

Built in Self Testing for Detection of Coupling Faults in Semiconductor Memories¹

Mark G. Karpovsky
Department of Electrical Computer and System Engineering
Boston University
44 Cummington Street, Boston MA 02215 USA
E-mail: mr@enga.bu.edu

Debaleena Das
DAD, Texas Instruments India
Wind Tunnel Road, Bangalore, India
E-mail: ddas@india.ti.com

Harsh Vardhan
Digital Semiconductor, A Digital Equipment Corporation business
77 Reed Road, HLO2-2/B10, Hudson, MA 01749 USA
E-mail: harsh.vardhan@hlo.mts.dec.com

ABSTRACT

In this work we investigate the problem of detection and location of single and unlinked multiple pattern sensitive faults in bit oriented RAMs and implementation of a built-in self-test (BIST) unit to test RAM chips in an efficient manner. Our fault model covers cross-talks between any k cells in RAMs. We have reduced the problem of memory testing to the problem of the generation of exhaustive backgrounds. Exhaustive tests for detecting coupling faults for k up to 6 (which covers Type I neighborhood) and near exhaustive tests for k up to 9 (which covers Type II neighborhood) are constructed.

The systematic nature of the tests constructed enables us to use BIST schemes, for RAMs, with low hardware overheads. We implemented the BIST units for a bit-oriented memory of size 1M and calculated the hardware overhead in terms of transistors and area.

1 Introduction

In this work we describe a new method for Random Access Memory (RAM) testing based on the PS(n,k) [16]

fault model. This model includes most classical fault models for SRAMs (Static RAMs) and DRAMs (Dynamic RAMs).

The problem of memory testing for PS(n,k) faults will be reduced to the problem of generation of ($n,k-1$) exhaustive backgrounds, where n is the size of the memory under consideration and $k-1$ is a parameter from the fault model of the memory. A background is defined as a memory state that is loaded into the memory.

A new method for constructing the backgrounds matrix $B(n,k-1)$ with minimal number of rows for n of the order of a million (the size of commonly used RAM chips) is described. We also propose the construction of matrices $B(n,k-1,\epsilon)$ such that in any $k-1$ tuple of columns all 2^{k-1} binary vectors appear at least once, except for a fraction of $k-1$ tuples which does not exceed ϵ . These matrices can be applied for the detection of all PS(n,k) faults with a probability of at least $1-\epsilon$.

1.1 Memory Modelling and Functional Faults

A functional model of a RAM, typically used for functional testing, contains 3 blocks. The blocks are: address decoder, memory cell array and the read-write

1. This research was supported in part by the National Science Foundation under grant MIP 9208487

logic. The fault model gives rise to the following classes of faults:

1. Fault in which a single cell is involved: These are the stuck-at-faults (SFs) and transition faults (TFs).

2. Faults in which two cells are involved: These are the coupling faults.

3. Faults involving k cells

3.1 Neighborhood Pattern Sensitive Faults(NPSFs): The k cells are clustered together in a physical neighborhood. These can be of the following types:

3.1.1 Active NPSFs or Dynamic NPSF[14][3]: The base cell changes its contents due to the change in the neighborhood pattern. These changes consist of a transition in one bit of deleted neighborhood, while the remaining deleted neighborhood and the base cell contains a certain pattern.

3.1.2 Passive NPSFs[3]: The contents of the base cell can not be changed due to a certain neighborhood pattern.

3.1.3 Static NPSF [3]: The contents of a base cell is forced to a certain state due to a certain neighborhood pattern.

3.2 k-Coupling Faults[15]: The k cells are allowed to be located anywhere in the memory. The k-coupling fault model is a generalization of the coupling fault described earlier. A set of k cells is said to be k-coupled when a WRITE operation of one cell produces a change in the contents of a second cell, subject to a particular data pattern in the remaining k-2 cells, which may be anywhere in the memory. k coupling faults cover ANPSFs and SNPSFs (neighborhood is not fixed).

4. Faults in the address decoder (AFs): AFs concern faults in the address decoder. It is assumed that the address decoder has not change into sequential logic due to the faults. Additionally, it is assumed that the faults in the decoder logic are the same during read/write operations.

