

Fault-Tolerance Considerations for Redundant Binary-Tree-Dynamic Random-Access-Memory (RAM) Chips

Bruno Ciciani

University of Rome II, Rome

Key Words — Fault-tolerant memory, Memory performability, Memory yield, Yield evaluation, VLSI chip design, VLSI memory

Reader Aids —

Purpose: Widen state of the art

Special math needed for explanations: Basic probability

Special math needed to use results: Same

Results useful to: Memory-chip designers, reliability engineers.

Summary & Conclusions — The binary-tree-dynamic RAM (TRAM) (Patent application #60882 on the TRAM architecture is pending with the US Patent Office) architecture has been proposed to overcome the performance and testing time limits of the traditional architecture of memory chips. A 64 Mb prototype of this architecture is being built by a DRAM manufacturer in Japan. This paper investigates manufacturing yield and operational performance of redundant TRAMs with respect to variation of tree depth and redundancy level. For this purpose, a based chip area, a yield and operational performance FoM (figure of merit) allowing the comparison of various choices has been formulated and used.

The yield is evaluated by a new Markov-chain based model. The memory operational performance has been analyzed by an innovative technique that substitutes the notion of chip state at the end of the mission time with the cumulative work performed by the chip during the mission time (performability). The FoM is —

- very straightforward and extremely easy to use in parametric studies of chip yield and operational performance vs redundancy level and reconfiguration strategy
- quite versatile for inclusion in CAM/CAD programming environments
- is appropriate for both VLSI chip designers and users in choosing the most suitable architecture.

An optimum value of the tree depth and redundancy level were found for a given RAM size, the adopted reconfiguration strategy, and the kind of redundancies. This is the result of a tradeoff between yield and operational performance. Indeed, independently of the number of spot manufacturing faults in the memory leaf, it is possible to find a redundancy level that can tolerate it; then the lesser the tree-depth, the greater the average wafer area needed to produce an acceptable chip. But, on the other hand, the bigger the tree-depth, the better the operational performance.

To evaluate the FoM, some assumptions, derived from the literature and experience, were adopted. While some of these assumptions can be changed, the methodology remains the same.

1. INTRODUCTION

Low reliability & availability and low yield (fraction of good chips out of a wafer [16]) are problems of increasing

importance as the density of memory chips increases [8, 28]. To enhance memory reliability and other performance measures such as memory-capacity availability, two main strategies of dynamic reconfiguration can be used:

- stand-by,
- graceful degradation (fail-soft) [8].

To improve yield appreciably, semiconductor manufacturers have employed redundancy [22]. Redundancy, however, is not free of the penalties:

- insertion of spare rows and columns contributes to a larger chip area,
- degradation of performance due to a longer access time [28].
- loss of productivity, viz, the increase in chip density can result in the failure of the sparing circuitry [28].

One way to overcome this drawback is to proceed from a traditional memory architecture to new types, eg, binary-tree architectures. Indeed, binary-tree architecture is not only versatile for parallel processing applications [10,24], it is also good for memory chips. Jarwala & Pradhan [12] have shown that large dynamic RAMs (random access memories), implemented according to this architecture (which partitions the RAM into modules, each appearing as the leaf-node of a binary interconnect network), can be faster (in terms of lower access time as well as reduced refresh time) than the traditional implementations. Besides, testing time is reduced thanks to the smaller size of the array under examination as well as to parallel testing. These benefits are obtained at only a small increase in chip area.

This paper investigates the tradeoff of yield vs operational performance of redundant binary-tree RAM as a variation of the tree depth and the redundancy level. This is done through analytic models that:

- evaluate the effectiveness of the proposed *static* and *dynamic* fault-tolerant techniques in increasing yield and improving performance
- find a tradeoff between the two.

As the fault-tolerant techniques can require hardware redundancy, a convenient benefit/cost FoM has to deal with chip-area cost. A better benefit/cost FoM includes the access time for the memories and/or a testability index (as testing-cost at different chip processing phase [18], or testing time, or fault coverage), but it poses the problem of how to weigh these indices against the cost factor. This paper formulates a FoM (named C.FoM by the editors for ease and clarity of use) that generalizes those already proposed in [13, 16, 19]. In particular, while these FoM's are based on evaluating the chip state at the end of the mission time, the C.FoM also considers the cumulative work/area performed during the mission time [2,4,5,21].

Finally, area is evaluated according to approaches in [11,12], with some differences to account for the reconfiguration strategy and associated redundancies. The yield is evaluated by a straightforward model which overcomes many of the drawbacks of the existing yield models [5]. Section 3 presents the TRAM architecture and properties from [12]. Section 4 presents the fault-tolerant techniques adopted for the TRAM architecture. Section 5 describes the C.FoM. Section 6 presents numerical results along with further discussion. Concluding remarks appear in section 7. The appendices describe the chip-area and yield-evaluation methods.

2. ASSUMPTIONS & NOTATION

Assumptions

A. Yield Evaluation

1. There are no gross defects.
2. The spot manufacturing defects (faults) are distributed on the wafer according to the negative binomial distribution.
3. Some faulty components are substituted by spare ones.

B. Performability Evaluation

1. There are only operational hardware permanent faults.
2. The memory components fail statistically independently with constant failure rates.
3. Faulty components are neither repaired nor substituted during the mission time.
4. The reward rate associated with each state of the memory chip is time-independent.

C. Component Replacement

1. Replacement of faulty components by redundant ones (word or bit lines) in the leaf-node is by fuse elements, as used in conventional architecture by AT&T Bell Labs, Hitachi, Toshiba, and Mitsubishi [22].

