Diagnostic Testing of Embedded Memories Using BIST *

Timothy J. Bergfeld† Dirk Niggemeyer†† Elizabeth M. Rudnick†

†Center for Reliable and High-Performance Computing, University of Illinois
1308 West Main Street, Urbana, IL 61801, USA

††Laboratory for Information Technology, University of Hannover
Schneiderberg 32, 30167 Hannover, Germany

Abstract

The increasing use of large embedded memories in Systems-on-Chips requires automatic memory reconfiguration to avoid the need for external accessibility. In this work, effective diagnostic memory tests of linear order $O(N)$ are proposed that enable memory reconfiguration, and their diagnostic capabilities are analyzed. In particular, these tests allow single-cell faults to be distinguished from multiple-cell faults, such as coupling faults. In contrast to conventional $O(N)$ tests, all cells involved in a fault are detected and localized, which allows complete reconfiguration using minimal-area BIST hardware that compares favorably with other BIST designs.

1 Introduction

Decreasing feature sizes and the advent of Systems-on-Chips have lead to the incorporation of large embedded memories, and in particular, static RAMs, in microprocessors and other kinds of VLSI circuits. The steadily increasing percentage of area used for embedded memories contributes significantly to overall system failures. Therefore, the manufacturing yield of the complete systems can be increased by repairing the faulty on-chip memories using reconfiguration resources, such as spare rows and spare columns. To allow economical reconfiguration strategies to be processed on-chip, exact knowledge about the memory cells involved in the faults is required, most importantly the locations of defective cells.

Previous work on memory diagnosis has been done in the following areas. In [1], a set of pseudorandom experiments was utilized to distinguish between different faults in a probabilistic way. However, the test results are not well suited for Built-In Self-Test (BIST) based reconfiguration, since the test experiment application is highly dependent on the results from previous experiments. In [2], a separate processing element is utilized to perform diagnostic tests that distinguish between a number of faults. The main drawback of the approach used is that the test set utilized includes tests that are of higher order, such as $O(N^2)$, which leads to unacceptable test application times. A deterministic approach of linear order was presented in [3] which distinguishes between different kinds of coupling faults. However, the proposed March Cd test does not allow idempotent coupling faults (*CFid*) to be distinguished from inversion coupling faults (*CFin*), and it leads to misdiagnoses when pattern-sensitive faults (*PSF's*) are present. In [4], tests with optimal test length are proposed for a realistic fault model. Although the approach allows certain types of coupling fault aggressors to be detected and localized, it fails to distinguish between single-cell and multiple-cell-faults.

In our work, a linear order test approach is proposed that detects all faults of a given fault model and allows single-cell faults to be distinguished from multiple-cell faults, which enables economical use of spare resources for automatic on-chip reconfiguration. Both unlinked and linked memory faults are handled. (Linked faults may influence the behavior of other memory faults.) In particular, in the case of two-cell coupling faults, the approach detects and localizes both the coupled and the coupling cell. This is achieved by using supplementary BIST hardware instead of higher order algorithms, such as $O(N \log N)$ or higher. The complexity of the proposed BIST hardware compares favorably with common memory BIST realizations [5]. Since most embedded memories are organized as $2^n * w$ bits, with $N = 2^n$ addresses and $w \geq 2$ bits per word, extensions to test word-oriented memories from [6] and [7] are taken into account.

The paper is organized as follows. In Section 2, the functional fault model used in this work is presented. In Section 3, the diagnostic approach is described. Section 4 describes the BIST circuitry utilized. Section 5 contains simulation results that demonstrate the capabilities of the proposed diagnostic tests. The paper is concluded in Section 6.

2 Functional Fault Model

The functional fault model used in this work is part of the fault model presented in [6] and is based on the assumption

*This research was supported in part by DOE under grant DE-FGO2-91ER40677 and in part by DARPA under Contract DABT63-95-C-0069.