1.2 Testing Strategies for RAMs

Due to miniaturization of RAMs, the types of faults have become more complex and the emphasis now is on the detection of NPSFs. Two types of neighborhoods were defined by Suk and Reddy [14] as being most appropriate for memories arranged as two dimensional arrays. An overview of tests for NPSFs developed by Suk and Reddy, Saluja and Kinoshita, and de Jong and van de Goor is given

in [3].

In all of the above tests the mapping from logical cell addresses to physical cell locations has to be known. However, within most memory chips the address lines are scrambled. This results in cells, with logically ascending addresses, not being physically next to each other. The scrambling table, which describes the relationships between the logical addresses and physical addresses, is usually known only to the manufacturers. In such a case, it is difficult to test for NPSFs.

Thus, there is a need for efficient tests to detect NPSFs regardless of the address mapping. The first step in this direction was the formulation of k-coupling fault model by Nair et al [15].

The 5-coupling fault model covers active NPSFs with a Type I neighborhood and the 9-coupling fault model covers Type II neighborhood.

The latest results in deterministic tests for the detection of k-coupling faults in RAMs are by Cockburn and can be found in [1]. All the above tests were for bit oriented memories. Extension to word-oriented memories has been done in [21].

1.3 Proposed Fault Model and Main results

The fault model considered in this work is the k out of n pattern sensitive fault model PS(n,k), where n is the size of the memory under test. The PS(n,k) fault model is one of the most general fault models which encompasses all the functional fault models mentioned in Section 1.1[21]. According to this model, the contents or the ability to change the contents of any memory cell in an n-bit memory is influenced by the contents or change in the contents of any other k-1 cells of memory.

The PS(n,k) fault model is similar to the one used in [16][20]. Our modification is that we will consider the memory as a whole instead of breaking it into blocks.

For $k < n$ and any k, PS(n,k) represents k-coupling faults. This can cover all NPSFs (Active, Passive and Static) with the size of the neighborhood equal to k. To cover these NPSFs (n,k-1) exhaustive backgrounds have to be used and for each background every cell has to undergo two consecutive sets of READ followed by WRITE - the *march element* [3]. The WRITES complement the data in the cells so we return to the original background after the march element. Addressing of cells can be in increasing or decreasing order. If the second READ is dropped, only ANPSFs and SNPSFs will be detected. The data presented henceforth excludes the second read.

TFs and SFs are covered by this model for $k=1$. This model does not directly cover faults in the address decoder (AFs). However the march component of our tests described

will cover the detection of all single AFs.

We construct both exhaustive and near-exhaustive tests for the detection of k-coupling faults. We have obtained practical test lengths, for a memory size around 1M, for detecting up to 6-couplings by exhaustive tests and 9 couplings by near-exhaustive tests. The best known results at the present time (by Cockburn [1]) provide for the detection of 5-couplings in a 1M memory, using exhaustive tests. Beyond these parameters, test lengths were impractical.

Furthermore, our method for generation of (n,k) exhaustive backgrounds[2] yields much shorter tests than the tests generated by Tang and Chen [10] and used by Cockburn[1]. As an example, our test for (1M,4) exhaustive backgrounds is 50% shorter than the test used in [1].

We have also obtained practical test lengths, for a memory size around 1M to cover up to 9-couplings using near exhaustive tests. These tests have fault coverage very close to one.

2 Exhaustive Tests

In a memory of size n, each cell can be affected by cross-talks with at most k-1 other cells of the memory. The problem of testing such memories can be reduced to generation of (n,k-1)-exhaustive backgrounds, where n is the size of memory and k is the size of the neighborhood. The k cells which form the neighborhood can be anywhere in the memory.

Definition 2.1: A (n,k-1)-exhaustive background is a matrix $B(n,k-1)$, with rows

$B_i = (B_i^0, B_i^1, \dots, B_i^{n-1})$, where $B_i^j \in \{0, 1\}$ and $j \in \{0, 1, \dots, n\}$ such that in any k-1 tuple of columns all 2^{k-1} binary vectors appear at least once as rows.

In the next section we will use a new and more efficient way of constructing exhaustive backgrounds for design of exhaustive tests for detection of PS(n,k) faults.

