.2 shows a 1-of-M Y-decoder distributed for the the normal column access device drivers and a separate and logically identical decoder for the data I/O buffers. Both decoders are inactive during mission mode operation but they are addressed using the serial address capture circuitry and system test bus. The redundant column is reconfigured to a given bit through the use of the distributed Y-decoders, identical to those discussed in section 4.1.2.1, which disable the defective normal column and enable a data bus switch to which the redundant column data are connected, as shown in Figure 4.1.4.3. This

Figure 4.1.4.2

Reconfiguration Concept in the Absence of Mission Mode Y-decoders

Figure 4.1.4.3

Datapath Modification in Absence of Mission Mode Y-decoders

circuitry is replicated for each column, or data bit, so that each may be replaced by the redundant column. The column access driver is a NOR gate, not an inverter, to accommodate the Y-decoder that is used only during test. In this configuration, the memory is a x1 device during test with M columns of cells. The circuitry suggested for implementing the test bus interface and control for each memory, detailed in Appendix 1, is directly applicable to this test architecture, with Y-decoders for testing only, for wide word RAMs with no mission mode column decode circuitry.

4.2 Related Issues

After having modified the embedded memory macro to include features allowing reconfiguration or redundancy, shift register wordline activation, and parallel bitline access during testing, there is an opportunity to utilise some of these added features for mission mode tasks or even to extend the memory architecture to allow for different testing techniques than proposed. In the case of the circuitry and architecture that have been proposed, memory self refresh, and parallel cell array response comparison testing may be added. These functions will be briefly discussed.

4.2.1 Memory Self Refresh

Refresh is required for dynamic RAMs. When embedded in an ASIC, the circuitry that generates the required address sequences for refreshing an embedded DRAM is not generally a part of the DRAM macro. Moreover, address counters which reside outside the memory macro require additional silicon area, although they are usually not separate from the counters or address generators used to access the memory during normal active cycles. With a shift register row decoder implementation, a simple method of achieving self refresh

Figure 4.2.1.1 Self Refresh Timing Example

without activating any memory addresses or incurring the power dissipation of row decoder NAND or NOR stages is possible. The accompanying timing diagram of Figure 4.2.1.1 illustrates the principle of self refresh. After a normal memory access cycle, an internally generated signal, RFSH*, is activated after some set delay period which is on the order of the duration of a complete memory cycle or a multiple thereof. This signal is used in conjunction with a system clock to generate the necessary internal signals, such as SH, SHB*, SHISO*, SHRAS, to operate the shift register and generate the required control signal sequence for a row access operation (see section 3.1 and 4.1.2.3). Depending upon the length of time the memory is held inactive, only a fraction of the memory may be refreshed. Depending upon the application environment, this may be accommodated through the use of a lockout circuit which prevents any normal memory access until refresh is completed. The added memory interface complexity, effectively making the memory a true pseudo-static RAM, and the loss of use during refresh cycles may be undesirable in certain ASICs, but the capability of self refresh releases the memory macro control circuitry from any requirement to monitor and control memory refresh.

4.2.2 Parallel Array Comparison Testing

The test and repair approach that has been proposed demonstrates the complexity, not only to memory design but also to algorithmic and test access technique, that is inherent in developing an architecture to accommodate this functionality. One significant area of improvement that would potentially restrict the row, column, and array configuration of embedded memories, but would also offer benefits to test time reduction and test data application flexibility is the use of parallel cell array response comparison.

Given the present test access approach, data, especially in the serial test modes, is

captured locally by each memory from an M-bit long serial data stream. Given the nature of the test algorithms applied, several, if not all, of these data bits can be identical for a given memory access. Two options for improving this situation exist. Firstly, if the data bits are repeated '1's or '0's for a single data word, a single data bit only may be sent and locally repeated to fill out the memory word size, which eliminates any possible data word application flexibility. Or, the data may be generated locally from, for example, a pseudo-random data generator and applied to the memory. By splitting the memory into two identical cell arrays and inserting circuitry which will write to each memory array simultaneously and perform a subsequent comparison of the output response from each cell array for a read operation [Haye85], the necessity of busing both input stimuli and expected output response data to a memory is eliminated, simplifying, in the case of the proposed memory system test architecture, the Memory Test Controller. This concept has been previously illustrated in Figure 2.3.1. With appropriate modifications to the Y-decoder, the parallel comparator circuit could, for example, compare the output responses of all of the bitlines in both arrays simultaneously, and utilise the same circuitry and principles of operation of the parallel test access port, including the CAM-style dynamic comparator which would then be driven by the two bitline pairs as opposed to output data and expected data. The shift register row decoder would still be exploitable with this test scheme. In the case of the architecture of the embedded memory proposed in section 4.1.2, simultaneous activation of duplicate cell arrays is implied since each cell array has its own dedicated sense amplifiers and parallel test access ports. Parallel array comparison, however, does not fit in easily with the present memory test and repair algorithm and access technique, but could conceivably be used to replace the proposed parallel test access technique, or become a simpler alternative for ASIC designs which do not require as elaborate and highly functional memory test and repair facility as is provided by the proposed system.

5. Conclusions

5.0 Summary of Research and Results

Many of the critical issues related to both memory design and memory and system test strategy have been addressed. Beginning with a memory test chip and design in advanced double level metal CMOS technology, a full analysis of the existing transistor level DRAM circuit was conducted to understand the internal operation of the memory macro. Based upon this circuit analysis and an inductive fault analysis of the DRAM cell and datapath, a fault model for this design was developed. The fault model developed contained fault species that heretofore had not been included in traditional memory fault models. Cell stuck-at and coupling faults are common to most memory fault models that have been proposed in previous literature [Abad83], however, the bitline pair loss of signal and bitline precharge degradation effect were two particular faults that were proposed for the DRAM based upon the circuit and datapath analysis. In addition, the folded bitline architecture of the DRAM led to a new cell neighbourhood configuration which differs significantly from the classical Moore and von Neumann neighbourhoods. Indeed, the concept of a column neighbourhood of cells in DRAMs was introduced to focus precisely on the critical DRAM operation of bitline charge differential sensing, which is the basis of the bitline pair loss of signal and bitline precharge degradation fault models. In semiconductor DRAMs which use charge storage techniques, data retention is a design and process characteristic which may or may not be considered a fault. It was not addressed in great detail because of the applications environment for which the DRAM is intended makes data retention a secondary issue.

Many memory test algorithms exist for static as well as dynamic RAM. March tests, galloping tests, and other test classes have been proposed in the past. Rather than design

new algorithms from ground zero, the fault model provided the information necessary to choose a test which would detect and cover all of the faults expected to occur in a DRAM design such as the one that was analysed in detail. In particular, two test algorithms, a march test [Nico85] and a more complex Walking Rows test were chosen to cover the DRAM fault model.

Ideally, it is preferable to use one algorithm so that a BIST circuit need be designed only for that single algorithm. However, after having developed the fault model and chosen the memory test algorithms, the DRAM test chips were tested using an industry standard memory tester and an Electron beam prober. This reinforced the need to retain a test, such as the Walking Rows test that was proposed (however, the memory tester used performs lengthier tests known as GALROW and GALPAT), to qualify the DRAM column neighbourhood. This test also covers static pattern sensitivity faults in the four cell DRAM neighbourhoods. The DRAM fault monitor test chip was found to be defective in design and no conclusive results that verify the proposed fault model were obtained. In particular, the need to test for neighbourhood pattern sensitivity in DRAMs, always proposed in the previous literature and theoretically conceivable for this DRAM design, was not physically established.

