JOURNAL OF ELECTRONIC TESTING: Theory and Applications, 3, 235-250 (1992)

An Efficient Design of Embedded Memories and their Testability Analysis using Markov Chains*

P. MAZUMDER
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109

J.H. PATEL
Coordinated Science Laboratory, Unverisity of Illinois, Urbana, IL 61801

Received August 23, 1991. Revised June 24, 1992.
Editor: V.K. Agarawal

Abstract. This article presents a design strategy for efficient and comprehensive random testing of embedded random-access memory (RAM) where neither are the address, read/write and data input lines directly controllable nor are the data output lines externally observable. Unlike the conventional approaches, which frequently employ on-chip circuits such as linear feedback shift register (LFSR), data registers and multibit comparator for verifying the response of the memory-under-test (MUT) with the reference signature of a fault-free *gold unit*, the proposed technique uses an efficient testable design, which helps accelerate test algorithms by a factor of $0.5\sqrt{n}$, if the RAM is organized into an $n \times 1$ array and improve the test reliability by eliminating the LFSR that is known to have aliasing problems. Another serious problem in embedded memory testing by random test patterns is the problem of memory initialization, which has been tackled here by adding word-line flag registers. The paper has made indepth empirical studies of the functional faults such as stuck-at, coupling, and pattern-sensitive by suitably representing these faults by Markov chains and by simulating these chains to derive various test lengths required for detecting these faults. The simulation results conclusively show that, in order to test a 1M-bit RAM for detecting the common functional faults, the proposed technique needs only one second as opposed to about an hour needed by the conventional random testing where memory cells are tested sequentially.

Keywords: Embedded random-access memories, random testing and testable design.

1. Introduction

As Ultra Large Scale Integrated (ULSI) chips with several billion transistors are becoming realities, powerful computer systems with gargantuan memories, fast processors and complex control circuits for several types of analog and digital I/O data are envisioned to be integrated within a single chip or wafer. Most of the embedded memories in such ULSI/WSI chips will have poorly controllable address, read/write and input lines, and also poorly observable data output lines Several strategies employing built-in self-test (BIST) circuits have been proposed in the past [1], [2], [3], [4], [5] for testing memories embedded within a ULSI/WSI chip. Jain and Stroud [1] proposed a simple scheme for testing embedded memories and their scheme was incorporated in SRAM module generators so that embedded SRAMs could be automatically augmented with the proposed BIST circuit. Mazumder and Patel [6] proposed a parallel testing scheme where in the test mode the decoder allowed the BIST hardware to simultaneously read the values of and write into multiple cells, thereby the cells were verified by comparing within themselves. Sridhar [5] employed a built-in parallel signature analyzer to simultaneously read and write on all the cells in a word-line and he developed

*An abridged version of this article was published in the IEEE International Conference on Wafer-Scale Integration, January 1989. This research was partially supported by the NSF under grant number MIP-9013092 and by ONR under grant number 85-K-0716.

parallel versions of functional test algorithms which could be externally applied to the parallel signature analyzer through serial scan-in line, or could be internally generated by a BIST circuit. BIST circuits in the above designs usually require large silicon overhead, and are also difficult to connect to memory arrays which are frequently distributed all over a chip in the form of register banks or small local memories. In order to tackle this problem, Nadeau-Dostie, Silbert and Agarwal [4] proposed a serial interfacing scheme in which several embedded memories shared the same BIST circuit and therefore considerable amount of routing overhead was saved. The serial interface allowed the BIST circuit to control a single bit of RAM's (or a group of RAMs') input data path. Only one bit of the output data path was made available to the BIST circuit for observation while the test algorithms were executed. The other bits in a wide-word memory organization were controlled and observed indirectly through the serial data path built using the memory itself and a set of multiplexers. In this article, a new strategy, which tests embedded memories by randomly generated test vectors, has been developed and examined as a viable alternative to deterministic built-in self-testing. Because of the presence of widely heterogeneous functional elements within a ULSI/WSI chip, random or deterministic test patterns applied at the external pins can be fairly *randomized*, and frequently then can be directly applied to the embedded arrays without requiring a separate built-in pseudorandom pattern generator. The proposed random testing technique can be easily adapted in embedded memory arrays and is expected to test them more rapidly than those used by the previous researchers [7], [8].

The conventional random testing approaches often employ a pseudorandom generator to excite the address and control lines in a RAM, and the response is stored in a linear feedback shift register (LFSR). After applying a preassigned number of test vectors, the signature in the LFSR is compared to a fault-free signature obtained from a *gold unit* by applying the same set of pseudorandom test vectors. Two potential problems with such conventional approaches can be: (i) an LFSR often introduces a small amount of error in the form of aliasing, and (ii) it is mandatory to initialize the memory cells deterministically so that the basis of comparison of signatures of the memory under test (MUT) with that of the gold unit is valid. Moreover, since, in many conventional random testing schemes, the memory cells are sequentially accessed, an enormous number of test vectors are needed by these testing techniques. It has been shown that in an n-bit RAM all stuck-at faults can be tested by applying about $49n$ random patterns, [7] as opposed to only $4n$ patterns in a Column Bar Test [9].

This article proposes an efficient strategy for testing embedded RAMs by *pure random* (as opposed to preassigned sequences in the conventional pseudorandom) test patterns. The proposed technique eliminates the LFSR, and it employs a parallel memory testing strategy to accelerate testing times by a factor of $0.5\sqrt{n}$. The problem of memory initialization has also been solved. The salient features of this paper are: (i) a new design strategy for testing embedded memories by pure random test patterns, (ii) elimination of memory initialization problem, (iii) a comprehensive study of random patterns required for testing functional faults in a RAM array, (iv) reduction of test complexity by a factor of $\sqrt{n}$, (v) a large fault coverage; for example, $1200n$ test vectors applied to an n-bit RAM can test all functional faults, including pattern-sensitive faults in any arbitrary 5-neighborhood, where individual cells may be located anywhere in the array, and finally (vi) improvement of reliability of random testing by eliminating the LFSR and the storage register. The proposed design can be used as an alternative approach to BIST techniques which use deterministic algorithms. The large number of random test vectors (about 1800 per cell) needed to test the functional faults, like stuck-at, 2-coupling [10], [11] and pattern-sensitive faults [12], [13] are efficiently used to test $0.5\sqrt{n}$ cells in parallel, and thereby the overall time required by the proposed random testing is less than 1 sec for testing a 1M-bit RAM with 50 nsec memory cycle time. For each n-bit memory array, the proposed design uses about an extra $2\sqrt{n}$ transistors in place of the LFSR, comparator and storage register in the existing designs.

The rest of the article has been organized as follows. Section 2 makes an in-depth analysis for examining the fault coverage by random test patterns. Markov chains represent the different functional faults, and the number of test vectors needed to detect these faults are estimated by numerical techniques. In order to detect the pattern-sensitive faults over a neighborhood of five cells, about 1803 million test vectors are required for testing a 1M-bit RAM array. This test time can be reduced by 512 times by employing a new design-for-testability scheme discussed in Section 3. The scheme allows many memory cells to be tested in one memory cycle, and thereby it significantly reduces the test time. Moreover, since the faulty cells are identified by mutually comparing the contents of the memory cells that are always accessed simultaneously for read and write operations, the proposed design does away with the LFSR and signature storage register. Another potential advantage of the proposed design is that the memory is not needed to be initialized in a deterministic manner. The conventional random testing requires that the memory must be initialized before pseudo-random test patterns are applied so that the resultant content of the LFSR can be compared correctly with that of the gold unit. In embedded applications, memory cells cannot be initialized externally.

2. A Comprehensive Study of Fault Coverage

In this section, a comprehensive study is made for estimating the test length coefficient (defined as the number of test vectors needed per memory cell) for 99.9% detection quality. Each type of functional fault is represented by an appropriate Markov chain, and the probability of fault detection has been enumerated by using numerical methods. Three most common types of functional faults are: stuck-at, coupling, and pattern-sensitive [9].