Acronyms

RAM	random access memory
TRAM	binary tree dynamic RAM
BITS	built-in-test structure
C.FoM	figure of merit recommended in this paper
XOR	exclusive OR

Notation

N	RAM size
Q	number of leaf nodes
l	tree depth
A_{TRAM}	chip area
Y	chip yield
F_S	A_{TRAM}/Y : wafer-area investment to obtain an acceptable chip
M	$\{M(n), n=0,1,\dots\}$: an acyclic Markov chain
E	$\{0,1,\dots,m\}$: the state space of M
P	$[p(i,j)]$: transition-probability matrix for E

X	number of manufacturing-faults/chip
α	manufacturing-fault clustering parameter
λ	average number of total manufacturing-faults/chip
λ_i	average number of type- i manufacturing-faults/chip
A, A_k	physical area of a general, or a type- k , circuit
B_{ik}	type- i manufacturing-fault density of the type- k circuit
T	$[0,t]$: time interval for observing memory performance
Z	$\{Z(\tau), \tau \in T\}$: an acyclic transient homogeneous stochastic Markov process
S	$\{0,1,\dots,m\}$: the state space of Z ; $0 \equiv$ absorbing state (set of chip unacceptable configurations), $m \equiv$ fault-free state
Q	$[Q(i,j)]$: transition probability matrix between states of Z ; this matrix is low triangular, due to the acyclic assumption
λ_i^d	transition rate of the Markov process in state i , $i \in S$; $\lambda_0^d=0$
λ_F^d	memory-leaf operational fault-rate
λ_{INT}^d	interconnection structure operational fault-rate
$f(i)$	reward rate associated with state i , $i \in S$
t	system mission time
f_{min}	performance level below which the chip becomes unacceptable
$F(Z(t))$	cumulative reward over $[0,t]$
m	process state at $t=0$
$K_D(t,m,I_{\text{min}})$	performability index: $\Pr\{F(Z(t)) > I_{\text{min}} Z(0) = m\}$
$F_D(t,m,H_{\text{min}})$	C.FoM: $\Pr\{F(Z(t))/F_S > H_{\text{min}} Z(0) = m\}$
H_{min}	I_{min}/F_S

Other, standard notation is given in "Information for Readers & Authors" at the rear of each issue.

3. BINARY-TREE DYNAMIC RAM OVERVIEW

3.1 Architecture

As specified [12], the TRAM of size $N=2^n$ is divided into modules, each of which appears as a leaf-node in a binary tree. The depth of the tree, the number of nodes, and their size are related by:

$$N = 2^n = 2^{mf} \times 2^{l-1} \quad (1a)$$

$$= \text{MF} \times Q \quad (1b)$$

Notation

n	number of address lines
l	depth of the tree
MF	2^{mf} : size of each leaf-node
Q	2^{l-1}
B_i	memory-node i , $0 \leq i < Q$
$A_{i,j}$	address of cell j within B_i , $0 \leq j < \text{MF}$.

In this TRAM the nodes are laid out using the well-known H-tree [12]. As organized in figure 1, the leaf nodes are in

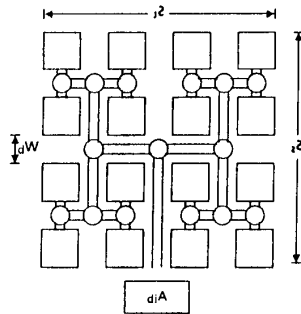


Figure 1. Area model for the redundant TRAM architecture.

groups of four, forming an H-tree, which are further connected hierarchically. Each non leaf-node in the TRAM is a switch node, which is a simple 1-out-of-2 decoder with buffers. The memory nodes are memory modules which have the traditional 4-quadrant organization with independent control units. Thus a TRAM chip with ‘depth = 1’ is equivalent to a conventional memory chip [3]. Furthermore, when ‘depth > 1’ the address/data/control bus is connected to the root-node which in this case is a simple switch node. The root-node decodes the most important part of the address and generates a left- or a right-subtree select. The other signals are buffered and propagated down the tree. This action occurs repeatedly at each level until a single memory node is selected. The remaining address bits are then used to select a cell within the node.

The device has two modes of operation:

- normal, in which it functions as a traditional RAM
- test, in which the tester verifies the presence of faults through a testing procedure.

Additional devices are introduced to support fault detection. Between every pair of nodes B_i and B_{i+1} ($0 \leq i < Q-1$), a comparator C_i (XOR gate) is placed such that C_i compares the data between nodes B_i and B_{i+1} . The output of each comparator is an input to a distributed NOR gate, whose output is brought out as a FAIL line. The structure of the comparators and the NOR gate has been referred to as built-in-test structure (BITS).

3.2 Properties

The TRAM architecture [12] has potential advantages over conventional architecture, and is briefly summarized here.

- Easily Testable. Partially self-testing with small testing times for very large RAM [12]. This architecture results in important savings in testing time, compared to conventional architecture. For memories with more than 16 leaves the saving in testing time is more than 90%. Using the algorithm in [23], the TRAM can be tested in $(16Q + 8 + 36N + 24N \log(N))$ operations. The proposed testing procedure has high fault coverage: all detectable stuck-at faults in the tree decoder, stuck-at faults as well as the changing the functionality of the XOR gates in the BITS, and stuck-at, two-coupling,

and limited three-coupling faults in the memory nodes can be detected.