The practical fault modelling, test algorithm specification, and testing of the DRAM fault monitor devices were the foundations for the development of a modified memory architecture and system level memory interconnection architecture for embedded memories in ASICs. Also factored into this equation was the requirement for a memory redundancy capability that facilitates "soft" reconfiguration of embedded memory blocks that are found to have defective cells during test. Hard redundancy techniques normally applied to commodity memory are not practicable in the multiple embedded memory environment typical of telecommunications ASICs [Few89]. For an ASIC test environment, the test strategy and much of the physical circuitry that orchestrates the memory system test is

required to be as portable as possible from one ASIC to another. In addition, since ASICs which incorporate embedded DRAM usually have multiple blocks of varying configurations, the memory test system architecture required that the "intelligence" to perform memory reconfiguration be incorporated in each embedded block, and that parallelism be used to test and reconfigure all embedded blocks simultaneously.

The resulting test system, memory architectures, and test and repair algorithms meet the system needs for test simplicity, parallelism, and portability, while at the memory level, coverage of the full, proposed DRAM fault model and an optional memory redundancy capability are obtained. The memory test and reconfiguration system architecture consists of a Memory Test Controller, a System Test Bus to connect all of the embedded blocks during test, and the modified DRAM macro block incorporating the circuitry necessary to interface to the System Test Bus and to perform the necessary test and reconfiguration functions.

The Memory Test Controller embodies the memory system test and repair flow algorithm. Because of the System Test Bus architecture, the entire memory system appears as a single memory to the MTC, characterised by the memory system parameters of largest memory address space, N , largest memory word width, M , and largest number of rows, R . From ASIC to ASIC, only N , M , and R change so that minimal circuit changes need to be made to the MTC circuit. The MTC may be embodied in an ASIC or, by multiplexing I/Os to accommodate the test bus, the MTC may be an external ASIC which is used at the board level to test and reconfigure several ASICs.

The System Test Bus is the vehicle through which the MTC communicates with the embedded memory blocks to provide control, address, data, and test mode and status information bits. A 10 bit test bus is required to incorporate these signals. Serial address and data is used to reduce the number of signals necessary and to allow parallel connection of all of the memories to the System Test Bus. Many of the signals in the test bus have

dual functionality depending upon the test mode or state in the memory test and repair flow algorithm.

The embedded DRAM architecture has three new additions. A parallel test access port and shift register row decoder, which access all bitlines in parallel and greatly simplify the Walking Row test addressing technique, are added. A reconfigurable column decoder, incorporating an SRAM cell, is used to provide the soft redundancy capability. All of these circuits have been proven in simulation. Additional circuitry to control them and to interface them to the System Test Bus has been developed.

The Memory Test and Repair Flow algorithm is unique. The System Test Bus and its interface to the embedded memories allows all of the embedded blocks to be tested in parallel. Initially, the Parallel Modified Walking Row algorithm is used to detect the DRAM specific faults such as neighbourhood pattern sensitivity, bitline precharge degradation, and bitline pair loss of signal. Following that, the march test is run twice with the addresses and data captured locally in serial format from the MTC via the System Test Bus. During the first run of the test, the memories execute the march test and the detection of errors is ignored by the MTC, but used by the memories locally to activate the reconfiguration of the column decoders and the instantiation of the redundant cell elements. The second test pass verifies the functionality of the newly added elements. Embedded blocks which have no redundancy capability are tested in the same way and their response is monitored, as it is for all other embedded blocks, during the second test pass. This MTRF does not change from PLASIC to ASIC and, because of the system level and local memory architectures, it is applicable to memory systems containing memories with or without redundant elements. In addition, SRAM blocks may be added to this test system with the addition of interface logic to mask out their operation during the PMWROW test cycle. Due to the parallel nature of the test application, and the parallel access port for the PMWROW algorithm, test time speed-up is $O(p^{3/2}\sqrt{N})$ which improves as the number of

parallel memories increases. The serialised address and data access techniques used by the MTC for the march test stimulus results in a slow down of $O(p/(\log_2 N + M))$ which is reduced as the number of parallel memories increases.

There are several advantages to the memory and memory test system architecture that is proposed. The system level and local memory test application parallelism and the embodiment of the test and reconfiguration "intelligence" in the local memory blocks ease the test controller design requirements and allows it to be reused. The memory circuitry added, excluding the additional control and interface logic, will be fitted into the row decoder and bitline layout pitches making it amenable to module generation techniques. Local redundancy activation is done without the need for fault location algorithms. Both SRAM and DRAM blocks, and embedded blocks with or without redundant elements may be used in the embedded memory test system. The test controller can be incorporated on an ASIC chip or can be external. The memory test architecture and reconfiguration scheme is applicable to stand alone memory chips to simplify complex test algorithms, such as Walking Row, and to replace the need for laser programmable fusible links in redundancy circuits.

The drawbacks of the proposed system test architecture include the inability to perform full speed tests, a possibly large test circuit and redundant element overhead, and the high complexity of the test controller. In addition, if power is removed, the entire memory test and reconfiguration algorithm must be repeated when the power is restored. This is not expected to be a great drawback since in the telecommunications applications environment, a full system diagnostic is normally done at power up. Another drawback is that IC tester capability may not be compatible with the System Test Bus and Memory Test and Repair Flow design. The need in some cases for acting on test status information indicated on the test bus is not possible for ASIC testers which normally report test status only.

5.1 Recommendations for Future Work

Much of the analysis and design work has been completed for the embedded DRAM. The physical validation of the DRAM fault model on a new fault monitor device is necessary to justify the need for the complex Walking Row memory test algorithm and its parallel modified version used in the system test architecture. Another possibility to investigate is the use of pseudo-random test techniques and parallel cell array testing as opposed to the use of the test algorithm implementations proposed. The potential exists to simplify the memory system test architecture and test flow algorithm using these techniques.

The ideal test vehicle for implementing the memory system test architecture is a production ASIC with multiple embedded blocks of memory. However, the soft reconfiguration scheme, demonstrated in simulation, should be incorporated first into a prototypical test chip and optimised in design.

Though the capability for using laser fuse programmable redundancy in the memory exists, a candidate for changing the soft programmable technique to a more permanent method of redundancy implementation is the use of EEPROM cells. The use of laser programmable fusible links in embedded memory redundancy schemes, requiring a means of fault detection, location, and location downloading off chip - the need for which was eliminated in the proposed memory architecture by embodying the redundancy activation logic in the local memories - is not a very likely short term alternative that will simplify this task. The use of electrically blown fuses may lend itself to the proposed test and reconfiguration approach.

A full speed test provision should be added to the Memory Test and Repair Flow to enhance the existing advantages of the memory and memory system test architecture.

References

- [Agar88] Agarwal, V.K., B. Nadeau-Dostie, A.L. Silburt. "A Serial Interfacing Technique for Built-In and External Testing of Embedded Memories", BNR Internal Technical Report. August, 1988.
- [Abad83] Abadir, M.S. and H.K. Reghbati. "Functional Testing of Semiconductor Random Access Memories", Computing Surveys, Volume 15, No. 3. September, 1983, pp.175-198.
- [Been86] Beenker, F., et al. "Macro Testing: Unifying IC and Board Test", IEEE Design and Test. December, 1986, pp.26-32.
- [Benn84] Bennetts, R.G. Design of Testable Logic Circuits. Addison Wesley, Toronto, 1984.
- [Butz88] Butzerin, T., et al. "ASIC Testing with High Fault Coverage", VLSI Systems Design. September, 1988, pp.51-57.
- [Cnat78] Chatterjee, P. et al. "The Hi-C RAM Cell Concept", IEEE Transactions on Electron Devices, Volume ED-25, No.1. January, 1978, pp.33-41.
- [Chat79] Chatterjee, P.K. et al. "Leakage Studies in High Density Dynamic MOS Memory Devices", IEEE Transactions on Electron Devices, Volume ED-26, No.4. April, 1979, pp.364-575.

[Dekk87] Dekker, R. "Fault Modelling and Self-Test of Static Random Access Memories". Technical Report No. 1-68340-28(1987)25, Delft University of Technology. September, 1987.