2.1. Markov Model for Stuck-at Fault

Principally there are two types of operations on a memory cell—read and write. A fault-free read operation does not change the state of a cell, while a write operation may or may not change its state. If a write operation on a cell x changes its current state $s(x)$ to $\bar{s}(x)$, then the operation is called a transition write. If $s(x)$ changes from 0 to 1, then the operation is denoted by $\uparrow(x)$. If $s(x)$ changes from 1 to 0, then the operation is denoted by $\downarrow(x)$. Faults in the memory cells are detected by transition write operations. If $\uparrow(x)$ is always faulty irrespective of the content of other cells, then the cell x is said to be stuck-at 0. Similarly, if $\downarrow(x)$ is always fault, the cell x is then stuck-at 1. Figure 1 shows the Markov chain for a cell that is being stuck-at 0. State S_0 represents that a memory cell contains 0, state S_1 represents that a memory cell contains 1, and state S_2 represents a memory cell is read and a stuck-at 0 fault is being detected. Let the probabilities of writing a 0, a 1, and reading a cell be w_0, w_1 and r, respectively. The resulting transition probabilities between the states are shown in figure 1 and the transition matrix is given by

$$M = \begin{bmatrix} 1-w_1 & w_1 & 0 \\ w_0 & 1-w_0-r & r \\ 0 & 0 & 1 \end{bmatrix}$$

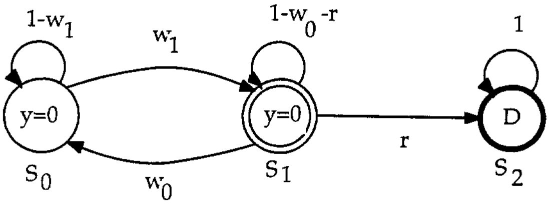


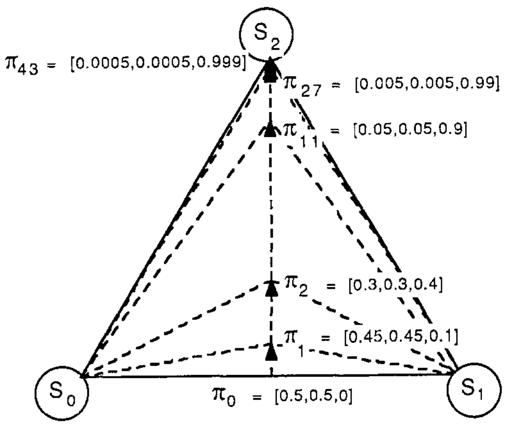
Fig. 1. Markov diagram for stuck-at 0 fault.

Initially, the memory cell is at state S_0 with a probability $I_0(=p_0(S_0))$ or at state S_1 with a probability $1-I_0(=p_0(S_1))$. For a memory cell which is stuck-at 0, the fault is sensitized if a transition write w_1 is made. The cell cannot be initially at state S_2 which is an absorbing state, and, therefore, the probability $p_0(S_2)$ that the cell is in state S_2 before any test vector is applied, is 0. In figure 1 and subsequently in all Markov chain diagrams, the initial states are shown by light circles, the states in which faults are sensitized are shown by double circle, and finally the detecting state is shown by a heavy circle. The detecting state is an absorbing state, and it is reached by a read operation from one of the sensitized states. The probability $p_L(S_2)$ that the fault is detected after L test vectors are applied, gives the confidence level and is known as *quality of detection* (denoted by Q_D). The probability that even after L test vectors are applied, the fault is not detected is known as *escape probability*, $e = 1-Q_D = p_L(s_0) + p_L(S_1)$, where $p_L(S_0)$ and $p_L(S_1)$ are the probabilities that after L test vectors are applied, the cell is in state S_0 and S_1, respectively. These probabilities can be computed from the following set of equations:

$$\begin{bmatrix} p_L(S_0) \\ p_L(S_1) \\ p_L(S_2) \end{bmatrix} = \begin{bmatrix} 1-w_1 & w_0 & 0 \\ w_1 & 1-w_0-r & 0 \\ 0 & r & 1 \end{bmatrix} \begin{bmatrix} p_{L-1}(S_0) \\ p_{L-1}(S_1) \\ p_{L-1}(S_2) \end{bmatrix}$$

$$= \begin{bmatrix} 1-w_1 & w_0 & 0 \\ w_1 & 1-w_0-r & 0 \\ 0 & r & 1 \end{bmatrix}^L \begin{bmatrix} x \\ 1-x \\ 0 \end{bmatrix}$$

The above equations represent linear recurrences and can be solved mathematically. In figure 2, the state probability diagram of the stuck-at 0 fault is shown, where the triplet $\pi_L = [p_L(S_0), P_L(S_1, p_L(S_2]$ denotes the state probability after L-th test vectors are applied to the initial condition (i.e., iterations are made). It may be noted that at the 47-th iteration the probability of detection, $p(S_2)$, increases to 0.999 from the initial value of 0 at either state S_0 or S_1. As it can be seen in subsequent fault models, the analytical solutions for these Markov chains with an absorbing state becomes quite cumbersome. Numerical techniques have, therefore, been applied in this article by running statistical packages on a Vax 11/780. Table 1 shows how the quality of detection (Q_D) improves with the average number of applied test vectors per cell, L/n, which is commonly known as *test length coefficient*. Clearly, this value depends on the initialization of memory cells, and is shown in table 1 for the cell being initialized both deterministically with unity probability to 0 (1) and randomly (i.e., initially it is in state S_0 or S_1 with probability 0.5). It can be seen that the probability of fault detection, Q_D is a monotonically increasing function of test-length coefficient, and depends on the initial content of the cell. In figure 3, the quality of detection is plotted for various test-length coefficient, and it can be seen that on the average about 47 test vectors are required to detect the stuck-at faults for a quality of detection of 99.9%. Also plotted are the number of cells (samples) vs. the test length required to detect the fault. It can be seen that for a large number of samples the fault is detected by applying less than 10 vectors. In this analysis, it is assumed that all the cells in the memory have uniform access probability, $p_a = 1/n$, where n is the number of cells. But, in practice, due to the presence of combinational logic circuits very frequently, this access probability may not be uniform. If an address line is selected by a k-input AND or NOR gate, then the address line will contain 1 with a probability of $1/2^k$, and it will contain 0 with probability of $1-(1/2^k)$. These address line probabilities will be exactly opposite if the line is selected by a k-input OR or NAND gate. Thus, the presence of combinational logic on the path of address lines considerably modifies the access probability. Figure 4 illustrates how the test length coefficient changes with the signal probability of address and data lines, where the signal probability



S_2
π_{43} = [0.0005,0.0005,0.999]
π_{27} = [0.005,0.005,0.99]
π_{11} = [0.05,0.05,0.9]
π_2 = [0.3,0.3,0.4]
π_1 = [0.45,0.45,0.1]
S_0
π_0 = [0.5,0.5,0]
S_1

Fig. 2. State probability diagram for a stuck-at 0 fault.

Table 1. Test length coefficient for stuck-at faults.

	$I_0 = 0$		$I_0 = 1$		$I_0 = 0.5$	
Q_D	*s-a-*0	*s-a-*1	*s-a-*0	*s-a-*1	*s-a-*0	*s-a-*1
0.1	2	1	1	2	1	1
0.2	3	1	1	3	1	1
0.3	4	1	1	4	2	2
0.4	5	2	2	5	3	3
0.5	7	2	2	7	4	4
0.6	8	3	3	8	6	6
0.7	10	4	4	10	8	8
0.8	13	7	7	13	10	10
0.9	18	11	11	18	15	15
0.99	33	27	27	33	31	31
0.999	49	43	43	49	47	47
0.9999	65	59	59	65	63	63
0.99999	81	75	75	81	79	79

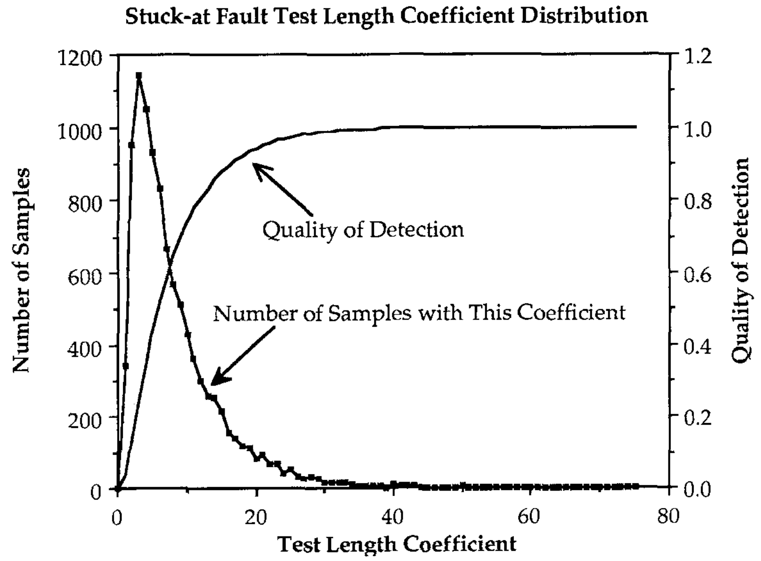


Fig. 3. Quality of detection vs. test length coefficient.