- Low Overhead. The additional area required for large RAMs, without redundancy, is typically 8-20% more than for conventional architecture.
- Improved Performance.
 - For large RAMs, this architecture is faster compared to conventional architecture, with a potential reduction in access time of about 30%.
 - Refreshing the nodes in parallel substantially reduces the amount of time the RAM is not available.
- Partitionable. Partitionability makes it possible to generate partially good products. This can improve the effective yield.
- Increased Reliability. Single-node failures are easily tolerated. If there is redundancy, the built-in test capabilities can reduce the mean time to repair appreciably; otherwise graceful degradation is straightforward.

3.3 Overview

This paper investigates the fault tolerance capabilities of the TRAM architecture. In particular, section 4 focuses on the fault tolerant techniques that I believe will increase yield and enhance operational performance.

4. FAULT-TOLERANT CHARACTERISTICS

The TRAM architecture introduces —

- both *static* and *operational* fault-tolerant strategies that can reconfigure the chip without loss of memory capacity
- mechanisms that, in the presence of faults can reconfigure the chip to obtain partially good chips.

While the stand-by strategy must use spares to substitute faulty components, the graceful-degradation strategy can perform memory remapping, either internally to the chip, or externally by specialized hardware (eg, memory management unit) or software (with some performance penalty).

In this section, stand-by strategy is used only for yield enhancement because:

- it avoids the insertion of complex specialized hardware (for storing fault conditions and for carrying out the necessary reconfiguration) which degrades the memory performance due to increased access time [8]
- the probability of operational faults (as compared to manufacturing faults) is considerably lower by as much as a factor of 100 or more [14,16].

The redundancies are to be only at the leaf level. Redundancies at the interconnection bus or at switching node level are not considered here since both kinds of components have a higher yield than memory and other logic-support circuits (indeed, the interconnects require fewer mask layers, while the switching nodes are very simple logic circuits). According to this technique (in assumption C#1), leaf static reconfiguration (elimination of faulty lines and replacement by spare ones, the

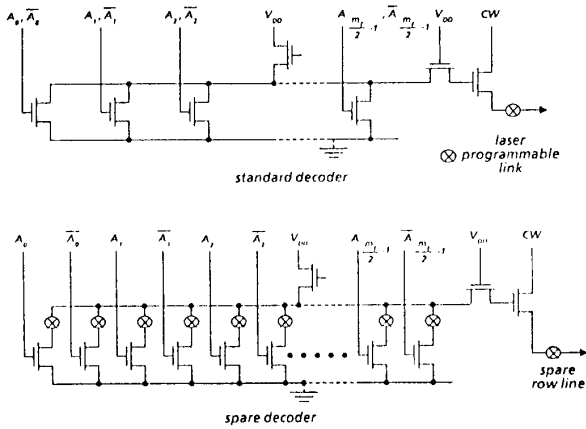


Figure 2. Row & Column Replacement Scheme

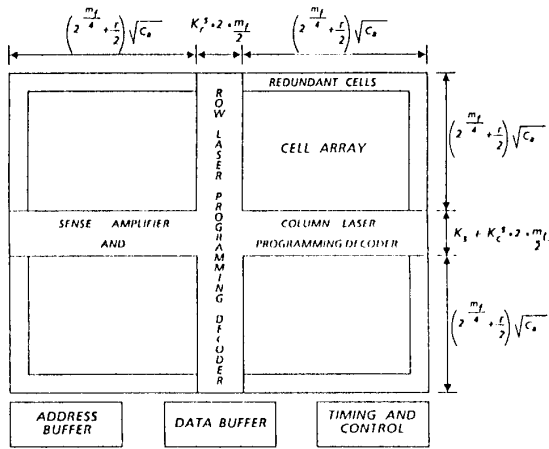


Figure 3. Area Model for a Leaf-Node of the Redundant TRAM Architecture.

number of redundant word and bit lines is r ; see appendix A) takes place by the firing of fuse elements (see figures 2 & 3).

Based on the presence or absence of redundancies (eventually due to be exhausted) the *static* reconfiguration strategy (at manufacturing time) is organized in the following manner. If a fault hits a component with available spares, then the *static* reconfiguration strategy decisions are a function of the nature of the detected fault and the redundancy level [5]; otherwise the entire memory chip is lost.

Instead, the *dynamic* reconfiguration strategy (during chip operational conditions) allows only memory graceful-degradation, and is organized as follows:

- if the fault hits a node-leaf, then the leaf disconnection from the structure is performed
- if the fault hits a line of the interconnection bus (address/control/data), or a switching node or the input buffers, then the entire memory chip is lost.

The *static & dynamic* reconfiguration processes are activated by the testing procedure in [12]. This procedure can detect faults caused by manufacturing defects and operational permanent faults. The transient fault presence [20] is not considered here, in order to simplify the presentation of this C.FoM. Indeed, the TRAM architecture allows the incorporation of redundant codes, either for fault mask or for fault detection only [11]. Moreover, with respect to the original TRAM architecture, the *static* reconfiguration strategy needs only laser programmable fuses in the leaf nodes, and no additional circuitry is required for the *dynamic* reconfiguration strategy. In the latter strategy this paper also assumes that for leaf node failure there is no internal memory remapping, and the lack of one or more leaf nodes is known at the memory management unit level (this information is passed to the memory management unit by the tester unit). Because this solution always permits use of the parallel test approach, it retains all the testability advantages of the TRAM [12]. Moreover, it does not affect memory access time.

5. THE PROPOSED FIGURE OF MERIT

The effectiveness of fault-tolerant techniques can be evaluated by using the *minimal performance* index [6,13,15,16]:

$$P_D(t, m, f_{\min}) = \Pr\{f(Z(t)) \geq f_{\min} | Z(0) = m\}. \quad (2)$$

For example, in digital controls of dynamic systems, f_{\min} specifies the chip minimal throughput value to attain a sufficiently high sampling rate which guarantees an acceptable process control level [2]. Similarly, in memory chips, f_{\min} gives the minimal capacity of the chip guaranteeing its user's operation. Indeed, memory affects the performance of computer systems in two important ways [17]:

1. Almost every system has a "constrained memory", viz, a limit on the number of *threads of control* that can be active simultaneously, imposed by the available capacity of the memory.