[Died87] Diedrich, P.M.J. "A Fast 64K SRAM Family: Architectural Considerations", Carleton University Technical Report. December, 1988.

[Died89] Diedrich, P.M.J. "Y57 Fault Monitor: Analysis, Fault Modelling, and Testing Strategy", BNR Internal Technical Report. March, 1989.

[Few89] Conversations with Dr.I.Few, Northern Telecom Electronics.

[Fuji88] Fujiwara, H. et al. "Test Research in Japan", IEEE Design and Test of Computers. October, 1988, pp.60-79.

[Haye75] Hayes, J.P. "Detection of Pattern Sensitive Faults in Random Access Memories", IEEE Transactions on Computers, Volume C-24, No. 2. February, 1975, pp.150-157.

[Haye85] Hayes, J.P. and Y. You. "A Self-Testing Dynamic RAM Chip", IEEE Journal of Solid State Circuits, Volume SC-20, No. 1. February, 1985, pp.428-435.

[Hida89] Hidaka, H. et al. "Twisted Bitline Architectures for Multi Megabit DRAMs", IEEE Journal of Solid State Circuits. Volume 24, No. 1. February, 1989, pp.21-27.

[Hoge86] Hogeboom, J.H. "Memory Test Algorithms and Generation", BNR Technical

Report. March, 1986.

[Inou87] Inoue, J. et al. "Parallel Testing Technology for VLSI Memories", Proceedings of the 1987 IEEE International Test Conference. Paper 45.1, pp.1066-1071.

[Jarw88] Jarwala, N.T. and Dhiraj K. Pradhan. "TRAM: A Design Methodology for High Performance, Easily Testable Multi-Megabit RAMs", IEEE Transactions on Computers. Volume 37, No. 10. October, 1988, pp.1235-1250.

[Kino86] Kinoshita, K. and K. Saluja. "Built-In Testing of Memory Using an On-Chip Compact Testing Scheme", IEEE Transactions on Computers, Volume C-35, No. 10. October, 1986, pp.862-870.

[Lu89] Lu, N.C.C. "Advanced Cell Structures for Dynamic RAMs", IEEE Circuits and Devices Magazine. January, 1989, pp.27-36.

[Mari82] Marinescu, M. "Simple and Efficient Algorithms for Functional RAM Testing", Proceedings of the IEEE International Test Conference. November, 1982, pp.236-239.

[Mazu87] Mazumder, P. "Testing and Fault Tolerance Aspects of High Density VLSI Memory". Technical Report No. UIUC-ENG-88-2206-CSG-81, University of Illinois at Urbana-Champaign, 1987.

[McAd86] McAdams, H. et al. "A 1Mbit CMOS Dynamic RAM with Design-For-Test Functions", IEEE Journal of Solid State Circuits. Volume SC-21, No. 5. October, 1986,

pp.635-642.

[Moor86] Moore, W. "A Review of Fault Tolerant Techniques for the Enhancement of Integrated Circuit Yield", Proceedings of the IEEE, Volume 74, No. 5. May, 1986, pp.684-698.

[Nair78] Nair, R., S.M. Thatte, and J. Abraham. "Efficient Algorithms for testing semiconductor random access memories", IEEE Transactions on Computers, Volume C-27, No. 3. June, 1978, pp.572-576.

[Nair79] Nair, R. "Comments on an Optimal Algorithm for testing Stuck-at Faults in Random Access Memories", IEEE Transactions on Computers, Volume C-28, No. 3. March, 1979, pp.258-261.

[Nico85] Nicolaidis, M. "An Efficient Built-In Self Test Scheme for Functional Test of Embedded RAMs", Proceedings of the IEEE International Test Conference. November, 1985, pp.118-123.

[Park88] Parker, K.P. "Testability: Barriers to Access", IEEE Design and Test. October, 1988, pp.11-15.

[Ride79] Rideout, V.L. "One Device Cells for Dynamic Random Access Memories: A Tutorial", IEEE Transactions on Electron Devices, Volume ED-26, No.6. June, 1979, pp.839-851.

[Sawa88] Sawada, K. et al. "A 72K CMOS Channelles Gate Array with Embedded

1Mbit Dynamic RAM", Proceedings of the 1988 IEEE Custom Integrated Circuits Conference, pp. 20.3.1-20.3.4.

[Schv86] Schvan, P. "Defectivity and Circuit Yield Part 2: Circuit Yield Analysis and Yield Projection", Northern Telecom Technical Report. September, 1986.

[Shen85] Shen, J.P., W.Maly, and F.J. Ferguson. "Inductive Fault Analysis of MOS Integrated Circuits", IEEE Design and Test, December, 1985, pp.13-26.

[Silb87] Silburt, A. "Y33 Narrow Port Memory Test Analysis", BNR Technical Report. October, 1987.

[Srin77] Srin, V.P. "API Tests for RAM Chips", IEEE Computer, Volume 10, No. 7, January, 1977. pp.32-35.

[Srin78] Srin, V.P. "Fault Location in a Semiconductor Random Access Memory Unit", IEEE Transactions on Computers, Volume C-27, No. 4, April, 1978, pp.349-358.

[Stap88] Stapper, C.H. "Clock Realignment: A method for Increasing the Yield of Memory Chips that are Partially Good". Paper 6.3. Proceedings of the IEEE Workshop on Fault Tolerance in VLSI Systems, October, 1988.

[Suk80] Suk, D.S. and S.M. Reddy. "Test Procedures for a Class of Pattern Sensitive Faults in Semiconductor Random Access Memories", IEEE Transactions on Computers, Volume C-29, No. 6, June, 1980, pp.419-429.

[Turi88] Turino, J. "Circuit Testability is Critical for Product Success", EDN, September, 1988, pp.219-232.

[MOS86] "An Analysis of the NEC μ PD41256 256Kx1 NMOS Dynamic RAM", MOSAID Inc. Design Analysis Services Technical Report. March, 1986.

"MS2200 Memory Tester User's Manual", Version 2.0. MOSAID Systems Inc. July, 1986.

Appendix 1

Test and Reconfiguration Circuit Overview

In the following four sections of this Appendix, an elaboration of the control circuitry that may be implemented to achieve the functional requirements for column decoder reconfiguration, parallel column access, shift register decoder operation, and serial address and data capture will be presented. Also included are sample timing diagrams and fundamental logical relationships for embedded memory control signals generated from the System Test Bus shared by all embedded memories, and from local internal signals that are required to sequence embedded memory access during mission mode operation.

A.1.1 Column Decoder Reconfiguration Scheme

The intended operation and options for implementing a configurable column decoder have been discussed in sections 4.1.2.1 and 4.1.4. Figure A.1.1.1 summarises the general principles. During testing when an error is detected by the local data response comparator in each embedded memory block, a local error signal, $ERROR_{loc}$, is used to initiate reconfiguration of the current column decoder address via the EN/DIS^* signal. If no reconfiguration is possible, either because no redundancy exists or because it has already been fully used, the reconfiguration control circuitry alerts the Memory Test Controller via the $STATUS^*$ bus. The circuitry used to achieve this control function is illustrated in Figures A.1.1.2 and A.1.1.3.

The EN/DIS^* control circuit functions as follows. During the first serial test pass, when a local error occurs the cross-coupled NOR gates generate an output error signal ERR . LD/CMP^* resets this latch during serial address and data capture, and enables it during the actual memory access phase of the serialised memory access. If the redundant circuits are not used up when an error is detected, that is, $REDUSE^*$ is in a logic high state, EN/DIS^* is driven low to program the current address out of the normal column decoder and to invoke the redundant column. If $REDUSE^*$ is low, indicating that the redundant circuitry is exhausted, EN/DIS^* cannot be driven low and remains tristated and precharged high, a function that is fulfilled by a precharge circuit that must be included in the embedded memory mission mode circuitry (See Appendix 2). In A.1.1.3, assuming that only one reconfiguration circuit exists, the signals R , RE^* , and RG^* set $REDUSE^*$ to its active low state when a redundant circuit is programmed. $TST1$, $TST2$, and LD/CMP^* reset this circuit during the System Test Bus memory test initialisation mode. If more than one redundant column decoder is implemented, these circuits must be repeated and their interface to the System Test Bus modified to accommodate the added redundancy.

