ICSE 2008 Proc. 2008, Johor Bahru, Malaysia

12N Test Procedure for NPSF Testing and Diagnosis for SRAMs

Julie R.R., Wan Zuha W.H and R.M. Sidek
Department of Electrical and Electronics Engineering
University Putra Malaysia
Email: julie@bmi.unikl.edu.my

Abstract- **Testing and diagnosis techniques play a key role in the advance of semiconductor memory technologies. The challenge of failure detection has attracted investigation on efficient testing and diagnosis algorithm for better fault coverage and diagnostic resolution. March algorithms are widely used in SRAM testing to detect and diagnose SRAM fault model since they are relatively simple and yet providing high fault coverage and diagnostic resolution. In this case to achieve high fault coverage the structure of the consecutive memory backgrounds are very important. This paper aims to prove the efficiency of March 12N algorithm in term of detection and identification capability and locate the NPSF model fault. The details of test and diagnosis procedures for NPSF are demonstrated in this paper. The fault detection and diagnostic of the SRAM memories in this paper is verified and proven. The required march elements, detection requirement, detection conditions and fault syndromes are also enlightened. Furthermore, these particulars are required to determine a good algorithm other applications.**

Keyword; ***Neighborhood Pattern Sensitive Fault, Test Procedure, March algorithm, multi data background SRAM.***

I. Introduction

In semiconductor Random Access Memory technology, the increasing number of memory cells on a single chip will cause more failure modes and faults need to be dealt with in order to maintain a good quality and reliability of the shipped products. Then memory testing becoming a critical issue and resulting high test cost. The need for efficient algorithms to detect fault is then becoming more critical.

Many work on fault model and test algorithms have been reported in the past for SRAM. It is known that the effectiveness of memory test algorithms can be obtained by test analysis [1].

In memory testing Neighborhood Pattern Sensitive Fault is among the faults that have appeared increasingly often in the literature on RAM testing due to the increasing number of memory cells packed into a single chip. A traditional march test cannot guarantee 100% coverage for NPSF. This is due to the fact that detection of NPSF depends on the contents of a number of coupling cell which can be arbitrary. At the same time the classic march test does not provide all possible test pattern's for NPSF. However, with multiple data background 100% coverage can be obtained for NPSF.

The purpose of this paper is to present in details on how march-based algorithms performs the detection and diagnosis on the SRAM memories for the Neighborhood Pattern Sensitive Fault (NPSF). Based-on this, a realistic flow of testing and diagnosis are then demonstrated. All these particulars can be used to design an automatic march test algorithm generator. This tool helps to generate more faster and accurate march-based test and diagnosis algorithm.

II. Fault Models And Notation

A pattern sensitive fault is a multi-cell coupling fault. The Pattern Sensitive Fault model allows the deleted neighborhood to take on any position in the memory array. When the deleted neighborhood is allowed to take on only a single position, one speaks about a NPSF. The model is therefore a subset of the PSF model [1].

All known algorithms specify that the deleted neighborhood has to be in the physical proximity of the base cell because that deleted neighborhood is most likely to influence the base cell.

 1-4244-2561-7/08/$20.00 ©2008 IEEE

I. Fault Model

The fault model is categorized into three sub-types of faults as follows [1].

i. Active NPSF / Dynamic NPSF

The definition of Active NPSF is the contents of the base cell change its contents due to changes in the pattern of the deleted neighborhood. An ANPSF occurs if the base cell is forced to a certain state when a transition occurs in a deleted neighborhood cell, while other deleted neighborhood cell assume a certain pattern. To detect type 1 ANPSF all the 128 test patterns must be applied and the generation of this pattern within neighborhoods cell by the test algorithm must be verified.

ii. Passive NPSF

The definition of Passive NPSF is the contents of the base cell cannot be change due to a certain deleted neighborhood. A PNPSF occurs if the base cell cannot change its state from 0 to 1 or from 1 to 0 due to the appearance of a certain pattern in the deleted neighborhood. To detect type 1 PNPSF all the 32 test patterns must be applied and the generation of this pattern within neighborhoods cells by the test algorithm must be verified.