2. There is an overhead associated with memory management. For example, swapping a user between primary memory and secondary storage places service demands on the I/O system as well as on the CPU; this interaction's service demand on the swapping device is a function of the available capacity of the memory.

The drawback of the *minimal performance* index in (2), is that it does not provide information on *how* the chip performance degrades over time. Figure 4 is an example which shows the $f(Z(t))$ random behavior for two systems (S1 & S2). In both cases, the reward rate is above the f_{\min} level during the mission time. Eq (2) gives the probability of this event, but fails to provide the probability that the S1 path remains above the S2 path in the course of the mission time. On the other hand, having this kind of information could be very important since, according to figure 4, S1 ensures a higher quality of service than does S2 (eg, a greater capacity in the memory example,

and a more accurate control level in the *dynamic* control example).

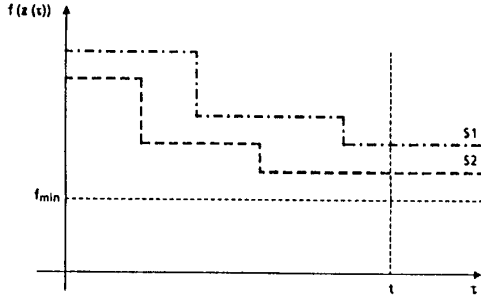


Figure 4. Random Path Behaviors of the Performance of 2 Systems [over mission time]

Therefore, a more appropriate index considers the $f(Z(t))$ random behavior during the mission, in addition to its remaining above a minimal performance level. For this purpose, the *performability index*, $K_D(t, m, I_{\min})$ — based on the integral of $f(Z(t))$ over the mission time [2, 21] — can be used in place of (2); K_D gives the probability that the chip's amount-of-work during the mission is above a minimum acceptable level. The *performability index* gives the system *reliability (availability)*, for a repairable system) when all the acceptable Markov process states have 'reward-rate = 1'.

The inclusion of fault-tolerant mechanisms in the chips requires an increase in the chip-area. Therefore, various modeling techniques have been introduced in order to evaluate the cost/benefit ratio [6,13,15,16]. This paper, given the reconfiguration strategy, for a convenient comparison of different tree depth and redundancy levels, relates K_D to the hardware investment which has already been made, viz, average wafer area needed to produce an acceptable chip, by the parameter, F_S . An efficient methodology to evaluate the yield is given in [5]. Appendix B shows how to use this methodology to evaluate the yield of the redundant TRAM architecture.

K_D & F_S are combined to define the chip C.FoM, F_D .

The computation of F_D & K_D is made possible by rewriting them in the recursive form:

$$F_D(t, m, H_{\min}) = \exp(\lambda_m^d t) u\left(H_{\min} - \frac{f(m)t}{F_S}\right) + \sum_{j=0}^{m-1} Q(m, j) \int_0^t \lambda_m^d \exp(-\lambda_m^d \tau) \cdot F_D\left(t - \tau, j, H_{\min} - \frac{f(m)\tau}{F_S}\right) d\tau \quad (3)$$

$$u(x) \equiv \begin{cases} 1, & \text{if } x \leq 0 \\ 0, & \text{otherwise} \end{cases}$$

Term #1 on the r.h.s. of (3) gives the probability that the chip's amount of work/area is above H_{\min} when there are no transitions out of state m in $[0, t]$. On the other hand, if a transition out of state m takes place at time $\tau < t$ (after an amount of work $f(m)\tau/F_S$ has already been performed). Term #2 on the r.h.s. of (3) gives the probability that the remaining work/area is above the necessary level.

Efficient algorithms exist [7,26] for the computation of either the probability distribution or its moments.

5.1. TRAM Markov Process

The dynamic reconfiguration strategy adopted for TRAM, as explained in section 3, implies that, if a permanent fault hits a component of the interconnection structure — a line of the interconnection bus (address/control/data), or a switching node, or the input buffers) — then the entire memory chip is lost. Otherwise if the permanent fault hits a node-leaf, then this leaf is disconnected from the structure with consequent memory capacity degradation. For the fault detection (see section 4) the fault-testing procedure in [12] is assumed. Fault-testing procedure has a good fault coverage (c) but, obviously, $c < 1$.

In order to use (3) we have to identify the minimum capacity f_{\min} (under which the chip is considered unacceptable), the TRAM Markov process, and (for each state) the associated reward rate. Figure 5 shows the Markov process for a TRAM with Q leaves. If the TRAM reward rate is related to the memory capacity, then the reward rate to be associated with Markov process state- i is proportional to the memory capacity of the TRAM with i leaves ($f_i = i/Q$). On this basis, the states in figure 5 are labeled according to the number of fault-free leaves:

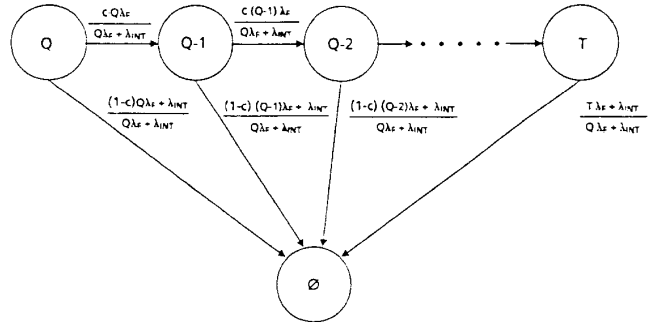


Figure 5. Markov-Process State-Transition Diagram [for a Binary Tree Dynamic RAM with Q leaves]

- Q is the initial fault-free state
- T is the last state which guarantees the minimal acceptable memory capacity
- \emptyset is the set of unacceptable chip configurations.

