

# Transparent Memory BIST\*

M. G. Karpovsky

Dep. Elec., Comp. & Syst. Eng.  
College of Eng., Boston University  
Boston, MA 02215, USA

V. N. Yarmolik

Computer Science Dep.  
Minsk Radio Eng. Inst.,  
Minsk, 220027, BELARUS

**Abstract:** *This paper presents a new methodology for testing of bit-oriented and word-oriented RAMs based on circular test sequences, which can be used for periodic and manufacturing testing and require lower hardware and time overheads than the standard approaches. The proposed approach is a transparent BIST technique combined with on-line error detection, which preserves the initial contents of the memory after the test and provides a high fault coverage for traditional fault and error models, as well as for pattern sensitive faults. Our methodology is useful for embedded RAMs and MCM implemented RAMs.*

**Key words**—Memory testing, random access memory, twisted ring (Johnson) counter, built-in self-test, transparent BIST, signature analysis.

## 1 Introduction

Advances in manufacturing technology allow larger and denser circuits to be placed on a single chip, especially when these circuits can be realized as regular/cellular structures. Random Access Memories (RAMs) fall into this category and memory chips are by far the most dense circuits. The important problem for such high density circuits is their testing. There are two main approaches for memory testing: [1] off-line testing, which is based mainly on deterministic RAMs tests, and [2] on-line testing using error-correction codes.

Traditionally, as the first step in development of a test for off-line RAM testing, a fault model representing physical defects is constructed [1, 3].

On-line testing is the main tool that ensures high reliability of storage data during operational life time by employing on-chip error-detecting/correcting circuits.

We will present in this paper a methodology for RAMs BIST based on circular test sequences, which can be used for periodic and manufacturing testing.

The paper is organized as follows. In the next section fault models are investigated. In section 3 and 4 transparent circular testing strategy is discussed. In section 5, the efficiency of the proposed approach is analyzed. In section 6 hardware implementations are presented. The results are summarized in section 7.

## 2 Fault Model

As a realistic and general fault model we propose the  $l$  out of  $n$  pattern sensitive fault ( $PS(n, l)$ ), where  $n < N$  is a block (region, row or column of memory array) size for memory with the size  $N$ . According to this model the contents of any memory cell which belongs to an  $n$ -bit memory block, or ability to change the contents, is influenced by the contents of any  $l - 1$  cells in the same block. These contents consists of a pattern of 0s and 1s or changes in these contents.

Depending on the values of  $l$  and  $n$   $PS(n, l)$  generalizes the following cases considered in the literature:

1. For  $l = n = N$ ,  $PS(n, l)$  represents pattern sensitive faults (PSF) [1], when the contents of a cell, or the ability to change the contents, is influenced by the contents of all other cells in the memory.

2. For  $l < n$  and  $n = N$  we have the case of the general  $l$ -coupling faults, when the base cell is influenced by a group of  $l - 1$  cells which can be placed anywhere in the memory.

3. For  $l = n < N$   $PS(n, l)$  is reduced to NPSFs, where the base cell is influenced by the  $l - 1$  cells, which take on only a single position.

4. The case when  $l < n$  and  $n < N$  or  $n \ll N$  is more common and realistic one, when the contents of any memory cell belongs to an  $n$ -bit memory block, or ability to change the contents, is influenced by the contents of any  $l - 1$  cells of the same block.

In the following sections we propose a methodology for memory testing based on the  $PS(n, l)$  model. These methodology is based on circular test sequences, which can be used for transparent periodic and manufacturing testing and require lower overheads.

\*This work was supported by the NSF under Grant MIP92-08487 and the NATO under Grant 910411.

### 3 BIST RAM Strategy

Efficient techniques for transparent BIST testing were proposed by B.Koeneman and M.Nicolaidis [4]. These techniques can be applied to any random access memory test patterns for manufacturing and periodic testing. BIST implementations for the transparent memory testing allow to reproduce only one memory test and as the rule they have the complexity of  $O(N)$  [4]. Memory testing sessions with the fixed test sequences may be not very efficient for manufacturing testing.

In this paper we present a new approach for transparent memory testing which preserves the initial memories content after a test.

The method consists of the simulation of the twisted ring counter (Johnson counter) operation on a block of memory-under-test and it has a simple hardware implementation.

Any test session consists of  $T$ , ( $T \leq W$ ) subsessions for the memory with the size  $N = W \times m$  bits, with  $W$  words and  $m$  bits per word. The implementation of a subsession requires the following three stages.

1. Compute the signature of the initial state for the block of the memory-under-test data.
2. The block with  $n$  memory words by  $m$  bits operates  $2n$  clocks as  $m$  parallel twisted ring counters.
3. Compute the signature of the new state of the memory block after stage 2 and compare it with the signature computed at stage 1.

The faults manifesting themselves during intermediate *Read* and *Write* operations for the above procedure would be detectable by the observation of distortions in final memory contents.

For relatively small  $n$  and  $m$ , as well as for a RAM with built-in error-detection/correction circuit, stages 1 and 3 can be avoided. In this case, a nonzero output (syndrome) of the error-detection/correction circuit indicates the presence of a fault.