Figure A.1.1.1 Reconfiguration Control Block Diagram

Figure A.1.1.2 EN/DIS Control Circuit Design

Figure A.1.1.3 Redundancy Usage Control Circuitry

Figure A.1.1.1.5 Sample Timing Diagram for Reconfiguration Scheme

capability.

To activate the STATUS* bus when the memory contains errors that it cannot repair through reconfiguration, a circuit such as that shown in Figure A.1.1.4 may be used. During read operations, when errors have occurred and REDUSE* is in its low state, a default setting for those embedded memories not possessing redundancy capability, STATUS* is driven low. The Memory Test Controller need only be concerned with the STATUS* signal during the second serial test pass since the first test pass is used only to allow reconfiguration to occur to repair detected errors. The second test pass verifies that the reconfigured circuitry and the rest of the memory is functional.

The timing diagram of Figure A.1.1.5 demonstrates the response of the various signals in the embedded memories and System Test Bus which must be used in the reconfiguration operation.

A.1.2 Parallel Test Access Port Scheme

The operation of the Parallel Test Access Port to the embedded memory is illustrated in Figure A.1.2.1. For a memory read cycle, or response compare cycle, RAS is asserted and the SADI/PD bus provides the expected data via the System Test Bus. The $PTST_{loc}$ signal is generated from test control signals and mission mode timing clocks in the embedded memories, as shown in Figures A.1.2.2 and A.1.2.3, to control the Parallel Test Access port. The expected data signals are compared to the output signals present on the bitlines, as described in section 4.1.2.2, and the resulting error signal, $ERROR_{ptst}$, is generated. This signal is also used to indicate to the Memory Test Controller, via the System Test Bus $ERROR^*$ bus as shown in Figure A.1.2.4, that an error has been detected. For write cycles via the parallel test access port, a similar sequence of events is invoked, however, as was discussed in section 3.1 concerning the embedded DRAM circuitry, a slightly different timing sequence is generated. Note that the $ERROR_{ptst}$ bus is activated on parallel write cycles, however, this has no effect on the $ERROR^*$ bus of the System Test Bus due to the circuitry suggested in Figure A.1.2.4.

The $PTST_{loc}$ signal is a timing signal that is used to synchronise the completion of the memory data sense operation with the beginning of parallel column response compare. The circuit shown in Figure A.1.2.2 is a modified version of the circuit described in section 3.1 which is used to synchronise the beginning of normal column decoding in the memory following the data sense operation.

The logical relationships for the various parallel test port control signals are shown in Figure A.1.2.3. The control signals, RAS and R/W^* , are System Test Bus generated but are passed through a test mode/mission mode control signal multiplexer that is explained in Figure A.1.3.5.

RAS, R/W, SADI/PD, SHC, ERROR are all System Test Bus Signals common to all embedded memories.

The remaining signals shown are generated locally by each memory.

Figure A.1.2.1 Parallel Access Compare and Write Sample Timing

Figure A.1.2.2 Local Parallel Test Signal Generators

Figure A.1.2.3 Parallel Test Access Signal Generation Logic

Figure A.1.2.4 Parallel Test System Test Bus ERROR Control

A.1.3 Shift Register Decoder Scheme

The Shift Register Decoder is designed to apply the Walking Row test algorithm with parallel access to all columns in an embedded memory block. In addition, it is also intended to allow all memories to be tested in parallel from the System Test Bus. An overview of the features is shown in Figure A.1.3.1, and the discussion of the Parallel Modified Walking Row Test algorithm in section 4.1.3.3 and the circuit itself in section 4.1.2.3 complement the following explanation of the operation of the shift register decoder.

At the beginning of the Memory Test and Repair Flow, the MTC sets its mode or status signals, TST1, TST2, and LD/CMF*, to an all low state. This is used locally by each memory to perform a bulk initialisation via BWRT when RAS is activated during this state. When the actual parallel testing sequence begins, a logic '0', the active state for a P-channel storage cell access, is shifted into the first shift register decoder stage, and the test algorithm is implemented simply by shifting from adjacent wordline to adjacent wordline. When the last wordline is reached for a particular embedded memory, an overflow flag, OVF, is set which inhibits any further operation. This allows the algorithm to be completed for the memory possessing the largest number of rows, R, an embedded memory system parameter. When the largest memory overflows its row decoder shift register, a reset pulse is issued on the System Test Bus RESET signal and the cycle continues again with the first row of all embedded memories.

The circuitry that can be used to fulfill the overflow function detector locally for each memory block is shown in Figures A.1.3.2 and A.1.3.3. The overflow flag is reset during the bulk initialisation operation. The use of a local shift signal, SH, to steer the wordline signal which writes the OVF latch prevents the OVF flag from being set before the read or write operation on the last row has been performed. OVF and its complement are then used to drive '1's only into the first shift register decoder stage following the overflow

Figure A.1.3.1 Shift Register Decoder Operational Overview

Figure A.1.3.2 Overflow Detector Logic

(a) Schematic

Figure A.1.3.3 Overflow Flag Propagation and Shift Register Shut Down

(b) Timing

Figure A.1.3.3 Overflow Flag Propagation and Shift Register Shut Down

Internal RAS
is derived from
System Test Bus
RAS signal only
for the parallel
test access mode.

SHC, TST1, TST2, and LD/CMP are System Test Bus signals

Figure A.1.3.4 Shift Register Control Signal Logic Relationships

Figure A.1.3.5 Control Signal Multiplexing for Test and Mission Modes

Figure A.1.3.6 Sample Timing Diagram for Shift Register Control Signals

condition, keeping all rows disabled until a RESET is received from the MTC. The timing diagram of Figure A.1.3.3 illustrates this principle. $WLB''R_1$ is the input to the first stage of the shift register decoder.

The logic relationships that are required for the various control signals used to operate the shift register decoder are summarised in Figure A.1.3.4. The signals SHRAS, SHISO*, XEN, and XEN0 all have dual functions during parallel test mode and during mission mode. XEN' in particular is functionally identical to RS' or LS' in Figure 3.1.7. With proper design, degradation of row decoder performance due to these additional logic relationships will be minimal.

Figure A.1.3.5 shows the use of multiplexers on the RAS and R/W* control inputs to memories. This done so that these signals may be generated by the System Test Bus independently of any control signal generation circuitry used locally for each embedded memory during mission mode. A typical timing sequence for shift register decoder signal operation is given in Figure A.1.3.6.

A.1.4 Serial Address and Data Capture Scheme

As explained in section 4.1.3.4, the address and data information is conveyed to the embedded memories during serialised test modes using the SADI/PD System Test Bus signal with the address bits and then the data bits in serial form, from most significant bit to least significant bit. Each embedded memory is to be equipped with the circuitry shown to in Figure A.1.4.1.

The Bit Mask Counter is a counter which determines the number of address bits in the serial address bit stream, the higher significant bits, that a particular embedded memory must ignore. After these bits have been sent by the MTC, the MEN, Mask Enable signal, allows local shift pulses which drive a serial address shift register to capture the remaining address bits. If any of the ignored address bits were logic '1', indicating that the address is out of the address space of the embedded block, the SEN, Shift Enable, signal remains low until the next RESET pulse, effectively disabling the memory. SEN is used in conjunction with the normal memory timing and control signals to lock out any access cycle for the memory under these conditions.