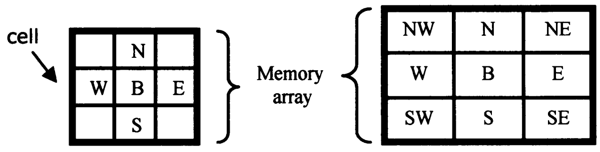
iii. Static NPSF

The definition of Static NPSF is the contents of a base cell are forced to a certain value by the contents of its deleted neighborhood. ASNPSF occurs if the base cell is forced to a certain state due to the appearance of a certain pattern in the deleted neighborhood. To detect Type 1 SNPSF all the 32 test pattern must be applied and the generation of these patterns within neighborhoods cells by the test algorithm must be verified.

Neighborhood is the total number of cell involved in a particular fault model. To perform the test for NPSF there is two type of neighborhood to be considered type 1 and type 2.

A type 1 neighborhood (Fig1.a) consists of five cells: the base cell. The four cells adjacent to the base cell on the same row and column as the base cell called deleted neighborhood cells.

A type 2 neighborhood (Fig1.b) consists of nine cells. The base cell and the eight cells adjacent to the base cell along the horizontal, vertical and both diagonal axes cell called deleted neighborhood cells.



a) Type 1 Neighborhood　　b) Type 2 Neighborhoods

Base cell/cell under test = B
Deleted neighborhood = N, W, E, S, NW, NE, SW, SE
Neighborhood = B+D

Fig.1. Type of neighborhood

An ANPSF is donated as (NWES; B/Bf) where N, W, E, S, B and Bf Є {010↑;0/1}. Note that B is the fault-free value of the base cell; Bf is the faulty value and N, W, E and S are the value of north, west, east and south of deleted neighborhood cells [2] as shown in figure 1a.

II. Notation

In this paper we are using March 12N algorithm, the following symbols and notation are used in this paper.

0(1)	= cell in logic state zero (one)
a	= cell in logic state a (origin state, either *0* or *1*)
x	= cell content is 'don't care for us.
↑(↓)	= write *1(0)* operation to a cell containing *0(1)*
↕	= write $\overline{a}$ operation to a cell containing a
Wa	= write *a* (*a* Є *{ 0,1}*) operation to a memory cell
ra	= read from a cell where *a* (*a* Є *{ 0,1}*) is expected
⇑	= addressing order is up
⇓	= addressing order is down
⇕	= addressing order is 'don't care'
A(B)	= March test phase, where *A* is an addressing order *(A* Є {⇑,⇓,⇕}, while *B* is a set of operations Applied to the cell *(ra, wa,* $r_{\overline{a}}$*, or* $w_{\overline{a}}$*).*

III. Test And Diagnosis Procedures

I. Proposed Test Algorithms

A march test consists of a sequence of march elements. A march element consists of a sequence of operations applied to each cell in the memory. An operation can consist of writing and reading a cell under test[1]. Note that the *w0* and *r0* operation are substitute with the *wa* and *ra* operations, where *a* is the state of a memory cell. Also, *w1* and *r1* are substituted with *wb* and *rb* respectively, where *b* is the complement of *a*.

The proposed algorithms are based on the multibackground method. NPSF required the multiple data background method which uses multiple data backgrounds to generate all possible neighborhood patterns under the March address sequences.

The proposed March-12N can detect not only ANPSF but also PNPSF, SNPSF and also conventional faults. Table 1 shows the ANPSF detected by March-12N under eight data background.

$$\{\Updownarrow(wa);\Uparrow(ra,wb,wa);\Uparrow(ra,wb);\Uparrow(rb,wa);\Uparrow(rb,wa,wb);\Updownarrow(ra)\}$$

March-12N algorithm

The test length of march-12N with eight data background is 96N and can detect 100% NPSF [2, 3]. The March-12N has total 12 operations including read/ write and it is divided into 6 elements. Each element has difference combination of read/write operations which good combination will provide a good coverage.

II. Address Scrambling

One of the constructive solutions to achieve high fault coverage in term of NPSF, as been shown in [1,2,3,4], is the multi-background method. To achieve high fault coverage of NPSF for multi run memory testing it is quite important to choose appropriate backgrounds depending on the type of memory test.