2.1 Construction of (n,k)-Exhaustive Backgrounds

Lemma 2.1.1: Consider the matrix, M, with number of rows R and number of columns $n \geq k$ such that the entry in the ith row and jth column is j modulo P_i where P_i is a prime number assigned to each row i. P_0 is the smallest prime number $\geq k$. P_1 is the next larger prime and so on (P_i and P_j are all different). Consider any k columns of matrix M:

$$0 \leq j_1 \leq j_2 \leq \dots \leq j_k \leq n-1,$$

and define $\Delta(j_1, j_2, \dots, j_k) = \prod_{p>q} (j_p - j_q),$ (1)

$$\Delta(n, k) = \max_{j_1, \dots, j_k} \Delta(j_1, j_2, \dots, j_k). \quad (2)$$

$$\text{If } \prod_{i=0}^R P_i > \Delta(n, k), \quad (3)$$

then any k columns in M will have at least one row in which all the entries are different.

Corollary 2.1.1: By the exclusion of all prime numbers less than k ($P_0 \geq k$)

$$\text{the inequality } \prod_{i=0}^R P_i > \Delta(n, k),$$

$$\text{may be strengthened to } \prod_{i=0}^R P_i > C(k) \times \Delta(n, k),$$

where $C(k) = C_2(k)C_3(k)\dots C_p(k)$; p is the largest prime number $< P_0$ and $C_i(k)$ is a proper fraction, which occurs as a result of the exclusion of the prime number i.

As an example, when the prime number 2 is excluded from the product of primes (4), the factors in $\Delta(n,k)$ can not be 2 or multiples of 2.

Corollary 2.2.2: The values of $\Delta(n,k)$ for $k = 3, 4, 5$ are:

$$\Delta(n,3) = (n-1)^3/4, \quad (4)$$

$$\Delta(n,4) = 0.018 (n-1)^6, \quad (5)$$

$$\Delta(n,5) = 3.3 \times 10^{-4} (n-1)^{10}, \quad (6)$$

Example 2.1.1: Consider $n = 10$ and $k = 3$, then

$$\Delta(10,3) = (n-1)^3/4 = 182.25,$$

$$C(k) = C_2(k) = 2^{-1},$$

$$\prod_{i=0}^R P_i > 2^{-1} \times 182.25, P_0 = 3, P_1 = 5, P_2 = 7,$$

($R=2, P_R=7$)

$$M = \begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 \end{bmatrix} \end{matrix}$$

and in any 3-tuple of columns in M there exists a row such that all entries in this row are different in the selected columns.

Lemma 2.1.2: If all the entries in the ith row and jth column of M are replaced by the jth column of a $B(P_i, k)$ matrix. then it yields a $B(n, k)$ matrix with the number of rows $T_{n,k}$, where

Fractions of good k -tuples, $1-\epsilon$ in $B(n,k-1,\epsilon)$ are presented in Table 2:

Table 2: $1-\epsilon$ for $n = 2^{18}$ to $n = 2^{24}$

$k \backslash s$	11	12	13	14
4	0.997	0.999	0.999	1.000
5	0.993	0.997	0.998	0.999
6	0.985	0.992	0.996	0.998
7	0.970	0.985	0.993	0.996
8	0.939	0.969	0.985	0.992
9	0.881	0.931	0.969	0.985

The number of rows in $B(n,k-1,\epsilon)$ (the number of ϵ -exhaustive backgrounds) is equal to $T_{n,k-1,\epsilon} = 2^{s+1}$.

3.3 Estimation of Testing Times

For ϵ -exhaustive testing of memories it takes an average of 4.5 clocks to write a background into the memory and complete the **march test** for one cell. Hence, we can write the expression for the total number of operations (N) required for testing as $N = \frac{9}{2}nT_{n,k,\epsilon}$, where $T_{n,k,\epsilon}$ is determined by Proposition 3.1.

Table 3 gives the testing times for memories of various sizes.

Table 3: Testing Times t for ϵ -Exhaustive Method with $k=9$, $\epsilon=0.01$ (access time = 100ns)

n	256k	1M	4M	16M
t	1.07h	4.30h	17.18h	68.72h

4 Design of BISTed RAMs

RAMs are often embedded in large integrated circuits which makes it difficult to access the RAM inputs and outputs for testing purposes. The solution is to use built-in-self-test (BIST) scheme. BIST also has other advantages like the reduction of expensive test equipment and the possibilities of *at-speed* testing [16]. In this section, we will discuss BIST implementation for exhaustive tests described in Section 2 and near exhaustive tests described in Section 3.