During the address or data capture mode of operation, the address and data buffers are set up as serial shift registers with elements as shown in Figure A.1.4.2 and A.1.4.3. The data capture control circuit is similar to that of Figure A.1.4.1 except that there would be no need for the Invalid Address Detector. The Bit Mask Counter is removed to allow extra data bits that are captured to be lost off the end of the data register.

Figure A.1.4.4 shows the error detection technique that is used. Using a dynamic NOR gate controlled by the System Test Bus status signals, when LD/CMP* is in its low state indicating a response verification operation, the expected data, DSi, and the output data, DOi, are compared. Each data bit has associated with it the exclusive OR gate and pulldown transistor shown, and all of the bits in the output data word are compared in

Notes: Not shown are the test bus inputs TST1, TST2, and LD/CMP which generate test mode status control signals which enable the serial capture circuitry for test and disable for mission mode operation.

SEN: Shift ENable

MEN: Mask ENable

SHA, SHAB: Shift clock for Addresses

Figure A.1.4.1 Serial Address and Data Capture Scheme

Figure A.1.4.2 Address Buffer Configuration

Figure A.1.1.4.3 Data I/O Buffer Configuration

Figure A.1.4.4 Parallel Data Comparator Circuit Concept

The reconfigurable column decoder circuit that has been simulated is shown in Figure A.2.1. The simulation circuit netlist SUBCCT file and the input stimulus LOGIC file are also included in this appendix.

The simulation results, in the form of waveforms shown in Figure A.2.2, show the following sequence of events. From 0 to 60ns, the inputs PHEN, TST, and ENDISB are activated to turn on the decoder and to pull ENDIS to a low state at 27ns. The normal column access signal, YI, is disabled 2ns later and the RYI redundant column access signal is driven high by ENDIS, completing the reconfiguration sequence. Following this programming sequence, PHEN and ENDISB are reset low and TST is reset high. Beginning at 63ns, PHEN rises and initiates a column access. YIB rises as usual, but the SRAM cell is accessed and ENDIS is discharged to drive RYI high, activating the redundant column at 67ns. The column access is completed when PHEN falls, resetting YIB and precharging ENDIS back to VCC.

Appendix 2

Configurable Column Decoder Simulation


```

FILE. STAPY    LOGIC    A1    BELL NORTHERN RESEARCH

PART E.    DECODER SIMULATION FOR RECONFIGURABILITY
$
$
$ CIRCUITS THAT CAN BE SIMULATED
$
$ MACRO, ON, DESIG
CM1DEC,U1,VDD,PHIEN,X3,V18,X5,X6,ENDIS, /
X8,X9,X10,YDIS,RESET,X13,V1,TST,X16,ENDISB,RV1,X19,X20,VSS
$ MACRO, OFF
$ INPUT STIMULI
$
VDD, VSOURCE, DC, 5.0
VSS, VSOURCE, DC, 0.0
$V1, VSOURCE, DC, 0.0
$V2, VSOURCE, DC, 0.0
$V19, VSOURCE, DC, 5.0
$V20, VSOURCE, DC, 0.0
$
$ PIECE WISE LINEAR INPUT WAVEFORMS
$
$ PHIEN, VSOURCE, P1", 0.0, 10e-09,0, /
12e-09,5, /
45e-09,5, /
47e-09,0, /
60e-09,0, /
63e-09,0, /
65e-09,5, /
85e-09,5, /
87e-09,0
$
$ TST, VSOURCE, PWL, 0.0, 10e-09,0, /
12e-09,5, /
52e-09,5, /
54e-09,0
$
$ ENDISB, VSOURCE, PWL, 0.0, 25e-09,0, /
27e-09,5, /
52e-09,5, /
54e-09,0
$
$ RESET, VSOURCE, PWL, 0.5, 5e-09,5, /
7e-09,0
$
$ $ THAT IS ALL

```

FILE: COLDEC SUBCCT A1 BELL-NORTHERN RESEARCH

(COLDEC CIRCUIT: THE RECONFIGURABLE COLUMN DECODER

THE CIRCUIT NODES ARE AS FOLLOWS:

'1 VCC
'2 VSS
'3 PHEN (DECODER ENABLE)
'4 Y1B (DECODER NORMAL HAND OUTPUT)
'7 ENDIS (ENABLE AND DISABLE PROGRAMMING AND ACTIVE SIGNAL)
'9 NO NAME OUTPUT OF LATCH TO TRANSFER GATE
'11 YDIS (PROGRAMMED SIDE OF LATCH WHICH GATES V_DRIVER
'12 RESET
'14 Y1 (NORMAL V ACCESS SIGNAL)
'15 TST (PROGRAMMING TEST INPUT FOR ENDIS)
'17 ENDISB (PROGRAMMING TEST INPUT FOR ENDISB)
'18 RY1 (REDUNDANT COLUMN ACCESS SIGNAL)

MISP1 1 3 2 1 L1.5 W8 16 16 PR1 16 16
MISP2 1 4 2 1 L1.5 W8 16 16 PR1 16 16
MISN1 21 3 21 5 L1.5 W4 8 8 PR1 8 8
MISN2 21 4 1 5 L1.5 W4 8 8 PR1 8 8
MISN3 21 5 1 6 L1.5 W4 8 8 PR1 8 8
MISN4 21 6 2 21 L1.5 W4 8 8 PR1 8 8
MISN5 21 7 10 8 L1.5 W8 16 16 PR1 16 16
MISN6 21 8 11 21 L1.5 W8 16 16 PR1 16 16
MISN7 21 7 10 9 L1.5 W4 8 8 PR1 8 8
MISP3 1 9 4 7 L1.5 W4 8 8 PR1 8 8
MISP4 1 10 4 1 L1.5 W4 8 8 PR1 8 8
MISN8 21 10 4 21 L1.5 W2 4 4 PR1 4 4
MISP5 1 9 11 1 L1.5 W6 12 12 PR1 12 12
MISP9 21 9 11 21 L1.5 W3 6 6 PR1 6 6
MISP6 1 11 9 1 L1.5 W6 12 12 PR1 12 12
MISN10 21 11 9 21 L1.5 W3 6 6 PR1 6 6
MISN11 21 11 12 21 L1.5 W6 12 12 PR1 12 12
MISP7 1 13 4 1 L1.5 W12 24 24 PR1 24 24
MISP8 1 14 11 13 L1.5 W12 24 24 PR1 24 24
MISN12 21 14 4 21 L1.5 W6 12 12 PR1 12 12
MISN13 21 14 11 21 L1.5 W6 12 12 PR1 12 12
MISP9 1 16 15 1 L1.5 W12 24 24 PR1 24 24
MISN14 21 16 15 21 L1.5 W6 12 12 PR1 12 12
MISP10 1 19 17 1 L1.5 W50 100 100 PR1 100 100
MISP11 1 17 16 19 L1.5 W50 100 100 PR1 100 100
MISN15 21 17 15 20 L1.5 W25 50 50 PR1 50 50
MISN16 21 20 17 21 L1.5 W25 50 50 PR1 50 50
MISP12 1 17 21 1 L1.5 W50 100 100 PR1 100 100
MISP14 1 17 1 1 L1.5 W6 12 12 PR1 12 12
MISP13 1 18 7 1 L1.5 W6 12 12 PR1 12 12
MISP17 21 17 21 21 L1.5 W120 240 240 PR1 240 240
MISP18 21 17 21 21 L1.5 W120 240 240 PR1 240 240
MISP19 21 18 7 21 L1.5 W3 6 6 PR1 6 6

FILE: CODES SUBJECT A: BELL-NORTHERN RESEARCH

PAGE 0002

C 3 2 10E-15
C 5 21 10E-15
C 6 21 10E-15
C 8 21 10E-15
C 13 21 10E-15
C 19 21 10E-15
C 20 21 10E-15
R 1 3 21 10000000
R 2 5 21 10000000
R 7 6 21 10000000
R 4 8 21 10000000
R 5 13 21 10000000
R 6 19 21 10000000
R 7 20 21 10000000

END

The reconfigurable column decoder circuit that has been simulated is shown in Figure A.2.1. The simulation circuit netlist SUBCCT file and the input stimulus LOGIC file are also included in this appendix.