Table 1: Test results for distinguishing between single-cell faults and multiple-cell coupling faults. Results for coupling faults depend on whether the aggressor/victim cell is accessed first.

	$\Uparrow(w0)$;	$\Uparrow(r0,$	$w1,r1,$	$w0)$;	$\Updownarrow(r0)$;	$\Updownarrow(w1)$;	$\Uparrow(r1,$	$w0,r0,$	$w1)$;	$\Updownarrow(r1)$;
Defect-free memory		0	1		0		1	0		1
SAF-0		0	0		0		0	0		0
SAF-1		1	1		1		1	1		1
TF $<\downarrow/1>$		1	1		1		1	1		1
TF $<\uparrow/0>$		0	0		0		0	0		0
CFid: $<\uparrow,1>,<\downarrow,1>$		1/0	1/1		0/1		1/1	0/0		1/1
CFid: $<\uparrow,0>,<\downarrow,0>$		0/0	1/1		0/0		0/1	0/0		1/0
CFin: $<\uparrow,\updownarrow>,<\downarrow,\updownarrow>$		1/0	1/1		0/1		0/1	0/0		1/0

of fault-free read operations. The following faults are taken into account.

- Stuck-at faults (SAF-0, SAF-1): A cell can be stuck at logic value 0 or 1.
- Transition faults (TF$<\uparrow/0>$, TF$<\downarrow/1>$): A cell is unable to transition from 0 to 1 or vice versa.
- Idempotent coupling faults (CFid$<\uparrow,x>$, CFid $<\downarrow,x>$, $x \in \{0,1\}$): A write transition to the coupling cell forces the victim cell to a certain state.
- Inversion coupling faults (CFin$<\uparrow,\updownarrow>$, CFin $<\downarrow,\updownarrow>$): A write transition to the coupling cell forces the coupled cell to invert its state.
- Pattern-sensitive faults (PSF): A more general kind of coupling fault involving $k-1$ influencing cells in the memory. Active, passive, and static pattern-sensitive faults as defined in [6] are considered. The targeted test complexity of $O(N)$ does not allow complete detection and localization of these kinds of faults. PSF's are considered only to avoid misdiagnoses of single-cell faults and two-cell coupling faults.
- Address decoder faults (AF): Combinational address decoder faults can be mapped to equivalent faults within the memory array for test purposes. However, the functional equivalence leads to problems in diagnostic testing. Testing for sequential address decoder faults [8], [9] is not considered here.

3 Diagnostic Scheme

The proposed diagnosis strategy can be described as follows. First, a bit-oriented March test is performed to distinguish between single-cell and multiple-cell faults. In addition, in the presence of an unlinked two-cell coupling fault, the type of fault can be determined. In the case of a single-cell fault, the test and diagnosis procedure can be stopped here. Otherwise, if a coupling fault is detected, a March-like linear order test is applied to localize the coupling cell. Next, intra-word inversion coupling faults are targeted using the methods from [7]. Finally, if the type of fault has not yet been determined, the fault is likely to be an address decoder fault, and a specific March test can be applied to confirm the diagnosis. Throughout testing, the possibility of a PSF being present is considered to prevent corrupting the diagnosis results. The individual test and diagnosis steps are described in more detail in the following subsections.