4.1 System Architecture for BIST implementation of Exhaustive Tests

The architecture of the proposed BIST RAM Scheme is given in Fig 1. The $n \times 1$ random access memory to be tested is the block labeled - Memory Under Test. The internal details of the test generator are shown in Fig 2.

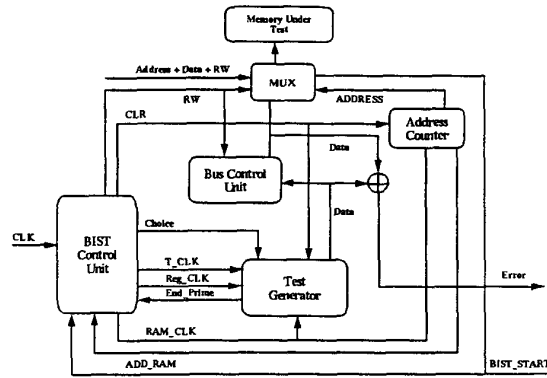


Fig 1: BIST Architecture for Exhaustive Testing

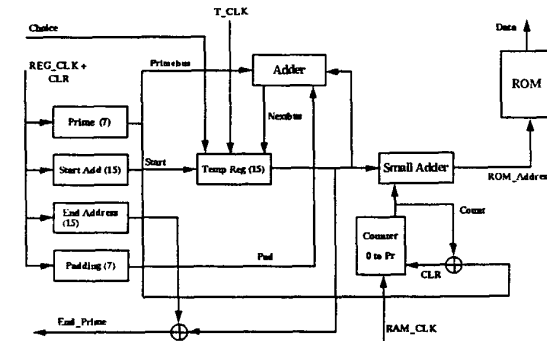


Fig 2: Test Generator for Exhaustive Testing

The BIST controller controls the sequence of the operations involved in the BIST using the internal status signals. This design has been done for ANPSFs and SNPSFs only. To cover PNPSFs as well, the only modification needed is an extra state between S_8 and S_9 which will correspond to the second read operation in the **march element**. The controller behavior is illustrated in Fig 3.

$$T_{n,k} = \sum_{i=0}^R T_{P_i,k}.$$

We will call matrices $B(P_i,k)$ ($i=0,\dots,R$) seeds for $B(n,k)$,

$$\text{if } \prod_{i=0}^R P_i > C(k) \times \Delta(n,k),$$

2.2 Estimation of Testing Times

From the state diagram in Fig 2 we can see that it takes an average of 4.5 clocks to write a background and complete the **march test** for one cell. Hence we can write the expression for the total number of operations required for testing (N) as:

$$N = \frac{9}{2} n T_{n,k-1}, \quad (7)$$

where $T_{n,k-1}$ is defined by Lemmas 2.1.1, 2.1.2.

Values for testing times for a few typical sizes of memories are given in Table 1.

Table 1: Testing Times t for Exhaustive Method with $k=5$, (access time = 100ns)

n	256k	1M	4M	16M
t	3.26m	14.43m	1.07h	4.71h

3 Near Exhaustive Probabilistic Tests

3.1 Survey of Probabilistic Tests

It has been shown in Section 2 that the problem of testing memories, of size n and each cell being affected by cross-talks with at most $k-1$ other cells, can be reduced to generation of $(n,k-1)$ -exhaustive backgrounds. In this section we consider the construction of $(n,k-1,\epsilon)$ -exhaustive backgrounds instead of $(n,k-1)$ -exhaustive backgrounds. A set of $(n, k-1, \epsilon)$ -exhaustive background can be represented by a matrix $B(n, k-1, \epsilon)$ such that in any $k-1$ tuple of columns all 2^{k-1} binary vectors appear at least once, except for a fraction of $k-1$ tuples which does not exceed ϵ ($B(n,k-1,0)=B(n,k-1)$). The number of rows in $B(n,k-1,\epsilon)$ is denoted by $T_{n,k,\epsilon}$. Similar to the situation in information theory, it can be expected that the number of test backgrounds can be reduced substantially if we allow a small fraction ϵ of all the possible $k-1$ tuples of columns not to be tested exhaustively. This effect holds even if ϵ approaches

zero with the increase in n [2].

Chandra et al. considered the effectiveness of probabilistic methods for finding (n,k) exhaustive codes for $k \geq 3$ [4]. Cockburn [18] suggested that effective probabilistic tests for detecting k -coupling faults can be obtained by basing the tests on random $n \times m$ matrices where $m \gg 1$. Probabilistic tests have also been evaluated by Savir, McAnney and Vecchio[7].