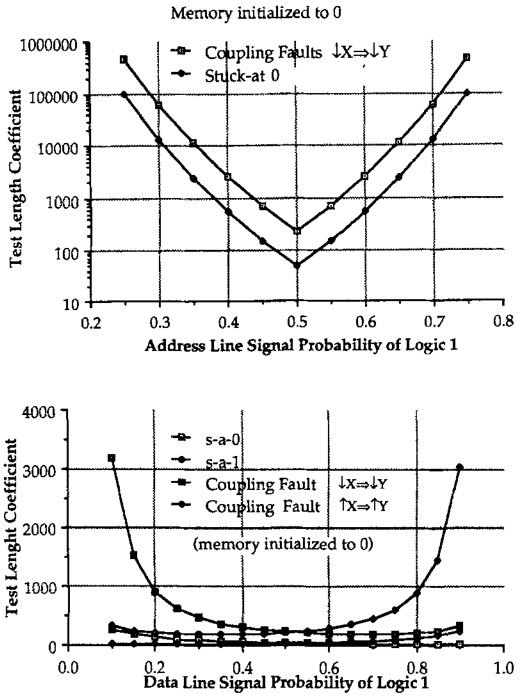


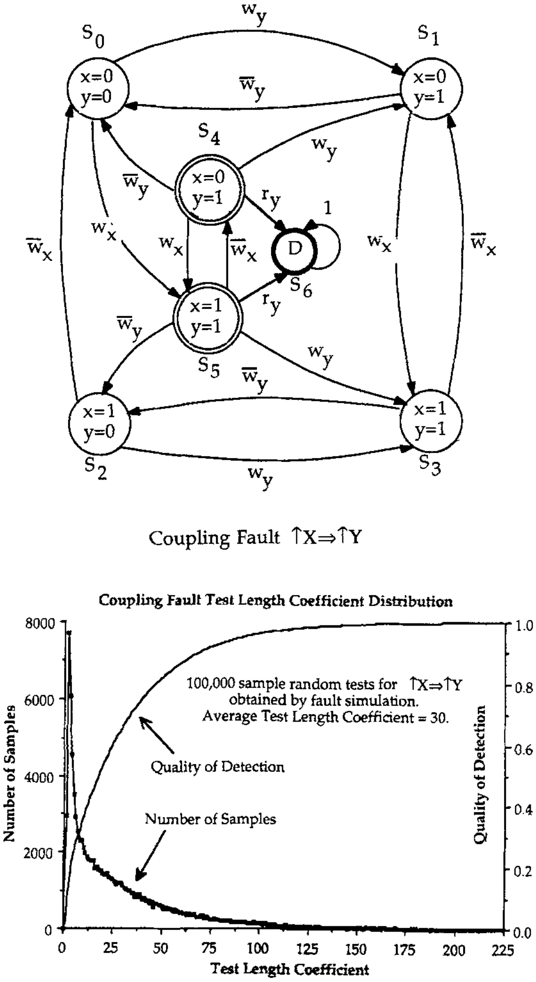
Fig. 4. Signal probability vs. test length coefficient.

has been varied over a wide range (from 0.1 to 0.9 in the case of the data line and from 0.25 and 0.75 in the case of address lines—beyond which the test length tends to increase rapidly).

2.2. *Markov Model for Coupling Fault*

Two cells, say x and y, are said to be 2-coupled if a transition write on x changes the state of y, whenever y is at a preassigned state, $\sigma_y \in \{0,1\}$. Such a fault is denoted by the doublet $<\Psi(x), \sigma_y>$, where Ψ is a transition write operation. Principally, there are four variants of this doublet, namely, $<\uparrow,0>$, $<\uparrow,1>$, $<\downarrow,0>$, and $<\downarrow,1>$. If only one of the above 2-coupling faults exist between two arbitrary cells x and y, they are said to be *1-way coupled*. The cells are said to be *2-way coupled*, if a combination of two or more of the above faults coexist. These faults can be denoted by $<\uparrow,X>$, $<\downarrow,X>$, $<\uparrow\downarrow,0>$,$<\uparrow\downarrow,1>$, and $<\uparrow\downarrow,X>$, where X is a don't care state.

For each type of coupling fault described above, a separate Markov chain can be represented. In this section only one type of 1-way coupling fault is represented corresponding to $<\uparrow,0>$, i.e., when a transition write $\uparrow$ is made on a coupling cell x, the coupled cell y changes state only if it is in state 0. Figure 5 represents the Markov chain for the corresponding coupling fault.



Coupling Fault ↑X⇒↑Y

Fig. 5. Quality of detection vs. test length coefficient.

States S_0, S_1, S_2 and S_3 represent the four possible initial states of the cells x and y, and are $<00>$, $<01>$, $<10>$ and $<11>$, where the doublet $<\sigma_x\sigma_y>$ denotes the content of cells x and y. Clearly in these four states the fault $<\uparrow(x), \sigma_y=0>$ is not sensitized. States S_4 and S_5 represent the fault being sensitized and correspond to $<01>$, $<11>$, respectively. In any of these two states if the cell y is read, then the fault will be detected, and the corresponding absorbing state is given by S_6. Markov transition matrix is constructed from figure 5 and using numerical technique, the quality of detection is enumerated (in figure 6) as a function of length coefficient. For a detection quality of 99.9%, on the average 220 (228) test lengths are required if the memory is deterministically initialized to 0 (1), and about 225 test length coefficient is required if the memory is randomly initialized. For other types of two-cells coupling faults, the Markov chain analysis has been made, and the length coefficients for different

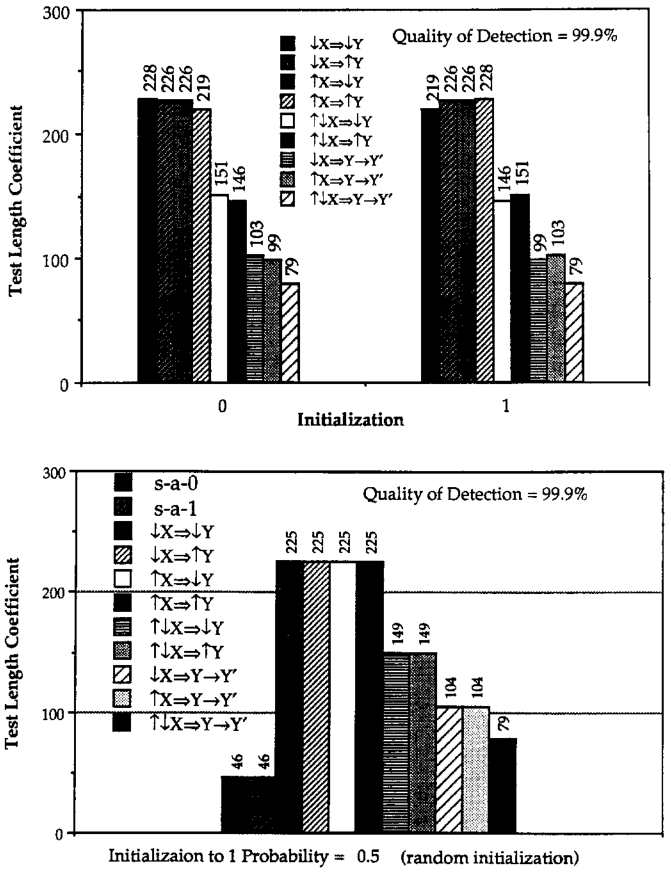


Fig. 6. Quality of detection vs. test length coefficient.

faults are shown in figure 6 and table 2. In embedded DRAMs with destructive read operation, the cell which is read is rewritten with the original data during restoration phase. Thus, the read operation is potentially vulnerable in the sense like the transition write operations, the read operation may introduce coupling faults. In figure 7, the Markov chain for the coupling fault $\uparrow x = > \uparrow y$ with destructive read operation is shown. It may be noted in the initial state if $\sigma_y = 0$ and the cell x is read, the coupling fault might occur, which is shown by heavy arc labeled r_x. To test DRAMs with destructive read operation, only about 87 test vectors are needed to detect the coupling faults with 99.9% confidence. In Appendix 1, the Markov chains for the different coupling faults are shown.