Figure 2 shows eight proposed data background for detecting ANPSF under March-12N algorithm. Each data background can detect 24 ANPSF. The total eight data back ground will cover all 128 ANPSF [2].

III. Background

One of the constructive solutions to achieve high fault coverage in term of NPSF, as been shown in [1,2,3,4], is the multi-background method. To achieve high fault coverage of NPSF for multi run memory testing it is quite important to choose appropriate backgrounds depending on the type of memory test.

Figure 2 shows eight proposed data background for detecting ANPSF under March-12N algorithm. Each data background can detect 24 ANPSF. The total eight data back ground will cover all 128 ANPSF [2].

IV. Test Procedure

The test procedure and diagnosis developed in this paper is based on NPSF type1 model which is have 5 cell (1 base cell and 4 deleted neighborhood cell) as shown in figure 1.a. Different type of NPSF will generate different test procedure and fault syndrome. Test pattern is used to load the possible input vector to check the faulty in the cell. The test pattern known as March-12N test algorithm is used in this procedures.

For PNPSF they have total 32 type of fault, SNPSF have total 32 type of fault and the higher fault is ANPSF with the 128 type of fault. ANPSF have 128 number of test pattern which is the total fault type for ANPSF.

In this paper, the development test and diagnosis procedure for ANPSF with 128 types of faults are covered. By using March-12N test algorithm, we need to used eight data background in order to detect all these 128 type of fault. Figure 2 shows the eight data background which each data background can detect 24 type of fault, so 24 x 8 = 192. However there are only 128 ANPSFs, so some ANPSF detected by multiple data backgrounds. In Table 1 shows only 96 ANPSF detected, but the actual is 192 because ($\updownarrow=\uparrow\downarrow$) so that the total ANPSF detected is 96 x 2 = 192. The proposed March algorithm with eight backgrounds has a test length of 12N x 8 = 96N.

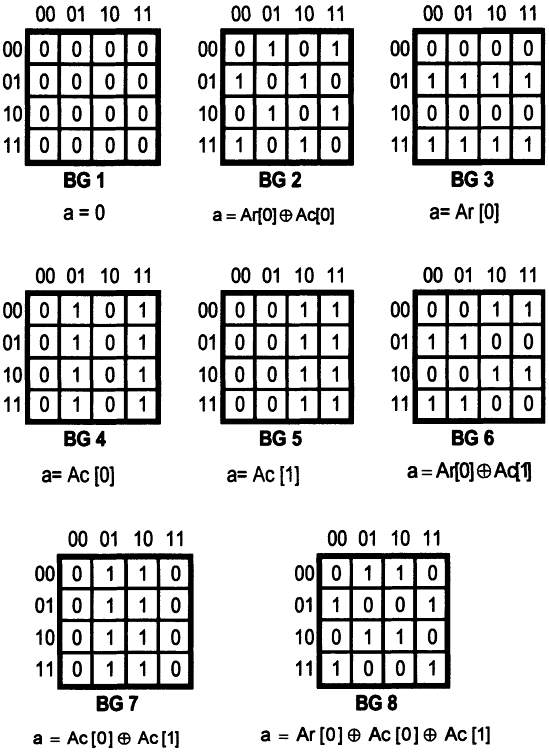


Fig. 2. The eight backgrounds with bit generator.

Table 1. ANPSF detected by 8 data backgrounds using March-12N algorithm.