3.1 Diagnosing single- and multiple-cell faults

In Table 1, the results of the initial March test sequence of length $12N$ are shown for the fault-free case and unlinked single or multiple faults in bit-oriented memories. Note that the test consists of two equivalent sequences of length $6N$, with inverted data in the second one. The results show that, with this specific test, it is possible to distinguish between the following five groups of faults: SAF-0/TF$<\uparrow/0>$, SAF-1/TF$<\downarrow/1>$, CFid$<x,1>$, CFid$<x,0>$, and CFin$<x,\updownarrow>$, with $x \in \{\uparrow,\downarrow\}$. Distinguishing between SAF's and TF's is not possible after initialization, e.g., utilizing $\Uparrow(w0)$. Diagnosis of the above groups of faults is possible since each group of faults is detected in a different set of read operations during test. It should be noted that the identical address order in $\Uparrow(r0,w1,r1,w0)$ and $\Uparrow(r1,w0,r0,w1)$ additionally guarantees detection of linked idempotent coupling faults [10]. The presence of PSF's, which would lead to incorrect diagnosis results, cannot be completely ruled out in this stage of diagnosis, unless the fault is a single-cell fault, for the following reason. The tests of length $6N$ use inverse data, i.e., opposite background patterns. Therefore, a PSF would only occur in one of the two tests and cannot be misdiagnosed as a single-cell fault. Since it can be diagnosed as a CFid or CFin, the subsequent stages of diagnosis have to be based on different background patterns to distinguish between coupling faults, which would be detected again, and a PSF, which would not be detected.

3.2 Locating the CF-Aggressor Cell

The aggressor cell of a coupling fault is located utilizing a March-like test phase of $O(N)$. In the proposed test, the location of the coupled cell is represented by the superscript

X; thus $r^X y$, with $y \in \{0,1\}$, verifies whether value y is read from the coupled cell. If the fault causes the coupled cell to make an up-transition, then the following test phase will detect the coupling fault and locate the aggressor:

$$\Updownarrow (w0); \Updownarrow (w^{N\setminus X}1, r^X 0, w^{N\setminus X}0, r^X 0)$$

where $N \setminus X$ represents the complete address space excluding the address of the coupled cell. The type of CF can be determined from the results of the read accesses. A coupling fault will either fail on both reads, if the fault is triggered by a write of 1 (e.g., CFid$<\uparrow/1>$), or on the second read if it is triggered by a write of 0 (e.g., CFid$<\downarrow/1>$). If only the first read fails, the presence of a complex fault outside the fault model is likely to be the cause. A similar test can be defined for faults causing the coupled cell to make down-transitions.

To prevent a pattern-sensitive fault from affecting a coupling cell diagnosis, this test is applied using a background pattern different from the background during detection in the first stage of diagnosis. To ensure test application with inverted – and therefore different – backgrounds, the memory has to be initialized to a value that depends on the first detection of the coupling fault in the previous tests. Fortunately, there are only two possibilities. If the coupling fault was detected in the first part of the single-cell tests (i.e., in $\Uparrow (w0); \Uparrow (r0, w1, r1, w0); \Updownarrow (r0)$), the previous background was 0 when the victim cell changed to a 1, and $B = 0$ (i.e., $B = \{00\ldots00\}$ for a word-oriented memory). If the fault was detected in the second part, the previous background was 1 when the victim cell changed to a 0, i.e., $B = 1$. In the complete test for locating the aggressor cell,the memory is initialized with the inverted background pattern:

$$\Updownarrow (w\overline{B}); w^X B; \Updownarrow (w^{N\setminus X}B, r^X B, w^{N\setminus X}\overline{B}, r^X B)\ ,$$

where $\overline{B}$ denotes the inverted background and $w^X B$ is a write to the victim cell with the original background value. (If the fault was detected in both parts of the test, the background $B = 0$ is utilized.) Note that after this test, all CF$< x, y >$, with $x \in \{\uparrow, \downarrow\}$, and $y \in \{0, 1, \updownarrow\}$, can be distinguished. A PSF that was detected in the previous test will not be detected in this test, and it is therefore correctly diagnosed as a PSF.

3.3 Extensions for Testing Intra-Word CFid's in Word-Oriented Memories

The test and diagnosis algorithms decribed so far have been based on a bit-oriented view of a memory; i.e., a single bit has been read or written during test application. The extension to word-oriented memories includes separate problems to be taken into account. First, multiple faults in different bits of a word do not allow diagnosis using a simple comparison with fault-specific expected data. Therefore, diagnosis has to be performed per bit, but in parallel for all w bits of a word. Secondly, while inter-word idempotent and inversion coupling faults are detected during word-wide reads and writes, intra-word coupling faults are not always detected. To resolve the intra-word coupling faults for word-wide memories, test patterns have been proposed in [7] (see Table 2), which are applied to each memory cell, resulting in an additional test phase. This test phase is shown for the case of a byte-wide memory organization, such as $2^n \times 8$ bits.