2.3. *Markov Model for Pattern-Sensitive Faults*

A pattern-sensitive fault occurs in a neighborhood of 3 or more cells. A pattern-sensitivity is called static (SPSF), if a transition write cannot be made in the presence of specific patterns in the neighborhood. Like the coupling faults, an SPSF is denoted by a doublet $<\Psi S>$, where S is the specific patterns in which the transition write operation Ψ on the base cell is faulty. A pattern-sensitivity is called dynamic (DPSF), if a transition write operation Ψ on a cell in the neighborhood changes the state of the base cell, provided it is in the preassigned state. A DPSF is denoted by a triplet $<\sigma_x, \Psi(y), S>$, which indicates that if the base cell x, is in state σ_x and a transition write Ψ is made on cell y in the neighborhood, then pattern-sensitive fault occurs if and only if the other cells contain specific pattern S.

Both the SPSFs and DPSFs can be analyzed using Markov chains. Figure 8 represents an SPSF, $<\uparrow, S>$, where the base cell x cannot make an operation $\uparrow$ in the presence of a specific set of patterns S in the neighborhood. Let the doublet $<y,S>$ denote the different states in the Markov model for the fault. Let $S = 1$ denote the presence of specific patterns in the neighborhood and $S = 0$ denote the absence of specific patterns. Initially, the cells may be in one of the four

Table 2. Test length coefficient for coupling faults.

$I_0 = 0$

Q_D	$\downarrow X \Rightarrow \downarrow Y$	$\downarrow X \Rightarrow \uparrow Y$	$\uparrow X \Rightarrow \downarrow Y$	$\uparrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow \downarrow Y$	$\uparrow\downarrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\uparrow X \Rightarrow Y \rightarrow \neg Y$
0.1	8	6	6	2	5	2	2	5	2
0.2	12	10	10	4	8	4	4	7	4
0.3	16	14	14	8	11	7	5	9	5
0.4	21	19	19	12	14	10	7	11	7
0.5	27	25	25	18	18	14	9	14	10
0.6	34	32	32	25	23	18	12	17	13
0.7	44	41	41	35	29	25	15	22	17
0.8	57	55	55	48	38	33	19	28	23
0.9	79	77	77	70	53	48	27	38	33
0.99	154	152	152	145	102	98	53	72	67
0.999	228	226	226	220	151	147	79	106	102

$I_0 = 1$

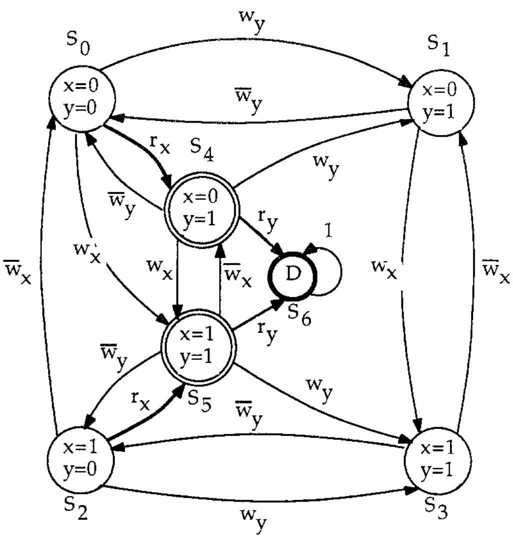
Q_D	$\downarrow X \Rightarrow \downarrow Y$	$\downarrow X \Rightarrow \uparrow Y$	$\uparrow X \Rightarrow \downarrow Y$	$\uparrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow \downarrow Y$	$\uparrow\downarrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\uparrow X \Rightarrow Y \rightarrow \neg Y$
0.1	2	6	6	8	2	5	2	2	5
0.2	4	10	10	12	4	8	4	4	7
0.3	8	14	14	16	7	11	5	5	9
0.4	12	19	19	21	10	14	7	7	11
0.5	18	25	25	27	14	18	9	9	14
0.6	25	32	32	34	18	23	12	13	17
0.7	35	41	41	44	25	29	15	17	22
0.8	48	55	55	57	33	38	19	23	28
0.9	70	77	77	79	48	53	27	33	38
0.99	145	152	152	154	98	102	53	67	72
0.999	220	226	226	228	147	151	79	102	106

$I_0 = 0.5$

Q_D	$\downarrow X \Rightarrow \downarrow Y$	$\downarrow X \Rightarrow \uparrow Y$	$\uparrow X \Rightarrow \downarrow Y$	$\uparrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow \downarrow Y$	$\uparrow\downarrow X \Rightarrow \uparrow Y$	$\uparrow\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\downarrow X \Rightarrow Y \rightarrow \neg Y$	$\uparrow X \Rightarrow Y \rightarrow \neg Y$
0.1	5	5	5	5	3	3	2	3	3
0.2	9	9	9	9	6	6	4	4	5
0.3	13	13	13	13	9	9	5	7	7
0.4	18	18	18	18	12	12	7	9	9
0.5	24	24	24	24	16	16	9	12	12
0.6	31	31	31	31	21	21	12	15	15
0.7	40	40	40	40	27	27	15	20	20
0.8	54	54	54	54	36	36	19	26	26
0.9	76	76	76	76	50	50	27	36	36
0.99	151	151	151	151	100	100	53	70	70
0.999	225	225	225	225	149	149	79	104	104

states S_0, S_1, S_2 and S_3 corresponding to the doublets $<00>$, $<01>$, $<10>$ and $<11>$, respectively. States S_4 and S_5 denote the fault is being sensitized, when y changes from 0 to 1 in presence of $S = 1$. State S_6 detects the fault and is the absorbing state. Similar to the other techniques, we computed the various length coefficients for different qualities of detection and for different size of neighborhood. This is shown in figure 9. Table 3 shows the test length coefficient for SPSFs with different neighborhood size.

Figure 10 represents a DPSF, $<\uparrow(y),x=0,S>$, where the content of the base cell x changes from 0 to 1, if the transition write $\uparrow$ is made on cell y in presence of the specific patterns S in the neighborhood. It may be noted that the fault does not occur if the pattern in the neighborhood is $\bar{S}$, i.e., other patterns than S. The states in figure 10 are represented by the triplet $<y,x, S>$, $S = 1$ indicates the presence of the pattern represented by the state S. Obviously, there are 2^3 = 8 initial states in figure 10 in which the fault is not

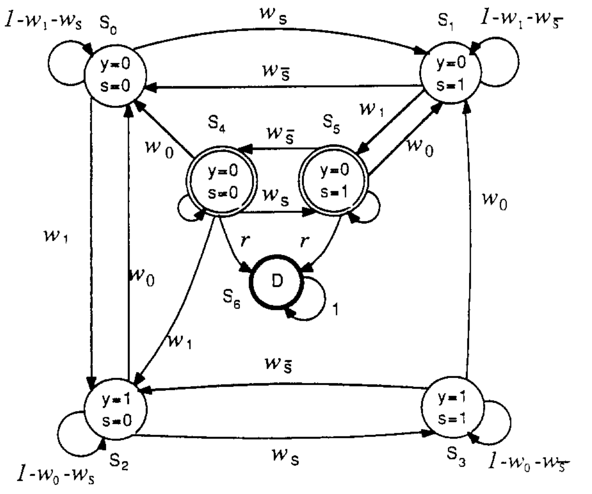


Coupling Fault↑X⇒↑Y with destructive Read

Test Length Coefficient for Destructive Read Coupling Fault ↑X⇒↑Y

Q_D	I_Q=0	I_Q=0.5	I_Q=1
0.1	1	2	4
0.2	2	3	5
0.3	3	5	7
0.4	4	7	9
0.5	6	9	11
0.6	8	12	14
0.7	12	15	17
0.8	17	20	22
0.9	25	29	31
0.99	53	56	59
0.999	81	84	87

Fig. 7. Coupling fault in DRAM with destructive read.



Markov Chain for SPSF < ↑,S>

Fig. 8. Static pattern-sensitive fault (SPSF).

sensitized. There are four states indicated by double circles which sensitize the faults, and it can be seen that only two of these states can be entered from the initial states when $S = 1$. After the fault is sensitized, the fault is detected in the absorbing state represented by dark circle when the base cell is read. Figure 11 shows the quality of detection as a function of test length coefficients. Table 4 shows the test length coefficient for DPSFs with different neighborhood size.

3. A New Testable RAM Design

From the above analysis it can be seen that, for accomplishing a detection quality of 99.9%, about 1800 million random patterns must be applied to test all functional faults in a 1 M-bit memory chip organized into a 1K × 1K array. Moreover, as discussed in earlier section, the conventional approaches using LFSR and comparator suffer from several limitations, such as aliasing, relatively large area overhead and initialization problems. In this section, we demonstrate how these two potential problems associated with the conventional random testing can be circumvented by introducing a new design-for-testability approach. In the proposed architecture, the memory subarrays are reorganized as shown in figure 12. An $n/d \times d$ memory is organized into d subarrays of $n/d \times 1$ size. Each subarray utilizes the organization of parallel testing discussed in [14], [15]. By utilizing this testable design, not only is the overall test length reduced by a factor of $\sqrt{n/d}$, but also are the LFSR and storage register eliminated in random testing.