3.2 Construction of Near-Exhaustive Tests

There is a good construction of ϵ -exhaustive tests based on linear codes [2]. Let $H(s)$ be the matrix formed by the codewords of a $(2^s, s)$ simplex code as the rows, where $s \geq k$ [22]. This is the Hadamard matrix over $\{0,1\}$. The columns of $H(s)$ also form a s -dimensional linear space with deleted vector of all zeros. We propose to use this matrix for generation of $B(n,k-1,\epsilon)$ backgrounds matrix.

(Hadamard matrices are used since they are two valued matrices and have a simple procedure for generation of their components which results in a low hardware overhead (see Section 5). Multivalued generalizations of Hadamard matrices (Chrestenson-Galois transforms and Reed Solomon Codes) have been used for testing of word oriented memories [21].)

Proposition 3.1: For any $s \geq k$ and any $a < n$, there exists an ϵ -exhaustive test with $T_{n,k,\epsilon}$ backgrounds

$$\text{where } n = a2^s, T_{n,k,\epsilon} = 2^{s+1},$$

$$1 - \epsilon = (1 - \epsilon_v)(1 - \epsilon_h), \quad (8)$$

$$1 - \epsilon_v = \left(2^s \prod_{i=0}^{k-2} (2^s - 2^i) \right) \left(\prod_{i=0}^{k-1} (2^s - i) \right)^{-1}, \quad (9)$$

$$\text{and } 1 - \epsilon_h = a^k \binom{2^s}{k} \left(a \cdot 2^s \right)_k^{-1}. \quad (10)$$

One can use the following construction for the background matrix $B(n,k,\epsilon)$:

$$B(n,k,\epsilon) = \underbrace{\begin{bmatrix} \hat{H}(s) & \dots & \hat{H}(s) \end{bmatrix}}_a \quad (11)$$

$$\text{where, } \hat{H}(s) = \begin{bmatrix} H(s) \\ \bar{H}(s) \end{bmatrix}, \quad (12)$$

and $\bar{H}(s)$ is the negation of $H(s)$.

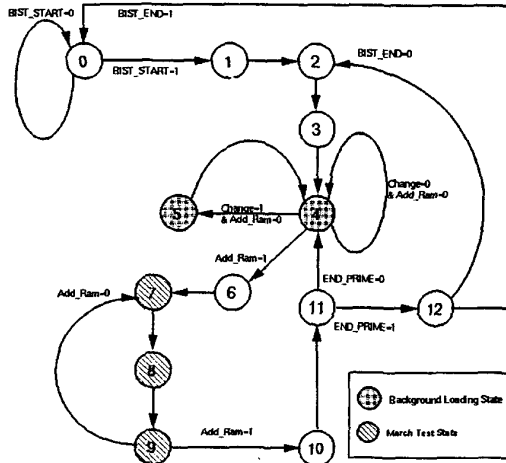


Fig 3: Controller State Diagram for Exhaustive Testing

4.2 BIST Implementation of Near-Exhaustive Tests

The BIST scheme using the Hadamard matrix is much simpler than the previous scheme. The subsystems ROM and address generator for ROM are not needed as there is no storage of seeds. Instead there has to be a subsystem to generate the Hadamard matrix on the fly. One simple approach for the generation of Hadamard matrix is based on the fact that each bit is the scalar product of the binary representation of its row and column numbers[19]:

$$H_{i,j}(s) = \langle i, j \rangle = i_0j_0 \oplus i_1j_1 \oplus \dots \oplus i_{s-1}j_{s-1},$$

where i and j are s -bit binary numbers;

$$i = (i_0, i_1, \dots, i_{s-1}), \quad j = (j_0, j_1, \dots, j_{s-1}).$$

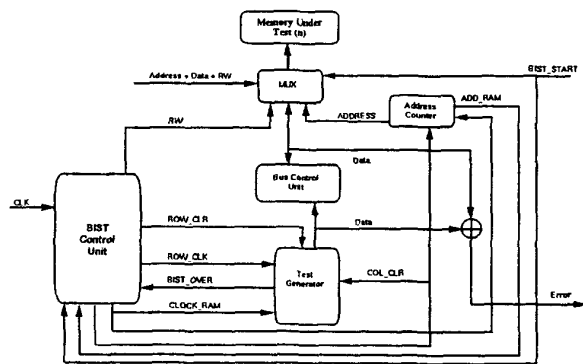


Fig 4: BIST Architecture for ϵ -Exhaustive Testing

The BIST controller architecture for this case is pre-

sented in Fig 4, the state transition diagram for the controller is given in Fig 6 and the block diagram for the test generator is given in Fig 5.