The simulation results, in the form of waveforms shown in Figure A.2.2, show the following sequence of events. From 0 to 60ns, the inputs PHEN, TST, and ENDISB are activated to turn on the decoder and to pull ENDIS to a low state at 27ns. The normal column access signal, YI, is disabled 2ns later and the RYI redundant column access signal is driven high by ENDIS, completing the reconfiguration sequence. Following this programming sequence, PHEN and ENDISB are reset low and TST is reset high. Beginning at 63ns, PHEN rises and initiates a column access. YIB rises as usual, but the SRAM cell is accessed and ENDIS is discharged to drive RYI high, activating the redundant column at 67ns. The column access is completed when PHEN falls, resetting YIB and precharging ENDIS back to V_{CC} .

Figure A.2.2 Configurable Y-decoder Simulation Plot

Appendix 3

Parallel Test Access Port Simulation

Numbers refer to the transistor
designaution MISN, P in SUBCCT file.

Figure A.3.1 Parallel Test Port Simulation Schematic

PAGE 00001

FILE: COMPAR_3 LOGIC A1 BELL-NORTHERN RESEARCH

* TITLE. PARALLEL COMPARATOR CIRCUIT SIMULATION: ERROR DETECTION

* \$COMPAR_3 PRECHARGE NETWORK ON THE TRANSFERRED BITLINES

\$

* \$CIRCUITS THAT CAN BE SIMULATED

\$

*MACRO. ON. DESIG

*COMPAR_3. U1. VDD. VSS. BL1. BL2. PTST. BLT1. BLT2. ED. EDB. /

X10. X11. X12. X13. M1CH. PREB. ERROR. PRE. VSS

*MACRO. OFF

\$ INPUT STIMULI

\$

VDD. VSOURCE. DC. 5.0

VSS. VSOURCE. DC. 0.0

\$

*PIECE WISE LINEAR INPUT WAVEFORMS

\$

*PTST. VSOURCE. PWL. 0.0.

10e-09.0. /

12e-09.5. /

45e-09.5. /

47e-09.0

\$

*PREB. VSOURCE. PWL. 0.0.

5e-09.0. /

7e-09.5. /

52e-09.5. /

54e-09.0

\$

*PRE. VSOURCE. PWL. 0.5.

5e-09.5. /

7e-09.0. /

52e-09.0. /

54e-09.5

\$

*ED. VSOURCE. PWL. 0.0.

20e-09.0. /

22e-09.5. /

38e-09.5. /

40e-09.

\$

*EDB. VSOURCE. PWL. 0.0.

20e-09.0. /

22e-09.5. /

38e-09.0. /

40e-09.0

\$

* THAT IS ALL

3 of/de 3

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

PAGE 00001

FILE: COMPAR_1 LOGIC A1 BELL-NORTHERN RESEARCH

*TITLE. PARALLEL COMPARATOR SIMULATION: NO ERROR DETECTION

\$

\$

\$

\$CIRCUITS THAT CAN BE SIMULATED

\$

*MACRO. ON. DESIG

*COMPAR. U1, VDD,VSS,BL1,BL2,PTST,BLT1,BLT2,ED,EDB, /

X10,X11,X12,X13,MATCH,PREF,ERROR,PREF,VSS

*MACRO. OFF

\$

\$INPUT STIMULI

\$

VDD, VSOURCE, DC, 5.0

VSS, VSOURCE, DC, 0.0

\$

\$PECE WISE LINEAR INPUT WAVEFORMS

\$

\$

\$ PTST, VSOURCE, PWL, 0.0, 10e-09,0, /

12e-09,5, /

45e-09,5, /

47e-09,0

\$

\$ PREF, VSOURCE, PWL, 0.0, 5e-09,0, /

7e-09,5, /

52e-09,5, /

54e-09,0

\$

\$ PRF, VSOURCE, PWL, 0.5, 5e-09,5, /

7e-09,0, /

52e-09,0, /

54e-09,5

\$

\$ ED, VSOURCE, PWL, 0.0, 20e-09,0, /

22e-09,0, /

38e-09,0, /

40e-09,0

\$

\$ EDB, VSOURCE, PWL, 0.0, 20e-09,0, /

22e-09,5, /

38e-09,5, /

40e-09,0

\$

\$ THAT IS ALL

PAGE 00001

FILE COMPAP SUBCOT A1 DELL-NORTHERN RESEARCH

PARALLEL COMPARE CIRCUIT AND P ARALLEL WRITE CIRCUIT

NODES IN THE FOLLOWING CIRCUIT ARE:

```

'1 VCC
'2 VSS
'3 BL1 (BITLINE)
'4 BL2 (BITLINE)
'5 PTST (PARALLEL TEST INITIATOR CLOCK)
'6 BLT1 (TRANSFERRED BITLINE)
'7 BLT2 (TRANSFERRED BITLINE)
'8 ED (EXPECTED DATA)
'9 EDB (EXPECTED DATA BAR)
'14 MTCH (CAM COMPARTOR OUTPUT ACTIVE LOW FOR ERROR)
'15 PREC (MTCH PRECHARGE CLOCK)
'16 FRORP
'17 PRE (ERROR CLOCK LOW PRECHARGE: SHARED BY ALL COMPARATORS)
'18 PWT (PARALLEL WRITE CLOCK)
  THIS CLOCK ALSO PRECHARGES BLT1,2 VCC-VTN

```

```

MISP1 1 3 2 1 L5.0 W1.5 3 3 PR1 3 3
MISP2 1 7 8 1 L1.5 W3 6 6 PR1 6 6
' THESE DEVICES BELOW HAVE BEEN ELIMINATED
MISP3 1 6 9 1 L1.5 W3 6 6 PR1 6 6
MISP4 1 14 15 1 L1.5 W3 6 6 PR1 6 6
MISP5 1 16 14 1 L1.5 W15 30 30 PR1 30 30
MISP6 1 16 1 1 L1.5 W940 2000 2000 PR1 2000 2000
MISP8 1 1 16 1 L1.5 W189 400 400 PR1 400 400
MISN14 2 2 16 2 L5.0 W95 200 200 PR1 200 200
' THAT WAS THE SIMULATED LOAD DEVICES ON THE ERROR BUS
MISN1 2 4 1 2 L10 W1.5 3 3 PR1 3 3
MISN2 2 3 5 6 L1.5 W4 8 8 PR1 8 8
MISN3 2 4 5 7 L1.5 W4 8 8 PR1 8 8
MISN4 2 7 8 10 L1.5 W6 12 12 PR1 12 12
MISN5 2 10 18 2 L1.5 W6 12 12 PR1 12 12
MISN6 2 6 9 11 L1.5 W6 12 12 PR1 12 12
MISN7 2 11 18 2 L1.5 W5 12 12 PR1 12 12
MISN8 2 14 6 12 L1.5 W6 12 12 PR1 12 12
MISN9 2 14 7 13 L1.5 W6 12 12 PR1 12 12
MISN10 2 12 8 2 L1.5 W6 12 12 PR1 12 12
MISN11 2 13 9 2 L1.5 W6 12 12 PR1 12 12
MISN12 2 16 17 2 L1.5 W25 50 50 PR1 50 50
MISP7 1 14 16 1 L1.5 W2 4 4 PR1 4 4
MISN13 2 14 16 2 L5.0 W2 3 3 PR1 3 3