In order to access multiple cells, the bit-line decoder has been modified such that all the even lines and odd lines are accessed simultaneously. This is achieved by adding two extra transistors Q_8 and Q_9 (in figure 13) to the decoder which will select the odd/even bit lines together through SELECT independent of the address bit pattern. If the write operation is selected, all the even (odd) bit cells in a word line are written simultaneously 0 or 1 based on the data in Data-In buffer. During a read operation, the contents of these cells are compared together by a parallel comparator which detects an error if all the even (odd) cells in a word line do not match. A mismatch is indicated in the error latch by setting the ERROR line high.

Figure 14 shows the parallel comparator and error detector. The p-channel transistors $T_1, \ldots, T_{m-1}$ are connected in parallel and detect a concurrent occurrence of 1's in bit-lines. The n-channel transistors P_1,

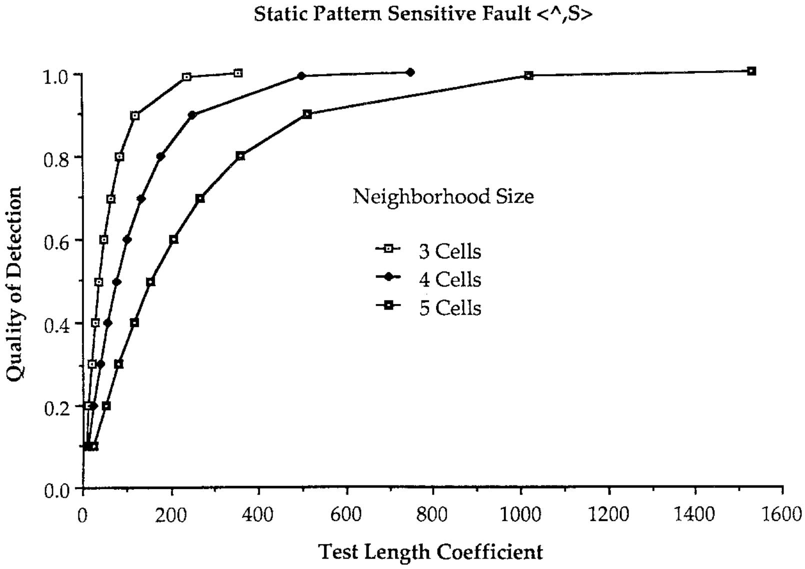


Fig. 9. Quality of detection vs. test length coefficient.

Table 3. Test length coefficient for static pattern fault < ↑, *S* >.

$l_0 = 0.5$	Neighborhood Size		
Q_D	3 Cells	4 Cells	5 Cells
0.1	7	13	25
0.2	13	26	51
0.3	20	40	81
0.4	28	57	115
0.5	37	77	155
0.6	49	101	204
0.7	63	132	268
0.8	84	176	358
0.9	120	250	511
0.99	239	499	1021
0.999	356	748	1531

Table 4. Test length coefficient for dynamic pattern fault < ↑, *S* >.

$l_0 = 0.5$	Neighborhood Size		
Q_D	3 Cells	4 Cells	5 Cells
0.1	6	12	22
0.2	14	27	53
0.3	23	44	88
0.4	33	65	129
0.5	45	88	176
0.6	59	118	235
0.7	78	155	310
0.8	104	208	416
0.9	149	299	598
0.99	299	600	1200
0.999	449	901	1803

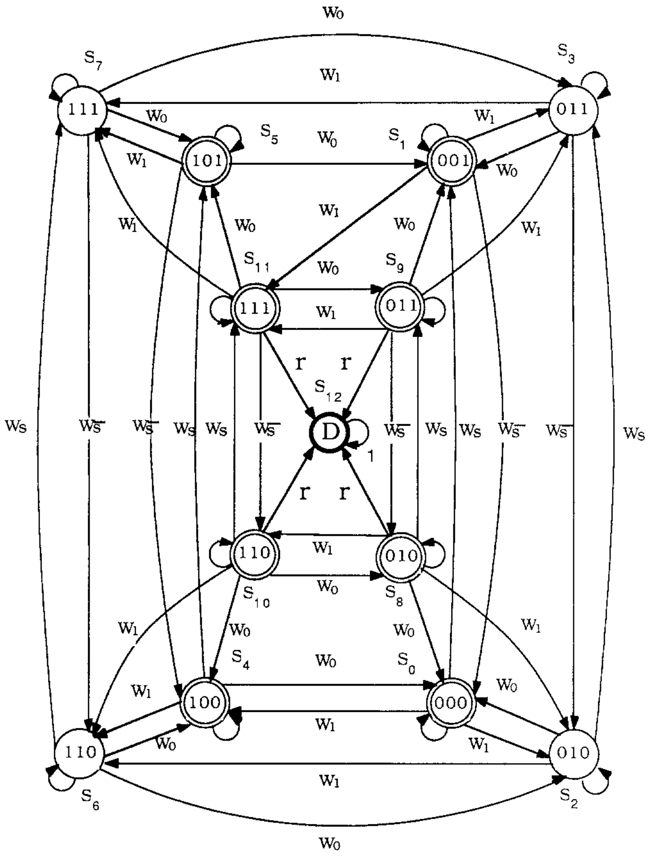


Fig. 10. Markov chain for DPSF.

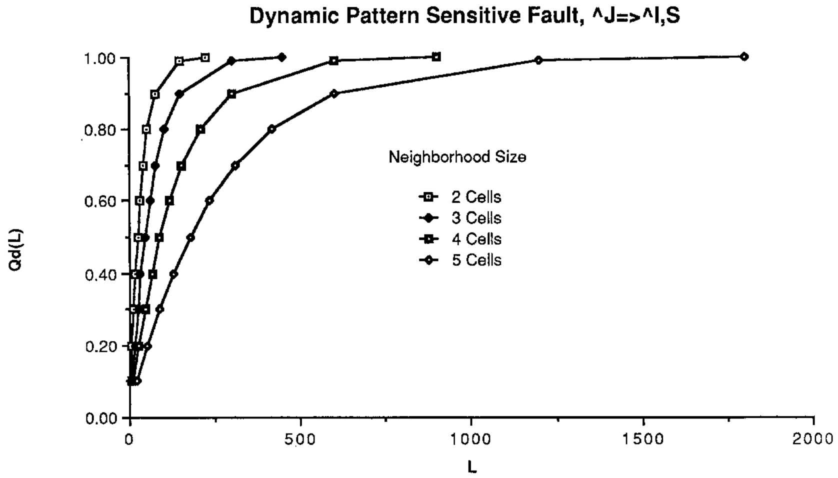


Fig. 11. Quality of detection vs. test length coefficient.

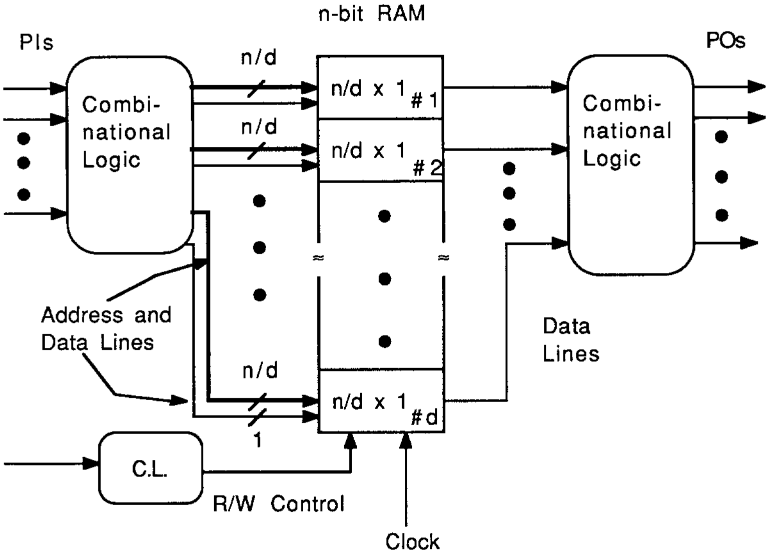


Fig. 12. Proposed organization of embedded memory.