Table 2: Example of Intra-word Test Patterns for a Word-Oriented Memory with w=8 Bits per Word [7].

	Word Pattern		Word Pattern
0	00000000	6	00110011
1	11111111	7	11001100
2	00000000	8	00110011
3	01010101	9	00001111
4	10101010	10	11110000
5	01010101	11	00001111

$$\Uparrow (w01010101, r01010101, \ldots, w00001111, r00001111)$$

This test is of length $6N \log_2 w$, where w is the number of bits per word. The first three patterns from Table 2 do not have to be applied again, since these patterns were utilized for testing in the previous March phases.

3.4 Address Decoder Faults

The detection of (combinational) address decoder faults [6] has been neglected in the previous diagnostic tests. In [10], it was shown that all address decoder faults are detected by applying the March sequences $\Downarrow (rB, \ldots, w\overline{B})$ and $\Uparrow (r\overline{B}, \ldots, wB)$. To avoid misdiagnoses in the proposed diagnostic tests, the two test sequences are not included in the previous tests but are applied separately, along with an initialization sequence, such as $\Updownarrow (wB)$. The minimal-length sequences are concatenated in the applied test, leading to the well known MATS+ test [6]: $\Updownarrow (w\overline{B}); \Uparrow (r\overline{B}, wB); \Downarrow (rB, w\overline{B})$. Because address faults may cause a high number of cells to show faulty behavior and the amount of resources for reconfiguration is assumed to be restricted, diagnosis of the type of address decoder fault is not performed here. Thus, reconfiguration for address decoder faults is not addressed.

3.5 Diagnosis of Pattern-Sensitive Faults

The linear order tests that are proposed do not detect all PSFs. However, some pattern-sensitive faults are detected, and the diagnostic tests are designed to take these faults into account. No attempt is made to determine the type of PSF. Some coverage of PSF's is achieved, since the

cases when the neighborhood of a cell is in the all-0's and all-1's states are explicitly tested during the test for single-cell faults. These two states represent the patterns of highest influence from the deleted neighborhood for each of the memory cells. Furthermore, each cell performs both transitions with these backgrounds in the proposed tests and thus is tested for passive PSF's.

3.6 Summary of the Diagnostic Scheme

The different tests proposed are summarized here for the complete test and diagnosis approach, which applies to word-oriented memories. The first part of the resulting test is of length $12N$ and distinguishes between single-cell faults and different groups of two-cell coupling faults:

$$\Uparrow (wB); \Uparrow (rB, w\overline{B}, r\overline{B}, wB); \Updownarrow (rB);$$
$$\Updownarrow (w\overline{B}); \Uparrow (r\overline{B}, wB, rB, w\overline{B}); \Updownarrow (r\overline{B}) \ ,$$

with $B = \{00\ldots00\}$. In case of coupling fault detection, the following test of length $5N+1$, which locates the aggressor cell and distinguishes between different coupling faults, is applied:

$$\Updownarrow (w\overline{B}); w^X B; \Updownarrow (w^{N\setminus X} B, r^X B, w^{N\setminus X} \overline{B}, r^X B).$$

To detect intra-word CF's, and to distinguish between stuck-at faults and pattern-sensitive faults, the following intra-word test of length $6N\log_2 w$ (i.e., $18N$ for a byte-oriented memory) is applied.

$$\Uparrow (w01010101, r01010101, \ldots, w00001111, r00001111)$$

Finally, the influence of address decoder faults is excluded by applying the appropriate March sequences combined to form the $5N$ MATS+ March test:

$$\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)$$

The length of the complete memory test and diagnosis sequence is $(22+6\log_2 w)N+1$.