```

```

C6 6 2 10E-15
C7 7 2 10E-15
L10 10 2 10E-15

```

PAGE 00002

FILE: COMPAR SUBJECT A1 BELL-NORTHERN RESEARCH

C11	11	2	10E-15
C12	12	2	10E-15
C13	13	2	10E-15
R6	6	2	10000000
R7	7	2	10000000
R10	10	2	10000000
R11	11	2	10000000
R12	12	2	10000000
R13	13	2	10000000

END

FILE: COMPAR_2 SUBCCT A' BELL-NORTHERN RESEARCH

PAGE 0001

```

PARALLEL WRITE CIRCUIT
*SEE COMPAR SUBCCT A FOR NODE LIST
*
*
*
*
* MISN1 1 3 2 1 L5.0 W1.5 3 3 PR1 3 3 REPLACED B. LOAD
* DEVICE MISN15 BL2 HAS MISN1 AS ITS WEAK DRIVER FOR WRITE SIMULATION
* MISN2 1 8 1 L1.5 W2 6 6 PR1 6 6
* MISN3 1 6 9 1 L1.5 W3 6 6 PR1 6 6
* MISN4 1 14 15 1 L1.5 W3 6 6 PR1 6 6
* MISN5 1 16 14 1 L1.5 W15 30 30 PR1 30 30
* MISN6 1 16 1 1 L1.5 W240 2000 2000 PR1 2000 2000
* MISN7 1 1 16 1 L1.5 W189 400 400 PR1 400 400
* MISN8 1 1 16 1 L1.5 W189 400 400 PR1 400 400
* MISN9 2 2 16 2 L5.0 W95 200 200 PR1 200 200
* THAT WAS THE SIMULATED LOAD DEVICES ON THE ERROR BUS
* MISN IS NOW A LOAD DEVICE VCC-VT PRECHARGE ON BITLINES
* MISN1 2 1 1 4 L10 W1.5 3 3 PR1 3 3
* MISN2 2 3 5 6 L1.5 W4 8 8 PR1 8 8
* MISN3 2 4 5 7 L1.5 W4 8 8 PR1 8 8
* MISN4 2 7 8 10 L1.5 W6 12 12 PR1 12 12
* MISN5 2 10 12 2 L1.5 W6 12 12 PR1 12 12
* MISN6 2 6 9 11 L1.5 W6 12 12 PR1 12 12
* MISN7 2 11 18 2 L1.5 W6 12 12 PR1 12 12
* MISN8 2 14 6 12 L1.5 W6 12 12 PR1 12 12
* MISN9 2 14 7 13 L1.5 W6 12 12 PR1 12 12
* MISN10 2 12 8 2 L1.5 W6 12 12 PR1 12 12
* MISN11 2 13 9 2 L1.5 W6 12 12 PR1 12 12
* MISN12 2 16 17 2 L1.5 W25 50 50 PR1 50 50
* MISN13 2 14 16 1 L1.5 W2 4 4 PR1 4 4
* MISN14 2 14 16 2 L5.0 W2 3 3 PR1 3 3
* MISN15 2 1 1 3 L10 W1.5 3 3 PR1 3 3

```

```

C6 6 2 10E-15
C7 7 10E-15
C10 10 4 10E-15
C11 11 2 10E-15
C12 12 2 10E-15
C13 13 2 10E-15
R6 6 2 10000000
R7 7 2 10000000
R10 10 2 10000000
R11 11 2 10000000
R12 12 2 10000000
R13 13 2 10000000

```

PAGE 00001

FILE: COMPAR_3 SUBCCT A1 BELL-NORTHERN RESEARCH

PARALLEL WRITE CIRCUIT

SEE COMPAR SUBCCT A FOR NODE LIST

MISPI 1 3 2 1 L5.0 W1.5 3 3 PRI 3 3
 MISPI 1 7 8 1 L1.5 W3 6 6 PRI 6 6
 MISPI 1 6 9 1 L1.5 W3 6 6 PRI 6 6
 MISPI 1 14 15 1 L1.5 W3 6 6 PRI 6 6
 MISPI 1 16 14 1 L1.5 W15 30 30 PRI 30 30
 MISPI 1 16 1 1 L1.5 W940 2000 2000 PRI 2000 2000
 MISPI 1 16 1 1 L1.5 W189 400 400 PRI 400 400
 MISPI 2 2 16 2 L5.0 W95 200 200 PRI 200 200
 THAT WAS THE SIMULATED LOAD DEVICES ON THE ERROR BUS

M1 1 2 4 1 2 L1.5 W3 3 3 PRI 3 3
 M1SN2 2 3 5 6 L1.5 W4 8 8 PRI 8 8
 M1SN3 2 4 5 7 L1.5 W4 8 8 PRI 8 8
 M1SN4 2 7 8 10 L1.5 W6 12 12 PRI 12 12
 M1SN5 2 10 18 2 L1.5 W6 12 12 PRI 12 12
 M1SN6 2 6 9 11 L1.5 W6 12 12 PRI 12 12
 M1SN7 2 11 18 2 L1.5 W6 12 12 PRI 12 12
 M1SN8 2 14 6 12 L1.5 W6 12 12 PRI 12 12
 M1SN9 2 14 7 13 L1.5 W6 12 12 PRI 12 12
 M1SN10 2 12 8 2 L1.5 W6 12 12 PRI 12 12
 M1SN11 2 13 9 2 L1.5 W6 12 12 PRI 12 12
 M1SN12 2 16 17 2 L1.5 W25 50 50 PRI 50 50
 M1SP7 1 14 16 1 L1.5 W2 4 4 PRI 4 4
 M1SN13 2 14 16 2 L5.0 W2 3 3 PRI 3 3
 M1SN14 2 1 1 3 L1.0 W1.5 3 3 PRI 3 3
 M1SN15 2 1 17 6 L1.5 W2 4 4 PRI 4 4
 M1SN16 2 1 17 7 L1.5 W2 4 4 PRI 4 4
 M1SN17 2 6 17 7 L1.5 W2 4 4 PRI 4 4

C6 6 2 10E-15
 C7 7 2 10E-15
 C10 10 2 10E-15
 C11 11 2 10E-15
 C12 12 2 10E-15
 C13 13 2 10E-15
 R0 6 2 10000000
 R7 7 2 10000000
 R10 10 2 10000000
 R11 11 2 10000000
 R12 12 2 10000000
 R13 13 2 10000000

The parallel test access port circuit that was simulated is shown in Figure A.3.1. The simulation circuit netlist SUBCCT and input stimulus LOGIC files are also included in this appendix. The simulation results for a parallel write operation and for parallel data compare operations with and without error detection are shown in Figure A.3.2-4.

Consider the parallel write simulation shown in Figure A.3.2. The two bitlines, BLT1 and BLT2 are precharged to a threshold voltage below V_{CC} . At 10ns, PWRT turns on and at 20ns, EDB is driven high allowing BLT1 to be pulled low. After the bitline differential is created, PTST is activated to transfer the bitline differential onto the cell array bitlines, BL1 and BL2, to overwrite the differential developed by the cell prior to bitline sense and restoration. This activation of PTST is not shown in this simulation. The ERROR line rises during the parallel write operation, and MTCH consequently falls. This is the expected operation of the parallel test access port circuitry during a parallel write operation.

Figure A.3.3 illustrates the circuit behaviour during a parallel compare operation when an error is detected. PRE and PREB release the circuit for the compare operation at 5ns. PTST rises and the BL1 and BL2 signals drive the comparator chains. The expected data signal, ED, rises at 20ns and since the expected data does not match the actual data according to the conventions of the circuit, the ERROR signal rises at 25ns and is latched. The circuit is reset at 55ns when the precharge signals are turned on. Figure A.3.4 shows a similar sequence of signal timing except that EDB is driven high and the expected data and actual data match under this condition. ERROR remains low.