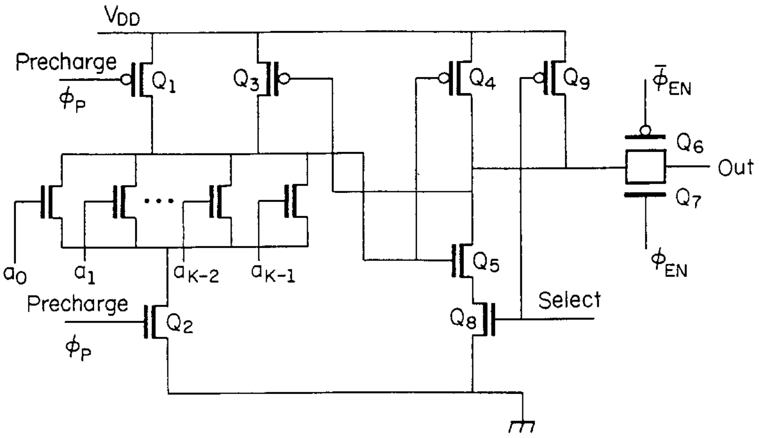


Fig. 13. Modified decoder circuit.

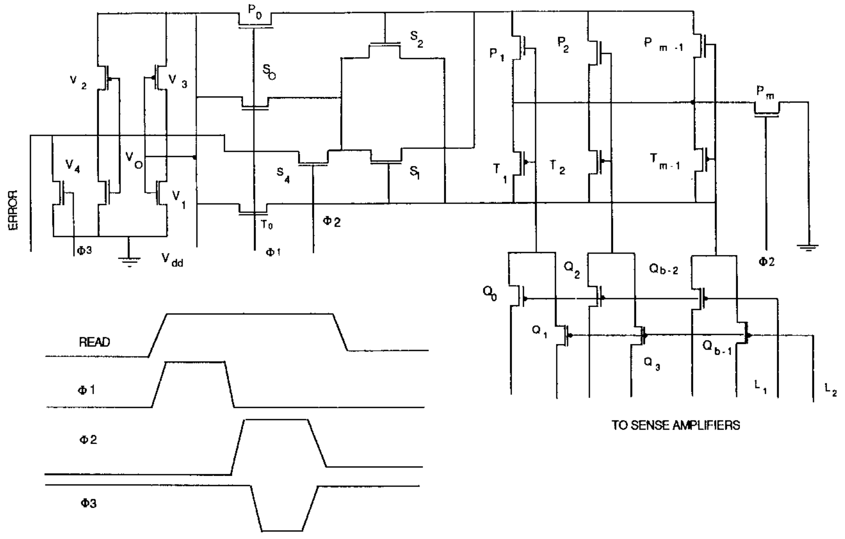


Fig. 14. Parallel comparator with error detector.

..., P_{m-1} are also connected in parallel and detect a concurrent occurrence of 0's in bit-lines. Transistors T_0 and P_0 are precharge transistors while the transistor P_m is a discharge transistor which remains cut off during the precharge phase and turns on during the discharge phase, $\phi 2$. Since bit-lines are divided into two groups (odd and even) pass transistors are incorporated to compare the contents of either all odd or all even bit-lines simultaneously. Signals L_1 and L_2 select these even and odd bit-lines, respectively. Transistors S_0, S_1 and S_2 form a coincidence detector. If all the selected bit-lines are 0's or 1's, then either S_1 or S_2 conducts and the output of the detector is 0. The output of the Ex-Or gate is connected to an error latch consisting of transistors V_0, ..., V_3. The error latch output is ERROR=0, when the selected bit lines are identical. If the bit lines are not identical, then both S_1 and S_2 remain cutoff and the detector output is 1. This triggers the error latch setting its output to ERROR=1. During the write phase and normal mode of operation, the error latch is clamped to zero by V_4. The error detector is inhibited by the discharge transistor P_m during the start of the read phase when the sense amplifiers outputs are not identical because of sluggish changes in some of the sense amplifiers.

The conventional RAM testing algorithms can be easily applied to this new architecture and a test speedup of 1000 can be achieved for a 4M-bit RAM. New parallel algorithms for testing pattern-sensitive faults have been proposed in [6], and built-in self test circuit has been designed for the proposed test algorithms. The proposed BIST design does not require too much extra circuitry and does not degrade the memory cycle time in normal applications. It can be applied in embedded applications where the memory is organized into a single large array inside the ULSI/WSI chip. The intent of this article is to demonstrate how this testable design can be applied for random testing in embedded applications where the memory is distributed in multiple arrays inside a ULSI/WSI chip. Because of the routing complexity associated with conventional self-testing techniques, random pattern testing is likely to be more desirable in many applications where embedded registers and small amount of SRAMs will be scattered all over a chip.

The shortcomings of the conventional random testing can be circumvented if the proposed testable architecture is employed for random testing. Since the cells written together are compared simultaneously, the problem of observability will not exist and no LFSR will be needed nor will any predetermined pseudo-random sequences be necessary to test the memory. In order to ensure that the technique does not need any deterministic initialization, the word-line decoder is added with a latch which is set whenever a write operation is done utilizing that word line. While reading the word line, the result of comparison activates the Error Detector if the latch is set; otherwise, the Error Detector is disabled, and thereby incorrect comparisons due to initial arbitrary data patterns in an uninitialized memory plane are not allowed to maliciously corrupt the testing scheme. The modified circuit with write enable latches for word-lines is shown in figure 15. For each word-line, there are two latches, L_j^E and L_j^O as shown in

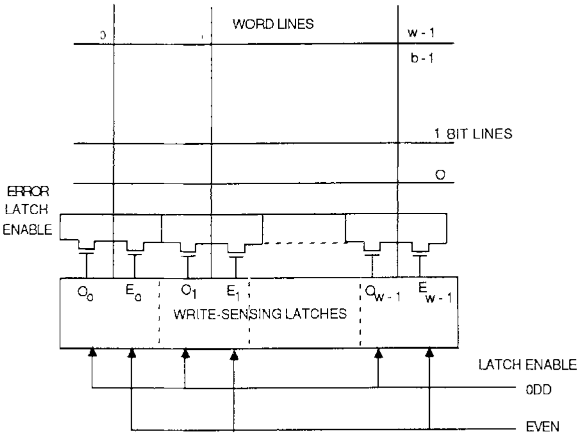


Fig. 15. Modified circuit with write enable latch.

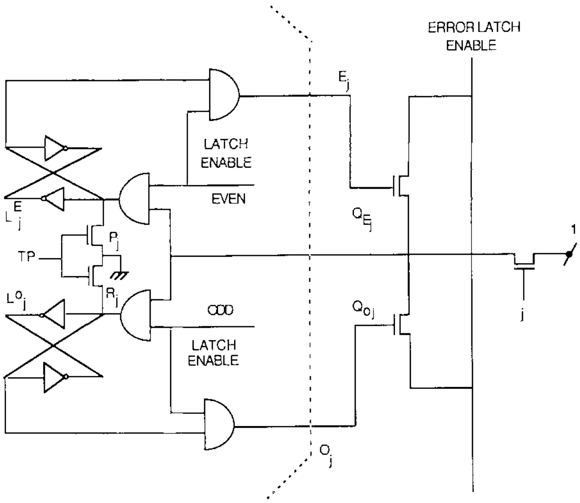


Fig. 16. Design of write-sensing latch.

figure 16. Each latch consists of back-to-back connected inverters. The LATCH ENABLE signal ODD is held high only in the write mode, if odd bit-lines are selected. Similarly, the EVEN is held high in the write mode if even bit-lines are selected. The word line j goes high, if it is selected by the word-line decoder; otherwise, it remains low. Thus the Write Sensing Latch, L_j^E is set to 1 and the pass transistor Q_j^E turns on, holding the ERROR LATCH ENABLE to high, if both the word-line j and the even bit-lines are selected during a write operation. During a subsequent read operation, the latch L_j^E will be set to high, and the pass transistor will be selected if the EVEN bit lines and word line j are selected. The overall random testing technique can be described by the following sequence of steps:

1. Set the TEST mode to 1. While all the test patterns are not applied, repeat steps 2 through 6; else exit successfully.
2. Select all the even bit lines or all the odd bit lines simultaneously.
3. Select a write word line arbitrarily.
4. Select 0 or 1 arbitrarily on the read/write line.
5. If the selected operation is write, set the flag on the selected word line in Write Sensing Latch. If the selected bit lines are odd, set the flag corresponding to odd-half; otherwise, set the flag for even-half in the selected word line.
6. If the selected operation is read, check if the corresponding latch flag is high. If high, the Error Detector is enabled and reports an Error if the parallel comparator finds a mismatch. If the flag is low, Error Detector is disabled, and no mismatch is found.