4 BIST Implementation

The proposed diagnostic tests can be realized in hardware as an addition to common BIST logic. To test the embedded RAM, internal signals, such as Data and Address lines and Read/Write signals, have to be multiplexed with the test signals. Similar to common BIST realizations [5], a global test controller controls the application of the different (March) test phases, the address generation by, e.g., a linear-feedback shift register (LFSR), and application of the data background patterns and control signals such as R/W. In Figure 1, the BIST circuitry is shown. Note that those parts of the circuitry that are specific to the proposed diagnostic approach are marked with a shaded background. Favorably, the proposed BIST logic does not lead to any

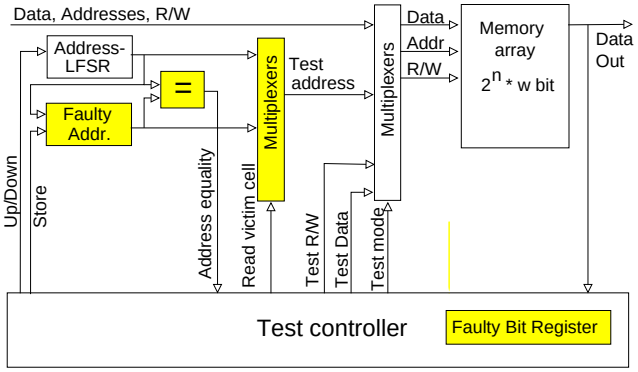

Figure 1: Architecture of the proposed Test and Diagnosis BIST-logic.

performance penalty of the memory, compared to standard BIST logic for embedded memories.

During test application, when faulty memory words are detected, the respective addresses have to be stored in registers, namely the *faulty address registers*, since test application is not interrupted every time a fault is detected. In the basic approach, the number of registers is restricted to one, and the *faulty address register* is loaded with the address of the first detected fault. To diagnose the result of the first $12N$ test, the failure pattern of the fault (see Table 1) has to be examined by the BIST circuitry. Therefore, during diagnosis, the faulty bit in the memory word has to be compared at every read access to the word containing the faulty bit. This is accomplished by a small extension of the controller logic.

5 Simulation Results

To numerically evaluate the performance of the diagnostic routines, a simulator representing a 1024×8-bit RAM has been programmed in *C* language. For each type of fault in the fault model, 1000 trials were performed in which single faults were injected at random addresses. Memory array faults, such as SAF's, TF's, and CF's were detected completely and diagnosed correctly. As expected, intra-word coupling faults showed the possibility of misdiagnosis as SAFs. However, the application of the intra-word test allows differentiation of the two fault types.

A second set of 1000 simulated trials was used to determine the diagnostic capability of the proposed approach in case of multiple faults. Ideally, the first fault is diagnosed, and the diagnosis is not affected by additional faults that occur. Indeed, in all cases, the first fault detected was diagnosed correctly. The inclusion of pattern-sensitive faults did not cause any problems.

Linked faults, defined as two faults affecting the same coupled cell, have also been simulated. If one of the two faults is a single-cell fault, it masks the behavior of the other fault and is detected and diagnosed. If two linked CF's are

Table 3: Diagnosis results for the case of multiple linked faults.

Linked Faults		Detected	Diagnosed as				Correct
			CFid	CFin	PSF	Addr.	diagnoses
CFid	CFid	1000	**183**	0	817	0	18.3%
CFid	CFin	1000	**171**	**809**	20	0	98.0%
CFid	Static PSF	1000	**705**	3	**292**	0	99.7%
CFid	Passive PSF	1000	**759**	0	**241**	0	100%
CFid	Active PSF	1000	**951**	0	**49**	0	100%
CFid	Address	1000	**907**	0	0	**93**	100%
CFin	CFin	694	29	**279**	274	112	27.9%
CFin	Static PSF	1000	32	**863**	**104**	1	96.7%
CFin	Passive PSF	1000	2	**943**	**55**	0	99.8%
CFin	Active PSF	1000	5	**965**	**30**	0	99.5%
CFin	Address	1000	186	**813**	0	**1**	81.4%