The transitions are labeled by transition-rate equations, in which three quantities appear:

c fault coverage
 λ_F^d fault-rate of a single leaf
 λ_{INT}^d considers the fault-rate of the tree interconnection structure.

State-to-state transitions are governed by the events of component faults and/or their detections. For example, transition from state $Q-1$ to state $Q-2$ takes place when a fault, detected by the testing procedure, is located in any one of the $Q-1$ fault-free leaves, while transition from $Q-1$ to state 0 takes place when a fault is located in one component of the tree interconnection structure or when a fault on the operating leaves is not detected.

I have no experimental data to extrapolate the dynamic fault-rates. For this purpose I hypothesize that λ_F^d and λ_{INT}^d are directly proportional to the area of the single leaf and the tree interconnection structure respectively:

$$\lambda_F^d = K_1 A_{leaf} \quad (4)$$

$$\lambda_{INT}^d = K_2 (N_{SWN} A_{SWN} + A_{BUS} + A_{ib}) \quad (5)$$

Notation

A_{leaf} leaf area
 N_{SWN} number of switch nodes (equal to $2^{n-mf} - 1$)
 A_{SWN} area of a switch node
 A_{BUS} area of the interconnection bus
 A_{ib} area of the input buffers
 K_1, K_2 constants.

6. NUMERICAL EVALUATION

Figure 6 shows the C.FoM for a 4 Mbit memory chip, vs the amount of work/area (H_{min}) performed during a mission time of 10^4 time units. Figure 7 shows the performability for the same 4 Mbit memory chip vs the amount of work (I_{min}) performed during the same mission time of 10^4 time units. The graphs in both figures are related to 4 leaf-numbers:

- 1 leaf (conventional memory organization)
- 4 leaves
- 16 leaves
- 64 leaves.

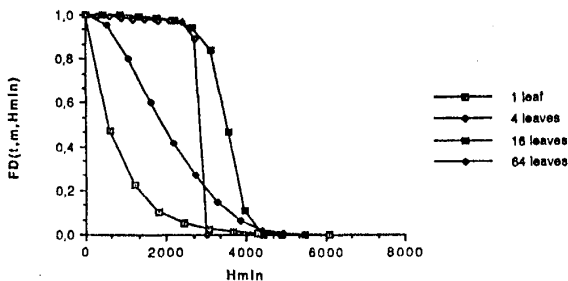


Figure 6. C.FoM for a 4 Mbit Memory Chip [$c = 1$]

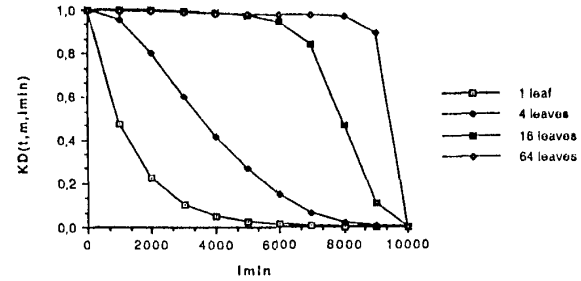


Figure 7. Performability for a 4 Mbit Memory Chip [$c = 1$]

The graphs are obtained with $c=1$, and according to the hypothesis that the chip is used until its fault-free leaves permit it to obtain a memory size with a value which is half of its initial value ($f_{min} = 1/2$). The fault-rates are obtained by fixing:

$$K_1 = 5.0 \times 10^{-4} \text{ faults/cm}^2 \cdot \text{unit_time}$$

$$K_2 = 0.2 \times 10^{-4} \text{ faults/cm}^2 \cdot \text{unit_time}$$

The area parameters are listed in table 3 (see below for the manufacturing-fault density). While it follows from figure 7 that it is possible, from the point of view of the performability index, that the redundant 64-leaf TRAM has the better behavior, figure 6 shows that, from the C.FoM view-point, the architecture with the better behavior is the one with 16 leaves. The better performability of the 64-leaf architecture is due to its greater granularity which permits it to degrade its operational performance more softly than the others. But this fact is not enough from the C.FoM view-point. The main reason is that the better operational performance behavior of the redundant TRAM with more leaves cannot counterbalance its wafer area investment (average wafer area needed to produce an acceptable chip). Indeed figure 8 shows the wafer area investment vs the redundancy level r ; the graphs are obtained for the same memory architectures and using the same parameters used to obtain the graphs of the previous figures, see table 2 and (B.4), (B.7), (B.8), (B.9). Figure 8 also shows that the redundant 64-leaf TRAM needs a bigger wafer area investment. This set of figures shows the effectiveness of the C.FoM in the tradeoff between operational performance improvement and wafer area investment (which in turn depends on yield). This tendency is proved by figure 9, which shows the C.FoM for the same architectures and parameters used to obtain the graphs of figure 6, but with $c=0.9$. Obviously the graphs of figure 9 show a lesser operational performance improvement than the corresponding graphs of figure 6, due to their lesser reconfiguration capability. The same behavior has been obtained for 1 Mbit and 16 Mbit memory chips with and without $c=1$.