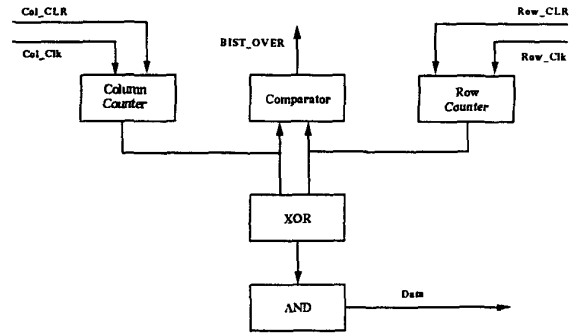


Fig 5: Test Generator for ϵ -Exhaustive Testing

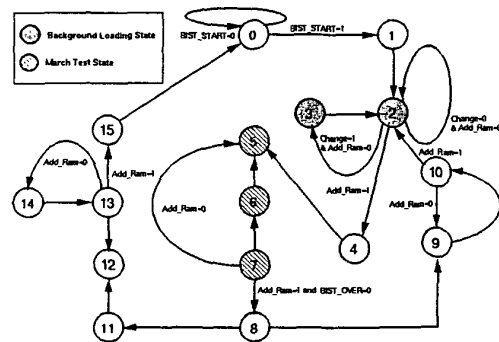


Fig 6: Controller State Diagram for ϵ -Exhaustive Testing

5 Conclusions

In this work we considered the problem of detection and location of single and unlinked multiple NPSFs (Active, Passive and Static) and couplings, with the size of the neighborhood equal to k , in $n \times 1$ RAMs, for $k = 4, 5, 6, 7, 8, 9$ and n of the order of $10^6/10^7$. A general fault model, the PS(n, k) fault model, covering these faults was proposed. In Table 4 we summarize the test times for our construction and the best previously known construction. Finally we have described BIST schemes for our proposed tests and have evaluated the hardware area and transistor overhead for a memory size of 1M.

For the exhaustive testing approach we have chosen $k = 5$. The transistor overhead for the BIST unit was 7,500 transistors in a control unit and around 25,500 transistors in a ROM which stores the seeds for the backgrounds. Assuming that a static RAM has 7 transistors per cell, the hardware

overhead in terms of transistors comes out to 0.47%. Even for a dynamic RAM with one transistor per cell and ignoring the overhead for refresh logic, the overhead is only 3.3%. In terms of area, the overhead is less than 1% for an SRAM and 6.8% for a DRAM. Most of the overhead in this case is in the ROM.

In the near exhaustive approach we increased the value of k to be 9. The hardware overhead in terms of transistors was 2,600. This comes out to be 0.04% for a Static RAM chip and 0.26% for a Dynamic RAM chip. In terms of area the overhead is less than 0.2% for an SRAM chip and 1.5% for a DRAM chip. The BIST unit transistor and area estimations were made by using Viewlogic's Powerview toolkit to compile VHDL descriptions into standard cell designs. The TimberWolfSC standard cell placement and global routing tool was used to generate the final physical design using a gates library.

Table 4: Comparison of Testing Times (access time = 100ns)

	1M	4M	16M
Suk and Reddy (Type I ANPSFs)	10.4s	41.7s	2m46.9s
Cockburn et al. (SSCTEST) (5 couplings)	19.24m	1.28h	5.13h
Karpovsky et al. (5-EXH) (5 couplings)	14.43m	1.07h	4.71h
Karpovsky et al. (6-EXH) (6 couplings)	2.06h	10.09h	-
Karpovsky et al. (6-εEXH) (6 couplings) (ε=0.01)	1.07h	4.30h	17.18h
Karpovsky et al. (9-εEXH) (9 couplings) (ε=0.01)	4.30h	17.18h	68.72h