**Figure A.3.2 Parallel Test Port Simulation Plot
Parallel Write**

**Figure A.3.3 Parallel Test Port Simulation Plot
Error Detection**

Figure A.3.4 Parallel Test Port Simulation Plot
No Error Detection

2.3

1.5

0.7

-0.075

EDB COMPAR 3
EDB COMPAR 2
EDB COMPAR 1
ERROR COMPAR

6.00E-08

5.00E-08

4.00E-08

3.00E-08

2.00E-08

1.50E-08

1.00E-08

5.00E-09

SEC

5.50E-08

4.50E-08

3.50E-08

2.50E-08

1.50E-08

1.00E-08

5.00E-09

Appendix 4

Shift Register Decoder Simulation

Numbers refer to the transistor designation MISN, P in SUBCCT file.

Figure A.4.1 Shift Register Decoder Simulation Schematic

FILE: F0W2N LOGIC A1 BELL-NORTHERN RESEARCH

*TITLE CIRCUIT SIMULATION: ROW DECODER SHIFT REGISTER TEST

*CIRCUITS THAT CAN BE SIMULATED

*MACRO, ON, DESIG

*CDSR01, VDD, VSS, XEN, X4, X5, X6, X7, WLB1, SH, SHD, /
X11, /12, X13, X14, X15, SHIS0, BWRT, SHRAS, X19, WLBJ, XEN0, /
X22

*MACRO, OFF

*INPUT STIMULI

VDD, VSOURCE, DC, 5.0
VSS, VSOURCE, DC, 0.0
\$V1, VSOURCE, DC, 0.0
\$V2, VSOURCE, DC, 0.0
\$V19, VSOURCE, DC, 5.0
\$V20, VSOURCE, DC, 0.0

*PIECE WISE LINEAR INPUT WAVEFORMS

XEN, VSOURCE, PWL, 0.0, 10e-09, 0, /
12e-09, 0, /
45e-09, 0, /
47e-09, 0, /
60e-09, 0, /
63e-09, 0, /
65e-09, 0, /
85e-09, 0, /
87e-09, 0

SH, VSOURCE, PWL, 0.0, 20e-09, 0, /
22e-09, 5, /
30e-09, 5, /
32e-09, 0, /
110e-09, 0, /
112e-09, 5, /
120e-09, 5, /
122e-09, 0

SHR, VSOURCE, PWL, 0.0, 2e-09, 5, /
10e-09, 5, /
12e-09, 0, /

PAGE 00001

SUBJECT A1 BELL-NORTHERN RESEARCH

HOW DECODER SHIFT REGISTER CIRCUIT FILE

THE NODES ARE AS FOLLOWS:

11 VCC
 12 VCC
 13 XEN (DECODER ENABLE)
 14 WBI (THE ADJACENT WORDLINE)
 15 SH (SHIFT CLOCK)
 16 SHB (SHIFT BAR CLOCK)
 17 SHISO (ISOLATE CLOCK FOR SHIFT OPERATION)
 18 BWRI (BULK WRITE CLOCK FOR MEMORY INITIALIZATION)
 19 SHRAS (CLOCK TO DISABLE THE WORDLINE DRIVER DURING RAS
 INACTIVE PERIODS)
 20 WBJ (THE ACTIVE WORDLINE FOR NORMAL OPERATION)
 21 XENO (RESETS WORDLINE FOR SELECTED DECODER. ALSO USED TO
 RESET ALL NODES OF SHIFT REGISTER. MUST BE INACTIVE DURING
 ENTIRE SHIFT OPERATION)
 22 ADDU NODE FOR MASTER SLAVE FLIP FLOP

MISP1 1 4 3 1 L1.5 W4 8 8 PR1 8 8
 MISP2 1 6 3 1 L1.5 W4 8 8 PR1 8 8
 MISP3 1 12 11 1 L1.5 W4 8 8 PR1 8 8
 MISP4 1 13 12 1 L1.5 W4 8 8 PR1 8 8
 MISP5 1 14 22 1 L1.5 W4 8 8 PR1 8 8
 MISP6 1 15 14 1 L1.5 W16 32 32 PR1 32 32
 MISP7 1 20 19 1 L1.5 W25 50 50 PR1 50 50
 MISP8 1 11 21 1 L3.0 W1.5 3 3 PR1 3 3
 MISP9 2 4 2 5 L1.5 W4 8 8 PR1 8 8
 MISP10 2 6 1 5 L1.5 W4 8 8 PR1 8 8
 MISP11 2 5 1 7 L1.5 W4 8 8 PR1 8 8
 MISP12 2 7 3 2 L1.5 W4 8 8 PR1 8 8
 MISP13 2 6 3 11 L1.5 W4 8 8 PR1 8 8
 MISP14 2 8 9 11 L1.5 W4 8 8 PR1 8 8
 MISP15 2 13 10 11 L1.5 W2 4 4 PR1 8 8
 MISP16 2 12 11 2 L1.5 W4 8 8 PR1 8 8
 MISP17 2 13 12 2 L1.5 W4 8 8 PR1 8 8
 MISP18 2 14 22 2 L1.5 W4 8 8 PR1 8 8
 MISP19 2 15 14 2 L1.5 W8 16 16 PR1 16 16
 MISP20 2 15 16 19 L1.5 W8 16 16 PR1 16 16
 MISP21 2 19 16 2 L1.5 W4 8 8 PR1 8 8
 MISP22 2 20 19 2 L1.5 W12 24 24 PR1 24 24
 MISP23 2 1 17 19 L1.5 W4 8 8 PR1 8 8
 MISP24 2 22 9 15 L1.5 W2 4 4 PR1 8 8
 MISP25 2 12 10 22 L1.5 W4 8 8 PR1 8 8

CWL 20 2 1E-12

CWL 8 2 1E-12

WORDLINE LOAD CAPACITANCES SET AT 1PF

PAGE 00002

TITLE: ROSRN SUBJECT: A1 BELL-NORTHERN RESEARCH

A4 4 2 10000000
A5 5 2 10000000
R6 6 2 10000000
R7 7 2 10000000
W11 11 2 10000000
P19 19 2 10000000
C22 22 2 10000000
L4 4 2 10E-15
C5 5 2 10E-15
C6 6 2 10E-15
C7 7 2 10E-15
C11 11 2 10E-15
C19 19 2 10E-15
C22 22 2 10E-15

The shift register circuit that was simulated is shown in Figure A.4.1. The simulation circuit netlist SUBCCT and input stimulus LOGIC files are also included in this appendix.

The simulation results, shown in Figure A.4.2, illustrate the following sequence of events which verify the functionality of the circuit. Beginning at 0ns, the logic '0' state of WLBI, the adjacent wordline driver input, is shifted into the latch circuit through the pass transistor MISN6 driven by SH at 20ns. XEN0 and SHB are driven high before the shift operation to allow the data to propagate into the latch. When SHISO is driven high at 40ns, WLBJ goes into its active low state to illustrate the activation of the wordline driver. At 60ns, SHISO is driven low and SHRAS driven high to reset the wordline signal WLBJ high. The wordline driver is released again at 80ns to drive WLBJ low. From 100ns to 130ns, the high state of WLBI is shifted into the latch circuit using the same sequence of SHB, XEN0, and SH as was used to shift in the logic '0' state. When SHISO and SHRAS release the wordline driver circuit at 130ns, WLBJ remains in an inactive high state. At 140ns, SHISO returns low and BWRT is driven high to begin the bulk write operation. WLBJ goes to a logic '0' state until BWRT and SHRAS are reset at 170ns. This simulation verifies the shifting of both wordline states, the wordline reset operation, and the bulk write activation of the wordline: all the functions that are required of the shift register decoder during test and mission mode as well.

Figure A.4.2 Shift Register Decoder Simulation Plot

EXPERIMENTAL
CHISORCOWS2N

CHISORCOWS2N

2.40E-07
SEC

END

0/2·03·9/2

FIN