Finally, figure 8 shows that, given the chosen parameters, from the point of view of the wafer area investment, there is no benefit in using redundant TRAM with more leaves — the same results can be obtained for different memory size and manufacturing fault clustering parameters. Indeed, it is possible to find a redundancy level that can tolerate any number of spot manufacturing faults in the memory leaf. Then the lesser

TABLE 3
Design Parameters

Cell area	(C_a)	$35 \mu\text{m}^2$
Standard row and column decoder pitch/bit	(K_r^s, K_c^s)	$25 \mu\text{m}$
Spare row and column decoder pitch/bit	(K_r^s, K_c^s)	$25 \mu\text{m}$
Pitch of metal in TRAM bus structure	(K_b)	$10 \mu\text{m}$
Depth of sense amp/bit	(K_s)	$100 \mu\text{m}$
Area per each bit address latch	(K_a)	$4000 \mu\text{m}^2$
Data latch area	(K_d)	$4000 \mu\text{m}^2$
Timing-&-control area	(K_t)	$100000 \mu\text{m}^2$

APPENDIX A

Area-Cost Analysis

The approach is based on [11, 12]; the analysis is repeated in order to consider the redundant rows and columns. Moreover, to simplify the modeling, I assume that the node leaf, as well as the TRAM chip, is organized in a square architecture.

This paper presents parametric studies of chip yield and operational performance vs redundancy level and tree depth for various memory sizes. For this purpose, I assume that the average number of manufacturing faults and operational faults are proportional to the chip area circuits (see appendix B and section 5, respectively). For this reason, the geometric values are directly in μm and in μm^2 instead of in λ (technology minimum dimension: $\lambda = k$ corresponds to $2k \mu\text{m}$) as proposed in [12]. The drawback of my approach is that for large memory sizes the die size might be physically unrealistic; on the contrary, one has to hypothesize a change of technology (and then λ a value) for each memory size, but in this case the drawback is the difficulty in determining a reasonable relationship between would relate the average number of manufacturing faults (operational faults) and the technology and consequently, to the memory size.

A.1. Leaf Node Geometric Area

The leaf node is based on classical 4-quadrant RAM architecture. Besides, the presence of row/column redundancies must be considered. Table 2 lists the circuit types present in the leaf node.

In order to consider the presence of fuses with the associated circuitry [25], from the point of view of the chip geometric area, one must consider a cell array with standard row decoders with disable fuses, spare row decoders, standard column decoders with disable fuses, spare column decoders, sense amplifiers, address/data buffers, timing & control unit (see figures 2 & 3).

Let there be r redundant rows and columns, for a node leaf with 2^{mf} addressable bits, then the array has $(2^{mf/2} + r)^2$ cells (for simplicity, assume mf is even); each cell has area = C_a (see table 3). The standard row/column decoder with disable fuses and spare row/column decoder are characterized by a width/(bit to be decoded), expressed as K_r , K_c , K_r^s , K_c^s respectively. The sense amplifier height is characterized by a

constant, K_s . The width is controlled by the spacing and size of the cell. The address/data buffers and timing & control area are given by K_a (per each bit to buffer), K_d , and K_t , respectively. Moreover, assume a location for the standard decoder parallel to the spare decoders, since the spare decoders have twice the input than the standard ones (see figure 2). The leaf-node area is:

$$A_{\text{leaf}} = [(2^{mf/2} + r) \times C_a^{1/2}] + K_r^s \times 2 \times mf/2 \\ \times [(2^{mf/2} + r) \times C_a^{1/2}] + K_c^s \times 2 \times mf/2 + K_s \\ + K_a \times mf/2 + K_d + K_t \quad (\text{A.1})$$

A.2. Redundant TRAM Geometric Area

In [12] the assumption is made concerning the bus-structure implementation of the TRAM architecture that the address bus is multiplexed. The lower address bits can be multiplexed; the upper bits propagate directly for subtree and leaf select. Therefore the bus carries $n - mf/2$ address lines, 1 data line, 2 Test & Fail lines, and 1 Read/-Write signal line. The comparators C_i present between every pair of nodes (see section 3.1), being simple XOR gates, are not taken into account in the geometric area evaluation because, given their location, they are masked by the spacing and size of the tree interconnection bus. The geometric area of the redundant TRAM architecture can be computed (see figure 1) by using the parameters:

- W_b width of the bus
- S_t length of the top of the chip
- S_s length of the side of the chip
- K_b pitch (the distance between neighboring signals)
- N_t nodes on the horizontal side; since mf is even, $N_t = Q^{1/2}$.

Therefore:

$$W_b = (n - mf/2 + 4)K_b \quad (\text{A.2})$$

$$S_t = S_s = N_t(A_{\text{leaf}})^{1/2} + (N_t - 1)W_b \quad (\text{A.3})$$

Therefore, the area of the nodes and the bus structure is $S_s S_t$. Some input buffers are required to drive the tree. Their area is estimated as:

$$A_{ib} = (n - mf/2 + 4)K_a + K_d \quad (\text{A.4})$$

Thus —

$$A_{\text{TRAM}} = S_s S_t + A_{ib} \quad (\text{A.5})$$

APPENDIX B

Yield Evaluation of Redundant TRAM Architecture

Notation

A_c	an array of numbers representing the sensitivity of a chip to random defects.
B_{ik}	manufacturing fault density of circuit type- k to generate fault type- i .
Y_{IS}	yield of the tree interconnection structure
Y_{LEAF}	yield of a leaf-node
λ_{IS}	total average number of manufacturing faults in the interconnection structure
λ_{IB}	average of manufacturing faults in the interconnection bus
λ_{SWN}	average of manufacturing faults in the switching nodes
λ_{AIB}	average of manufacturing faults in the input buffers.