present, these are detected as a result of the identical addressing order of $\Uparrow (r0, w1, r1, w0)$ and $\Uparrow (r1, w0, r0, w1)$ in the first test. With the exception of CFin linked with CFin, all linked coupling faults are detected, and most faults are diagnosed correctly, i.e., all cells involved are located. The diagnosis results are shown in Table 3. Note that the percentage of correct diagnoses is based on the total number of injected faults. Two linked faults can easily give a pattern of incorrect reads not matching any single fault, resulting in diagnosis of a PSF. For example, two CFid's are likely to be diagnosed as a single PSF. If a CF is linked to an existing (i.e., injected) PSF, it is likely to be correctly diagnosed, since the PSF is mostly inactive. A small percentage of the CF's were misdiagnosed in the presence of a linked AF, leading to a false diagnosis of the type of CF, even though the aggressor and victim cells are localized correctly.

6 Conclusions

In this work, a set of effective diagnostic memory tests of linear order $O(N)$ is proposed for word-oriented memories, and diagnostic capabilities of the tests are analyzed. The test application allows single-cell faults to be distinguished from multiple-cell faults and all faults of the fault model to be distinguished from each other (e.g., CFid $<\uparrow/1>$ from CFid $<\downarrow/1>$.) All cells involved in the targeted faults are detected and localized, using supplementary minimal-area BIST hardware that compares favorably with other BIST designs. Even for the case of multiple linked faults , the diagnostic resolution of the approach is close to 100% for most combinations of linked faults. The proposed diagnostic approach enables economical use of spare resources for automatic on-chip reconfiguration with both unlinked and linked memory faults.

References

[1] R. David and A. Fuentes, "Fault Diagnosis of RAM's from Random Testing Experiments," *IEEE Trans. Comp.*, vol. 39, no. 2, pp. 220-229, 1990.

[2] R. Treuer and V. K. Agarwal, "Built-In Self-Diagnosis for Repairable Embedded RAMs," *IEEE Design and Test of Computers*, vol. 10, no. 2, pp. 24-33, 1993.

[3] V. N. Yarmolik, Y. V. Klimets, and A. J. van de Goor, "Diagnostic RAM Tests," *Automatic Control and Computer Sc.*, vol. 31, no. 2, pp. 11-16, 1997.

[4] L. Shen and B. Cockburn, "An Optimal March Test for Locating Faults in DRAMs," *Rec. IEEE Int. Workshop on Memory Testing*, pp. 61-66, 1993.

[5] R. Rajsuman, "RAMBIST Builder: A Methodology for Automatic Built-In Self-Test Design of Embedded RAMs,"*Rec. IEEE Int. Workshop on Memory Technology, Design and Testing*, pp. 50-56, 1996.

[6] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, J. Wiley & Sons, 1991.

[7] A. J. van de Goor and I. B. S. Tlili, "March Tests for Word-Oriented Memories," *Proc. Design Automation and Test in Europe Conf.*, pp. 501-508, 1998.

[8] M. Sachdev, "Open Defects in CMOS RAM Address Decoders," *IEEE Design and Test of Computers*, vol. 14, no. 2, pp. 26-33, 1997.

[9] J. Otterstedt, D. Niggemeyer, and T. W. Williams, "Detection of CMOS Address Decoder Open Faults with March and Pseudo Random Memory Tests," *Proc. Int. Test Conf.*, pp. 53-62, 1998.

[10] V. G. Mikitjuk, V. N. Yarmolik, and A. J. van de Goor, "RAM Testing Algorithms for Detection Multiple Linked Faults," *Proc. European Design and Test Conf.*, pp. 435-439, 1996.