BG1	BG2	BG3	BG4	BG5	BG6	BG7	BG8
(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)	(NWES;B/BF)
(↕000;0/1)	(↕000;1/0)	(↕110;1/0)	(↕001;1/0)	(↕010;0/1)	(↕100;1/0)	(↕011;1/0)	(↕101;0/1)
(0↕00;0/1)	(0↕00;1/0)	(0↕10;1/0)	(1↕01;1/0)	(0↕10;0/1)	(0↕00;1/0)	(1↕11;1/0)	(1↕01;0/1)
(00↕0;0/1)	(00↕0;1/0)	(01↕0;1/0)	(10↕1;1/0)	(00↕0;0/1)	(01↕0;1/0)	(10↕1;1/0)	(11↕1;0/1)
(000↕;0/1)	(000↕;1/0)	(011↕;1/0)	(100↕;1/0)	(001↕;0/1)	(010↕;1/0)	(101↕;1/0)	(110↕;0/1)
(0↕11;1/0)	(1↕00;1/0)	(0↕01;0/1)	(1↕10;0/1)	(0↕01;1/0)	(0↕11;0/1)	(1↕00;0/1)	(1↕10;1/0)
(11↕0;1/0)	(00↕1;1/0)	(01↕1;1/0)	(10↕0;1/0)	(00↕1;0/1)	(01↕1;1/0)	(10↕0;1/0)	(11↕0;0/1)
(1↕00;0/1)	(1↕00;1/0)	(1↕10;1/0)	(0↕01;1/0)	(1↕10;0/1)	(1↕00;1/0)	(0↕11;1/0)	(0↕01;0/1)
(11↕0;1/0)	(11↕0;0/1)	(10↕0;0/1)	(01↕1;0/1)	(11↕0;1/0)	(10↕0;0/1)	(01↕1;0/1)	(00↕1;1/0)
(↕111;1/0)	(↕111;0/1)	(↕001;0/1)	(↕110;0/1)	(↕101;1/0)	(↕011;0/1)	(↕100;0/1)	(↕010;1/0)
(1↕11;1/0)	(1↕11;0/1)	(1↕01;0/1)	(0↕10;0/1)	(1↕01;1/0)	(1↕11;0/1)	(0↕00;0/1)	(0↕10;1/0)
(11↕1;1/0)	(11↕1;0/1)	(10↕1;0/1)	(01↕0;0/1)	(11↕1;1/0)	(10↕1;0/1)	(01↕0;0/1)	(00↕0;1/0)
(111↕;1/0)	(111↕;0/1)	(100↕;0/1)	(011↕;0/1)	(110↕;1/0)	(101↕;0/1)	(010↕;0/1)	(001↕;1/0)

III. Result And Discussion

The complete testing and diagnosis procedures for NPSF have been logically verified in details. The March base algorithm was used to show that the fault are detected and diagnosed. In Table 2 shows the required march-based algorithm and march based detection condition. More than one march-based algorithm can be used to detect the faults and the march-based detection condition is written based on march based detection.

From table 2, March-12N algorithm is used to detect ANPSF, element M1⇑ (ra,wb,wa) with backgrounds all zero in deleted neighborhood (N, W, E, S) are able to detect ANPSF (↕000;0/1). Transition occurred in N cell, the test pattern applied to detect this fault is ↑000 & ↓000. The status of victim cell (base cell (B)) is state as fault free if it detects 1 value and faulty if it detects 0 values.

The same element in March-12N algorithm is used to detect ANPSF (↕110; 1/0) with (background (N, W, B, E, S) = (0, 1, 1, 1, 0). The same transition occurred in N cell and test patterns applied are ↑110 & ↓110. The status of victim cell (base cell (B)) is state as fault free if it detects 0 value and faulty if it detects 1 value.

Table 3 shows the list of all fault syndromes corresponding to algorithm March-12N. For example, if a fault was detected in element M1 with background all-0, it fault syndrome was 11000 which means the operation R0 in element M1 and R1 in element M2 can detected total 8 faults with all-0 background.

Furthermore, the diagnosis is verified throughout the read operation of the march-based detection. The purpose of this finding is to provide a good solution for implementing automatic march test generation and a good march-based algorithm can be obtained faster. This march-based algorithm enables to be implemented in designing memory built in self test and self diagnosis (MBIST/SD). The other advantage of these procedures is providing an easy step to understand on how faulty SRAM are being detected and diagnosed.