Yield evaluation is very simple when there are no redundancies, ie, $\text{yield} = \Pr\{\text{there are no defects that cause the circuit not to meet its operational specifications}\}$. Instead, when there are redundancies, it is necessary to correlate the defect presence with the level of redundancy and the reconfiguration strategy. Since the majority of manufacturing defects can be classified as random spot defects [27] (caused by minute particles deposited on the wafer), the following derivation does not consider the gross yield evaluation problem.

The yield evaluation equation essentially consists of two terms [16,20,27]:

- the random defect (fault) statistics term
- the term providing the probability of chip acceptability given n defects.

This paper (as suggested in [27] and followed in [9,12,13,16]) uses the generalized negative binomial statistics defect model as the first term of the equation. The Markov-chain approach [5] improves on previous methods and puts together generalities, ease of computation and predictability in approximation levels; it is used as tool to evaluate the second term.

As demonstrated by [3,27], adequate evaluation of random fault statistics requires knowledge of:

1. The *defect types* ($1, 2, \dots, D$) from the manufacturing process.
2. The *defect densities* (d_1, d_2, \dots, d_D) for each defect type. d_i denotes the average number of type- i defects per unit area.
3. The *manufacturing fault types* ($1, 2, \dots, F$) originating from the defects. The average number of type i faults/chip is λ_i ($i = 1, 2, \dots, F$).
4. The *fault clustering* parameter, α .

To find the average number of faults per chip, [27] relates faults to defect densities through the *critical area* matrix:

$$\lambda = A_c \mathbf{d}$$

$$\lambda_i = \sum_{j=1}^D A_{ij}^c d_j \quad (\text{B.1})$$

This sensitivity to defects is obtained by calculating the circuit

area in which a defect must fall in order to cause a fault [12]. Thus in [3] the critical area matrix terms were evaluated by relating them to circuit types, $(1, 2, \dots, C_i)$ for $i = 1, 2, \dots, F$, causing type- i faults with their A_k and a *fault-sensitivity to defects* parameter (S_{ijk}): the ratio of the number of type- i faults to the average number of type- j defects (in circuit type- k).

$$A_{ij}^c = \sum_{k=1}^{C_i} S_{ijk} A_k \quad (\text{B.2})$$

$$\lambda_i = \sum_{j=1}^D \sum_{k=1}^{C_i} A_k S_{ijk} d_j \quad (\text{B.3})$$

If the defect densities and the fault-sensitivity-to-defects parameters are known, then it is possible use (B.3); otherwise (B.3) can be used to generate λ_i parametrically to the physical area of the circuits. Thus (B.3) can be rewritten:

$$\lambda_i = \sum_{k=1}^{C_i} A_k \sum_{j=1}^D S_{ijk} d_j = \sum_{k=1}^{C_i} A_k B_{ik} \quad (\text{B.4})$$

To evaluate the average number of faults, (B.4) is used in the following.

Given the modular organization of TRAM, the yield is:

$$Y = Y_{IS} \times Y_{LEAF}^Q \quad (\text{B.5})$$

As described in section 4 redundancies are not considered at the interconnection bus, the switching node level, nor the input buffers. Therefore the yield of the interconnection structure is $\Pr\{\text{there are no faults}\}$:

$$Y_{IS} = (1 + \lambda_{IS}/\alpha)^\alpha \quad (\text{B.6})$$

$$\lambda_{IS} = \lambda_{IB} + \lambda_{SWN} + \lambda_{AIB}$$

For each interconnection-structure manufacturing fault there is only one possible source, the bus line, the 1-out-of-2 decoder with buffers, and the buffers respectively. So we deduce that:

$$\lambda_{SWN} = N_{SWN} A_{SWN} B_1 \quad (\text{B.7})$$

$$\lambda_{IB} = A_{BUS} B_2 \quad (\text{B.8})$$

$$\lambda_{AIB} = A_{ib} B_3 \quad (\text{B.9})$$

$$N_{SWN} = 2^{n-mf} - 1$$

For the yield evaluation choose:

$$B_1 = 0.25 \text{ number-of-faults/cm}^2,$$

$$B_2 = 1.5 \text{ number-of-faults/cm}^2,$$

$$B_3 = 0.5 \text{ number-of-faults/cm}^2,$$

Instead, since in the leaf node there are some redundancies, it is necessary to correlate the fault (defect) presence with the level of redundancies and the adopted reconfiguration strategies. If the leaf node is organized as a traditional memory and the reconfiguration strategy is the same as that in [5], then it is possible to use the Markov chain in [5]. Then, too, the transition probability equations for the Markov chain are the same as those in [5]. The manufacturing fault types considered for the state transition are listed in table 1 and are the same as those in [27]. Table 2 lists the possible sources and the associated B_{ik} values for each manufacturing fault type.

Using the same approach, it is possible to determine the chip *apparent yield* (taking into account all the chips tested as good even if they are faulty) [16]. In this case, for the evaluation of transition probability $t(k,j)$ it is also necessary consider c for the fault-testing procedure adopted.