Table 5 compares the proposed random testing with the conventional random testing schemes where an LFSR is used and the cells are tested in a sequential fashion.

Table 5. Proposed technique vs. conventional random testing.

Criterion	Proposed Technique	Sequential Random Test
Test Time	$O(\sqrt{n})$	$O(n)$
Input Sequence	Pure Random	Pseudo-Random
Overhead	Multibit 0/1 Detector	LFSR, Register, Comparator
Memory Initialization	Not Needed	Mandatory
Comparison Basis	Self	Gold Unit
Reliability	Very Good	Good

4. Conclusion

This article discusses an efficient technique for testing embedded RAMs by using randomly generated test patterns. Unlike the conventional sequential approaches that use LFSRs to generate signatures and multi-bit registers to store fault-free signatures, the proposed technique uses a testable design which consists of a multi-bit 0/1 comparator and some write-sensing latches. The random testing is accomplished by simultaneously

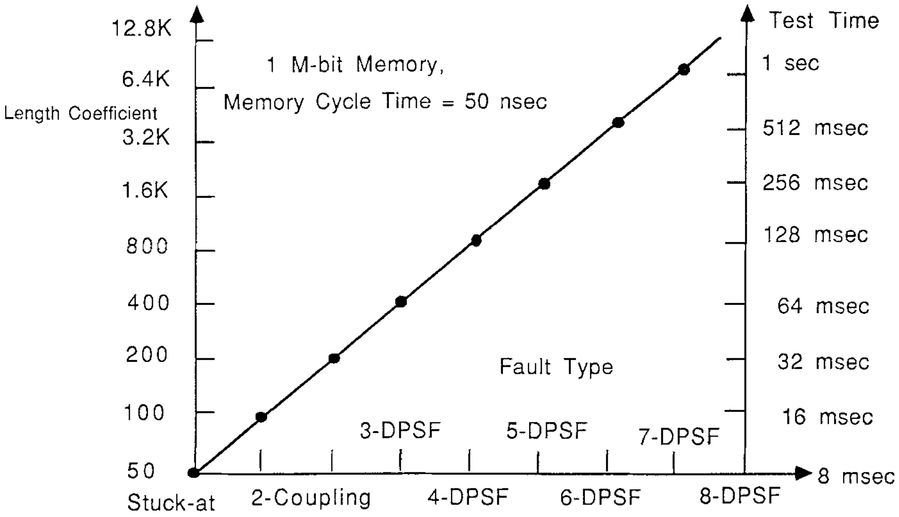
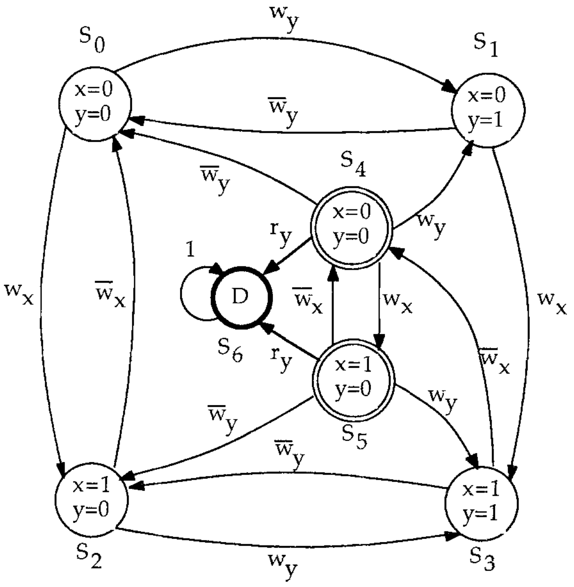


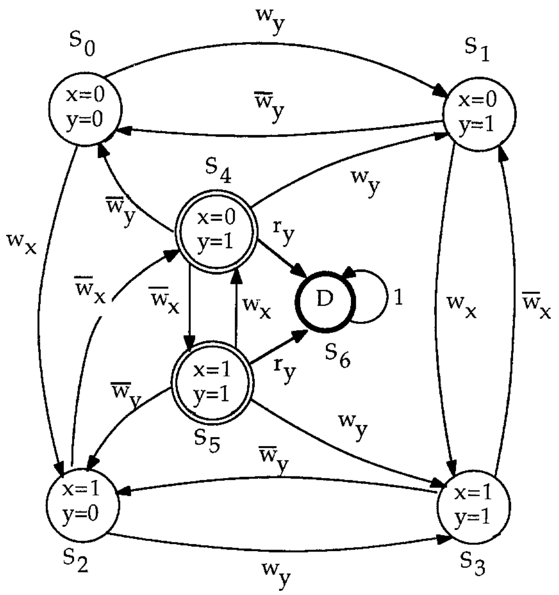
Fig. 17. Fault type and their test length.

accessing multiple cells in a word-line during a single memory cycle, and thereby the overall time to test the embedded RAM is considerably reduced in comparison to many conventional approaches that use sequential cell testing. The article has made detailed empirical studies of the well-known functional faults, namely stuck-at, coupling and patterns-sensitive, using the Markov chain model for various types of faults. From the analysis, it is observed that, in order to test a 16M-bit (1 M-bit) RAM for these commonly occurring functional faults, the proposed technique needs only 4 (1) seconds as opposed to about 16 (4) hours in the conventional random testing where memory cells are tested sequentially (see figure 17). The proposed testable design uses about $2\sqrt{n}$ extra transistors to build a multi-bit 0/1 detector, and evidently has low silicon overhead. The modified memory architecture can also be used for external testing and built-in self-testing by deterministic test algorithms [6], [16].

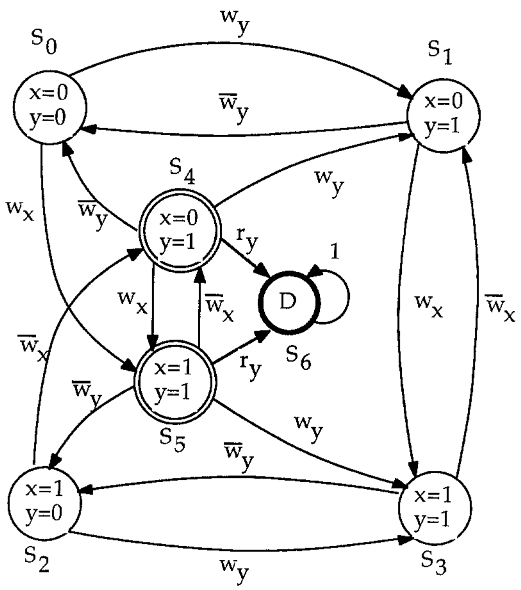
Appendix 1: Markov Chains for Coupling Faults. (See Section 2.2)