Table 2. Fault Syndrome for ANPSF

Name of Fault		Aggressor	Victim		Back ground
	Test Pattern	Deleted Neighbor hood	Base cell fault free	Base cell faulty	
ANPSF	NWES;B	NWES	B	Bf	NWBES
ANPSF ↕000;0/1	↑000 & ↓000= ↕000;0	↕000	0	1	00000
ANPSF (↕001;1/0)	↑001 & ↓001= ↕001;1	↕001	1	0	10101
ANPSF (↕010;0/1)	↑010 & ↓010= ↕010;1	↕010	0	1	00010
ANPSF (↕100;1/0)	↑100 & ↓100= ↕100;1	↕100	1	0	01100
ANPSF (↕011;1/0)	↑011 & ↓011= ↕011;1	↕011	1	0	10111
ANPSF (↕101;0/1)	↑101 & ↓101= ↕101;0	↕101	0	1	10111

Table 3. Fault Syndrome for March-12N

MARCH PS (12N) ALGORITHM		BG	ANPSF DETECTED	Fault Syndrome
Element		NWBES	(NWES;B/BF)	R0,R1,R2,R3,R4
M0	⇕(wa)	00000		
M1	⇑(ra,wb,wa)	00000	(↕000;0/1)(0↕00;0/1) (00↕0;0/1)(000↕;0/1)	1,1,0,0,1
M2	⇑(ra,wb)	00000	(1↑00;0/1)(11↑0;1/0)	0,1,0,0,0
M3	⇑(rb,wa);	00000	(0↓11;1/0)(00↓1;0/1)	0,0,1,0,0
M4	⇑(rb,wa,wb)	00000	(↕111;1/0)(1↕11;1/0) (11↕1;1/0)(111↕;1/0)	0,0,0,1,0
M5	⇕(ra);	00000		
		BG2		
M0	⇕(wa)	11011		
M1	⇑(ra,wb,wa)	11011	(↕111;0/1)(1↕11;0/1) (11↕1;0/1)(111↕;0/1)	1,1,0,0,1
M2	⇑(ra,wb)	11011	(0↑11;0/1)(00↑1;1/0)	0,1,0,0,0
M3	⇑(rb,wa);	11011	(1↓00;1/0)(11↓0;0/1)	0,0,1,0,0
M4	⇑(rb,wa,wb)	11011	(↕000;1/0)(0↕00;1/0) (00↕0;1/0)(000↕;1/0)	0,0,0,1,0
M5	⇕(ra);	11011		
		BG3		
M0	⇕(wa)	01110		
M1	⇑(ra,wb,wa)	01110	(↕110;1/0)(0↕10;1/0) (01↕0;1/0)(011↕;1/0)	1,1,0,0,1
M2	⇑(ra,wb)	01110	(1↑10;1/0)(10↑0;0/1)	0,1,0,0,0
M3	⇑(rb,wa);	01110	(0↓01;0/1)(01↓1;1/0)	0,0,1,0,0
M4	⇑(rb,wa,wb)	01110	(↕001;0/1)(1↕01;0/1) (10↕1;0/1)(100↕;0/1)	0,0,0,1,0
M5	⇕(ra);	01110		
		BG4		
M0	⇕(wa)	10101		
M1	⇑(ra,wb,wa)	10101	(↕001;1/0)(1↕01;1/0) (10↕1;1/0)(100↕;1/0)	1,1,0,0,1
M2	⇑(ra,wb)	10101	(0↑01;1/0)(01↑1;0/1)	0,1,0,0,0
M3	⇑(rb,wa);	10101	(1↓10;0/1)(10↓0;1/0)	0,0,1,0,0
M4	⇑(rb,wa,wb)	10101	(↕110;0/1)(0↕10;0/1) (01↕0;0/1)(011↕;0/1)	0,0,0,1,0
M5	⇕(ra);	1001		
		BG5		
M0	⇕(wa)	00010		
M1	⇑(ra,wb,wa)	00010	(↕010;0/1)(0↕10;0/1) (00↕0;0/1)(001↕;0/1)	1,1,0,0,1
M2	⇑(ra,wb)	00010	(1↑10;0/1)(11↑0;1/0)	0,1,0,0,0
M3	⇑(rb,wa);	00010	(0↓01;1/0)(00↓1;0/1)	0,0,1,0,0
M4	⇑(rb,wa,wb)	00010	(↕101;1/0)(1↕01;1/0) (11↕1;1/0)(110↕;1/0)	0,0,0,1,0
M5	⇕(ra);	00010		
		BG6		
M0	⇕(wa)	01100		
M1	⇑(ra,wb,wa)	01100	(↕100;1/0)(0↕00;1/0) (01↕0;1/0)(010↕;1/0)	1,1,0,0,1
M2	⇑(ra,wb)	01100	(1↑00;1/0)(10↑0;0/1)	0,1,0,0,0
M3	⇑(rb,wa);	01100	(0↓11;0/1)(01↓1;1/0)	0,0,1,0,0
M4	⇑(rb,wa,wb)	01100	(↕011;0/1)(1↕11;0/1) (10↕1;0/1)(101↕;0/1)	0,0,0,1,0
M5	⇕(ra);	01100		
		BG7		
M0	⇕(wa)	10111		
M1	⇑(ra,wb,wa)	10111	(↕011;1/0)(1↕11;1/0) (10↕1;1/0)(101↕;1/0)	1,1,0,0,1
M2	⇑(ra,wb)	10111	(0↑11;1/0)(01↑1;0/1)	0,1,0,0,0
M3	⇑(rb,wa);	10111	(1↓00;0/1)(10↓0;1/0)	0,0,1,0,0
M4	⇑(rb,wa,wb)	10111	(↕100;0/1)(0↕00;0/1) (01↕0;0/1)(010↕;0/1)	0,0,0,1,0
M5	⇕(ra);	10111		
		BG8		
M0	⇕(wa)	11001		
M1	⇑(ra,wb,wa)	11001	(↕101;0/1)(1↕01;0/1) (11↕1;0/1)(110↕;0/1)	1,1,0,0,1
M2	⇑(ra,wb)	11001	(0↑01;0/1)(00↑1;1/0)	0,1,0,0,0
M3	⇑(rb,wa);	11001	(1↓10;1/0)(11↓0;0/1)	0,0,1,0,0
M4	⇑(rb,wa,wb)	11001	(↕010;1/0)(0↕10;1/0) (00↕0;1/0)(001↕;1/0)	0,0,0,1,0
M5	⇕(ra);	11001		