REFERENCES

- [1] R. Abbott, K. Kokkonen, R. I. Kung, R. J. Smith, "Equipping a line of memories with spare cells", *Electronics*, 1981 July 28, pp 127-130.
- [2] B. Ciciani, V. Grassi, G. Iazeolla, "Yield and performability evaluation of VLSI reconfigurable multiprocessor structures", *Computer Performance and Reliability*, 1988, pp 369-382; Elsevier Science Publishers.
- [3] B. Ciciani, G. Iazeolla, "A straightforward yield model for fault-tolerant VLSI memory chips", *IFIP TC-10 Conf. Design Methodology in VLSI and Computer Architecture*, 1988 Sep, pp 99-112; Pisa, Italy.
- [4] B. Ciciani, G. Iazeolla, "A yield model for the evaluation of topologically constrained chip architectures". *IEEE Int. Conf. Computer Design (ICCD)*, 1989 Oct, pp 443-446; Cambridge, Mass.
- [5] B. Ciciani, G. Iazeolla, "A Markov-chain based yield model for VLSI fault tolerant chips", *IEEE Trans. Computer-Aided Design*, vol 10, 1991 Feb, pp 252-259.
- [6] J. A. B. Fortes, C. S. Raghavendra, "Gracefully degradable processor arrays", *IEEE Trans. Computers*, vol C-34, 1985 Nov, pp 1033-1043.
- [7] A. Goyal, A. N. Tantawi, "Evaluation of performability for degradable computer systems", *IEEE Trans. Computers*, vol 36, 1987 Jun, pp 738-744.
- [8] K. E. Grosspietsch, "Schemes of dynamic redundancy for fault tolerance in random access memories", *IEEE Trans. Reliability*, vol 37, 1988 Aug, pp 331-339.
- [9] J. C. Harden, N. R. Straden, "Architectural yield optimization for WSI", *IEEE Trans. Computers*, vol 37, 1988 Jan, pp 88-110.
- [10] E. Horowitz, A. Zorat, "The binary tree as an interconnection network: Applications to multiprocessor systems and VLSI", *IEEE Trans. Computers*, vol 30, 1981 Apr, pp 247-253.
- [11] N. T. Jarwala, D. K. Pradhan, "Cost analysis of a chip error control coding for fault tolerant dynamic RAMs", *17th IEEE Fault Tolerant Computing Symp.*, 1987, pp 278-283.
- [12] N. T. Jarwala, D. K. Pradhan, "TRAM: A design methodology for high-performance, easily testable multimegabit RAMs", *IEEE Trans. Computers*, vol 37, 1988 Oct, pp 1235-1250.
- [13] I. Koren, M. A. Breue, "On area and yield considerations for fault-tolerant VLSI processor arrays", *IEEE Trans. Computers*, vol 33, 1984 Jan, pp 21-27.
- [14] I. Koren, D. K. Pradhan, "Introducing redundancy into VLSI designs for yield and performance enhancement", *15th IEEE Fault Tolerant Computing Symp.*, 1985, pp 330-334.
- [15] I. Koren, D. K. Pradhan, "Yield and performance enhancement through redundancy in VLSI and WSI multiprocessor systems", *Proc. IEEE*, vol 74, 1986 May, pp 699-711.
- [16] I. Koren, D. K. Pradhan, "Modeling the effect of redundancy on yield and performance of VLSI systems", *IEEE Trans. Computers*, vol 36, 1987 Mar, pp 344-355.
- [17] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik, "Quantitative system performance: computer system analysis using queuing network models", 1984; Prentice-Hall.
- [18] W. Maly, A. J. Strojwas, S. W. Directo, "VLSI yield prediction and estimation: A unified framework", *IEEE Trans. Computer-Aided Design*, vol CAD-5, 1986 Jan, pp 114-130.
- [19] T. E. Mangir, A. Avizienis, "Fault-tolerant design for VLSI: Effect of interconnect requirements on yield improvement of VLSI design", *IEEE Trans. Computers*, vol 31, 1982 Jul, pp 609-618.
- [20] T. E. Mangir, "Sources of failures and yield improvement for VLSI and restructurable interconnections for RVLSI and WSI: Part I - Sources of failures and yield improvement for VLSI", *Proc. IEEE*, vol 72, 1984 Jun, pp 690-708.
- [21] J. F. Meyer, "On evaluating the performability of degradable computing systems", *IEEE Trans. Computers*, vol C-29, 1980 Aug, pp 720-731.
- [22] W. R. Moore, "A review of fault-tolerant techniques for the enhancement of integrated circuit yield", *Proc. IEEE*, vol 74, 1986 May, pp 684-698.
- [23] C. A. Papachristou, N. B. Sahgal, "An improved method for detecting functional faults in semiconductor RAM's", *IEEE Trans. Computers*, vol C-24, 1985 Feb, pp 110-116.
- [24] A. D. Singh, "A reconfigurable modular fault tolerant binary tree architecture", *17th IEEE Fault Tolerant Computing Symp.*, 1987, pp 298-304.
- [25] Smith, Chlipala, Bindels, Nelson, Fisher, Mantz, "Laser programmable redundancy and yield improvement in a 64K DRAM", *IEEE J. Solid-State Circuits*, vol SC-16, 1981 Oct, pp 506-513.
- [26] R. M. Smith, K. S. Trivedi, A. V. Ramesh, "Performability analysis: Measures, an algorithm, and a case study", *IEEE Trans. Computers*, vol 37, 1988 Apr, pp 406-417.
- [27] C. H. Stapper, A. N. McLaren, M. Dreckmann, "Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product", *IBM J. Research & Development*, vol 24, 1980 May, pp 398-409.
- [28] Chin-Long Wey, F. Lombardi, "On the repair of redundant RAMs", *IEEE Trans. Computer Aided Design*, vol 6, No. 2, March 1987, pp 222-231.

AUTHOR

Prof. Bruno Ciciani; University of Rome II; Electronics Engineering Dept.; via Raimondo 8; 00173 Rome ITALY. [temporary address]

Bruno Ciciani is an Associated Professor in Computer Science. His research includes distributed computer systems, languages for parallel processing, fault-tolerant computing, and evaluation of computer system performance & reliability.

Manuscript TR89-074 received 1989 May 17; revised 1990 July 7; revised 1990 November 30; revised 1991 March 11.