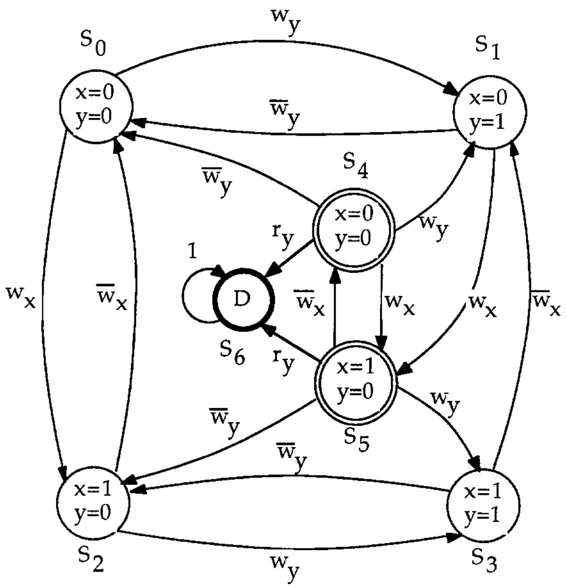
Coupling Fault ↓x⇒↓y



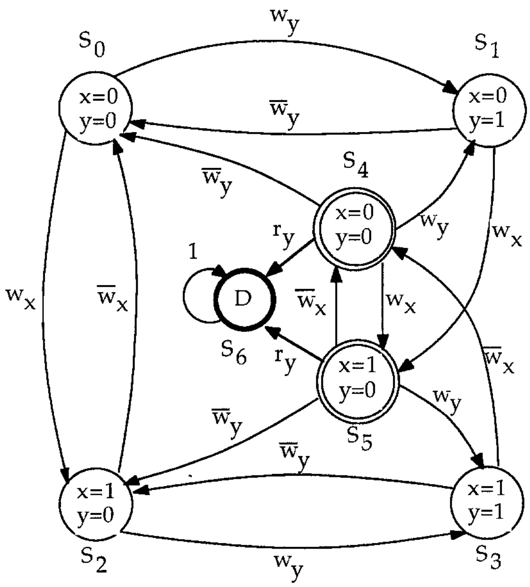
Coupling Fault ↓X⇒↑Y



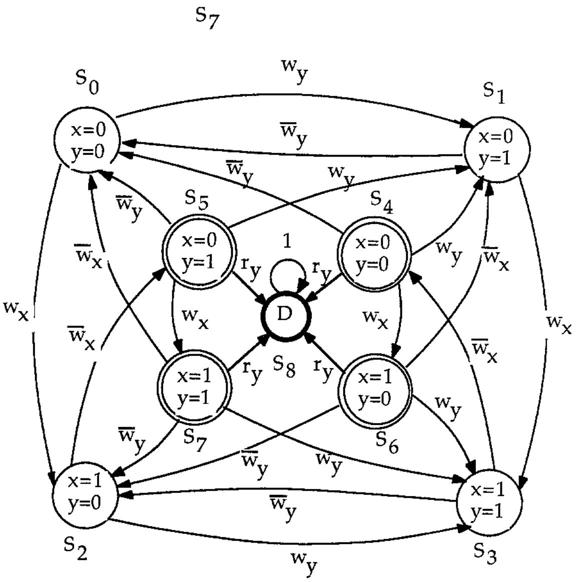
Coupling Fault ↑↓X⇒↑Y



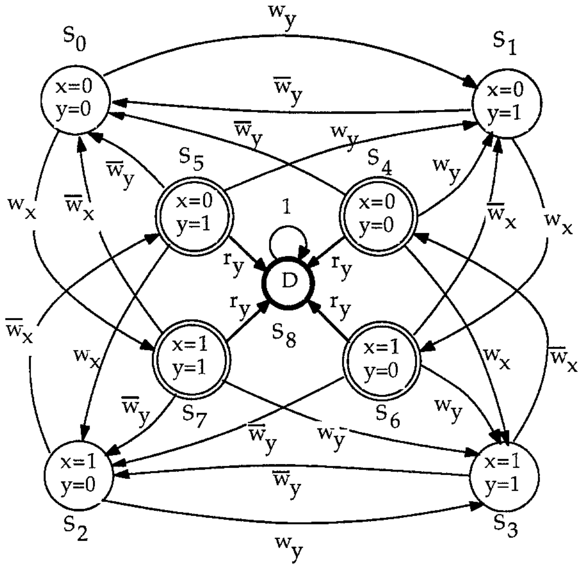
Coupling Fault ↑X⇒↓Y



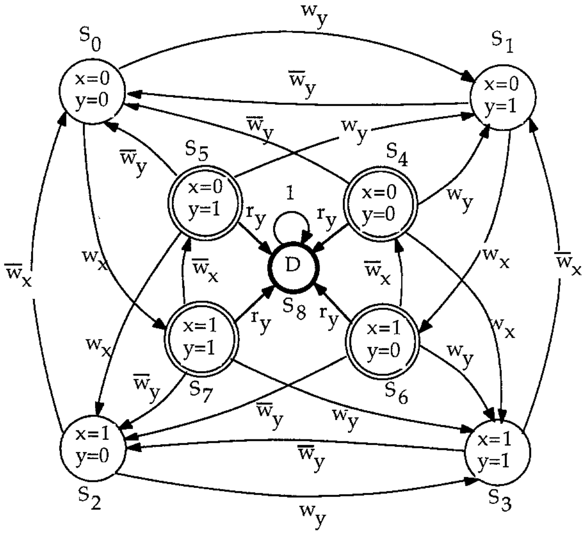
Coupling Fault ↑↓X⇒↓Y



Coupling Fault $\downarrow X \Rightarrow Y \rightarrow \overline{Y}$



Coupling Fault $\uparrow\downarrow X \Rightarrow Y \rightarrow \overline{Y}$



Coupling Fault $\uparrow X \Rightarrow Y \rightarrow \overline{Y}$

References

1. S. Jain and C. Stroud, "Built-in self testing of embedded memories," *IEEE Design and Test of Comp.*, pp. 27–37, October 1986.
2. M. Nicolaidis, "An efficient built-in self test scheme for functional test of embedded RAMs," *Proc. of Int. Test Conf.*, pp. 118–123, 1985.
3. P. Mazunder and J.H. Patel, "An efficient built-in self-testing of random-access memory," *IEEE Trans. on Industrial Electronics*, May 1989.
4. B. Nadeau-Dostie, A. Silburt, and V.K. Agarwal, "Serial interfacing for embedded-memory testing," *IEEE Design and Test of Comp.*, pp. 52–63, April 1990.
5. T. Sridhar, "A new parallel test approach for large memories," *Proc. of Int. Test Conf.*, pp. 462–470, 1985.
6. P. Mazunder and J.H. Patel, "An efficient built-in self-testing of random access memory," *Proc. of Int. Test Conf.*, September 1987.
7. V.P. Gupta, et al., "Random testing for stuck-at storage cells in an embedded memory," *Proc. of Int. Test Conf.*, pp. 157–166, 1984.

8. A. Fuentes, et al., "Random testing versus deterministic testing of RAMs," *Proc. of 16th Fault-Tolerant Comp. Symp.*, pp. 266–271, July 1986.
9. M.A. Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems*, Woodland Hills, Los Angeles, 1976.
10. R. Nair, S.M. Thatte, and J. Abraham, "Efficient algorithms for testing semiconductor random-access memories," *IEEE Trans. on Comp.*, vol. C-27, pp. 572–576, June 1978.
11. D.S. Suk and S.M. Reddy, "A march test for functional faults in semiconductor random-access memories," *IEEE Trans. on Comp.*, vol. C-30, pp. 982-985, December 1981.
12. J.P. Hayes, "Testing memories for single-cell pattern-sensitive faults," *IEEE Trans. on Comp.*, vol. C-29, pp. 249–254, March 1980.
13. K. Kinoshita and K.K. Saluja, "Built-in testing of memory using on-chip compact testing scheme," *Proc. of Int. Test Conf.*, pp. 271–281, 1984.
14. P. Mazumder and J.H. Patel, "Parallel testing for pattern-sensitive faults in random-access memory," *IEEE Trans. on Comp.*, vol. 38(3), pp. 394–404, March 1989.
15. P. Mazunder, J.H. Patel, and W.K. Fuchs, "Design and algorithms for parallel testing of random-access and content-addressable memories," *Design Automation Conf.*, vol. 24, pp. 688-694, July 1987.
16. P. Mazunder, "Parallel algorithms for parametric faults testing in three-dimensional DRAM," *IEEE J. of Solid-State Circuits*, vol. 23, pp. 933–941, August 1988.

Pinaki Mazumder received a B.S.E.E. degree from the Indian Institute of Science in 1976, and M.Sc. degree in Computer Science from the University of Alberta, Canada, in 1985, and a Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 1987. Presently, he is working as an Associate Professor at the Department of Electrical Engineering and Computer Science of the University of Michigan at Ann Arbor. Prior to this, he worked two years as a research assistant at the Coordinated Science Laboratory, University of Illinois, and over six years in India at Bharat Electronics Ltd. (a collaborator of RCA), where he developed several types of analog and digital integrated circuits for consumer electronics products. During the summers of 1985 and 1986, he worked as a Member of Technical Staff in the Naperville branch of AT&T Bell Laboratories. He is a recipient of Digital's Incentives for Excellence Award, National Science Foundation Research Initiation Award, and Bell Northern Research Laboratory Faculty Award. His research interest includes VLSI testing, physical design automation, ultrafast digital circuit design, and neural hardware. He has published over 50 papers in IEEE and ACM archival journals and refereed international conference proceedings. He has coauthored a book on memory testing. He is a Guest Editor of the IEEE Design and Test Magazine's special issue on memory testing to be published in 1993.

Dr. Mazumder is a member of IEEE, Sigma Xi, Phi Kappa Phi, and ACM SIGDA.

Janak H. Patel is with the University of Illinois at Urbana-Champaign, where he is currently a Co-Director of Center for Reliable and High Performance Computing and a Professor of Electrical and Computer Engineering and Computer Science, and a Research Professor with the Coordinated Science Laboratory. He is a co-founder of Sunrise Test Systems, an ATG and testability company. He has also provided consulting and tutorial services to design and test industry. He is currently engaged in teaching, research, and consulting in the areas of automatic test generation, design for testability, fault simulation and diagnosis. Dr. Patel is a Fellow of the IEEE and holds a B.Tech. degree from Indian Institute of Technology, Madras, India, and M.S. and Ph.D. degrees from Stanford University.