IV. Conclusion

In this paper, procedure of testing and diagnosis SRAM has been demonstrated using March-12N algorithm. A March-12N algorithm fault location and full diagnosis algorithm is proposed for word-oriented memories. The developed of test and diagnosis procedure is to provide batter understanding on the faulty memory. In addition a good march-based algorithm with a batter faults coverage and diagnostic resolution of faulty SRAM can be determined in a shorter time.

REFERENCES

[1] A.J. van de Goor, *Testing Semiconductor Memories: Theory and Practice,* John Wiley & Son, Chichester, England,1991.

[2] K. L. Cheng, M. F. Tsai and C. W. Wu, "Neighborhood Pattern Sensitive Fault Testing and Diagnostics for Random-Access Memories", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, no. 11, Nov. 2002.*

[3] K. –L. Cheng and C.-W. Wu, " Efficient neighborhood pattern-sensitive fault test algorithms for semiconductor memories", *in Proc. IEEE VLSI Test Symp(VTS), Marina Del Rey, California, Apr 2001, pp. 225-237.*

[4] S.V. Yarmolik and I. Mrozek, "Multi Background Memory Testing", *In Proc. MIXDES 14th International Conference, Ciechocinek, Poland, June 200.*

[5] V.N. Yarmolik, Y. Klimets and S. Demidenko, "March PS(23N) Test for DRAM Pattern-Sensitive Faults", *In Proc. Seventh IEEE Asian Test Symp. (ATS), page 354-351, Singapore, December 1998.*

[6] D. S. Suk and S. M. Reddy, " Test Procedures for a Class of Patten Sensitive Faults in Semiconductor Random Access Memories", *IEEE Trans. On Computer, Vol. C-29, no.6, June 1980.*

[7] W.H. Wan Zuha, O. Mahsuri and S. Bambang Sunarjo, "Test and Diagnosis Procedures for Coupling Faults on SRAM Memories", *World Engineering Congress, Penang Malaysia, August 2007*

[8] C.F Wu, C.T Huang, K.L. Cheng and C.W. Wu, "Fault Simulation and Test Algorithm Generation for Random Access Memory", *IEEE Trans On CAD Of IC and System, Vol. 21, No.4, April 2002*