References

[1] B.F. Cockburn "Deterministic Tests for Detecting Single V-Coupling Faults in RAMs," *J. Electronic Testing*, v.5, no 1, pp 91-113, (1994)
 [2] L.B. Levitin and M.G. Karpovsky, "Exhaustive Testing of Almost all Devices with Outputs Depending on Limited Number of Inputs." *Open Systems and Information Dynamics*, v.2, no.3, pp

1-16, (1994)
 [3] A.J. Goor, "Testing Semiconductor Memories, Theory and Practice," *John Wiley & Sons*, Chichester, UK, (1991).
 [4] A.K. Chandra, L.T. Kou, G. Markowsky, and S. Zaks, "On Sets of Boolean n-vectors With all k-Projections Surjective," *Acta Informatica*, v.20, pp 103-111, (1983)
 [5] D.T. Tang and L.S. Woo, "Exhaustive Test Pattern Generation with Constant Weight Vectors," *IEEE Transaction on Comp.*, v-32, no 12, pp 1145-1150, (1983)
 [6] N.J.A Sloane, "Covering Arrays and Intersecting Codes." *J. Combinatorial Design*, v.1, no.1, pp 51-64, (1993)
 [7] Hongzhong Wu, "On the Construction of $L \times n$ Boolean Matrices With all $L \times k$ Submatrices Having 2k Distinct Row Vectors," *Technical Report, Sarlan University*, (1989)
 [8] G.D. Cohen and G. Zemor, "Intersecting Codes and Independent Families", submitted to IEEE Transactions on Information Theory, 1996.
 [9] L.B. Levitin and M.G. Karpovsky, "Efficient Exhaustive Tests Based on MDS Codes", *Proc. IEEE International Symposium on Information Theory*, Ann Arbor (1986)
 [10] D.T. Tang and C.L. Chen, "Iterative Exhaustive Pattern Generation for Logic Testing," *IBM Journal of Res. Develop.*, v.28, no.2, pp 212-219, (1984).
 [11] A.J. Goor and C.A. Verruijt, "An Overview of Deterministic Functional RAM Chip Testing," *ACM Computing Surveys*, v.22, no.1, (1990)
 [12] L.B. Levitin and M.G. Karpovsky, "Travelling Salesman Problem in the Space of Binary Vectors," *Proc. of International Symposium on Information Theory*, Norway (1994)
 [13] J.P. Hayes, "Detection of Pattern Sensitive Faults in Random Access Memories," *IEEE Transactions on Computers*, v-24, pp 150-157 (1975)
 [14] D.S. Suk and S.M. Reddy, "Test Procedures for a Class of Pattern-Sensitive Faults in Semiconductor Random-Access Memories," *IEEE Transactions on Computers*, v-29, pp 419-429, (1980)
 [15] R. Nair, S.M. Thatte, and J.A. Abraham, "Efficient Algorithms for Testing Semiconductor Random-Access Memories", *Dig. of Papers, 7th Intl. Conf. on Fault Tolerant Computing*, Los Angeles, CA, pp 81-87, (1977)
 [16] M.G. Karpovsky and V.N. Yarmolik, "Transparent Memory Testing for Pattern Sensitive Faults," *Proc. International Test Conference*, pp 862-867, (1994)
 [17] J. Savir, W.H. McAnney, and S.R. Vecchio, "Testing for Coupled Cells in Random-Access Memories," *IEEE Transactions on Computers*, v-40, n 0.10, pp 1177-1180, (1991)
 [18] B.F. Cockburn, "A 20MHz Test Vector Generator for Producing Tests that Detect Single 4- and 5- Coupling Faults in RAMs." *Records of the 1993 Intl. Workshop on Memory Testing*, San Jose, CA, (1993)
 [19] M.G. Karpovsky, "Finite Orthogonal Series in the Design of Digital Devices", *John Wiley*, New York (1976)
 [20] M.G. Karpovsky and V.N. Yarmolik, "Transparent Memory BIST", *Proc. Intl. Workshop on Memory Tech.*, pp 106-112, (1994)
 [21] M.G. Karpovsky, V.N. Yarmolik and A.J. van de Goor, "Pseudoexhaustive Word-Oriented DRAM Testing", *Proc. 1995 European Design and Test Conference*, (1995)
 [22] F.J. MacWilliams and N.J.A. Sloane, "The Theory of Error-Correcting Codes", *North Holland Publishing Company*, Amsterdam, (1977).