Any test session consists of  $T = T(n, \Delta)$  subsessions with length  $n$  overlapping by  $\Delta$  words (see Fig. 1), where  $T = T(n, \Delta)$  is determined by the equation

$$T(n, \Delta) = \begin{cases} \lceil \frac{W}{n-\Delta} \rceil + 1, & 2\Delta + [W - \lfloor \frac{W}{n-\Delta} \rfloor](n - \Delta) < n; \\ \lceil \frac{W}{n-\Delta} \rceil, & 2\Delta + [W - \lfloor \frac{W}{n-\Delta} \rfloor](n - \Delta) \geq n, \end{cases} \quad (1)$$

where  $0 \leq \Delta < n$  and for  $n \ll W$  we will get  $T(n, \Delta) \approx \frac{W}{n-\Delta}$ .

If a fault occurs, it is important to get the full information about its location. For circular test patterns

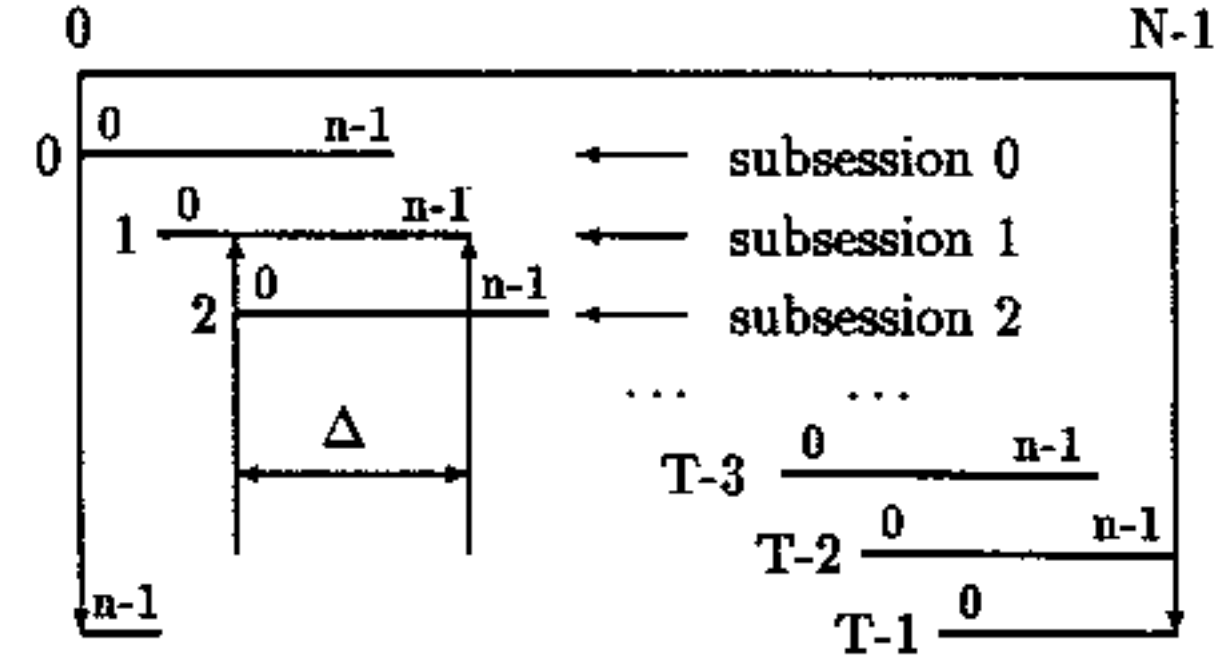


Figure 1: The timing diagram for the self-test procedure

any stuck-at fault will be manifested by a distortion of contents of several  $n$ -bit blocks, which allows to locate a block of memory containing faulty cells. To measure a diagnostic capability of the test let us introduce the resolution function which in our case has the following form

$$R(n, \Delta) = \max \begin{cases} \lfloor \frac{n}{n-\Delta} \rfloor (n - \Delta) - \Delta, \\ n - \lfloor \frac{n}{n-\Delta} \rfloor (n - \Delta). \end{cases} \quad (2)$$

From (2) we have  $(n - \Delta)/2 \leq R(n, \Delta) \leq n - \Delta$ .

Small values of  $\Delta$  can reduce a complexity, but at the same time reduce of the test resolution. The best resolution can be achieved for  $\Delta = n - 1$ , and in this case we can determine the faulty memory cell. It should be mentioned that for faulty modes the data within  $n$  memory cells may be destroyed after the test.

### 4 Transparent Circular Test Patterns

Test patterns for transparent testing should have the circular property to ensure recovery of a memory content. For the fault-free case at the end of a test session a memory content will have the same value as before the session, and for a faulty memory a content at the end of the test session will be different from the initial content.

Twisted ring counters of various types have been used for many years for creating well-defined periodic pulse patterns [5]. A twisted ring counter is a circular shift register with the complemented output of the last flip-flop connected to the input of the first flip-flop.

It is easy to show that the twisted ring counter with  $n$  stages has the cycle with the constant length  $2n$  for any initial state iff  $n = 2^k$ ,  $k = 0, 1, 2, \dots$

Thus, in this case  $2^n$  elements of the space are divided into

$$Q = \frac{2^n}{2n} = \frac{2^{2^k}}{2 \cdot 2^k} = 2^{2^k - k - 1} \quad (3)$$

orbits with  $2^{k+1}$  elements in any orbit. If  $n \neq 2^k$ , then the space of  $2^n$  elements consists of orbits of various lengths.

The following simple procedure can be used to model the twisted ring counter operation over an  $n$ -bit block of a  $W \times 1$  memory-under-test. Here  $D_0$  and  $D_1$  are the first and a second stage of an additional two stage shift register  $SR$ ;  $n = 2^k$  is the memory-under-test block size;  $[a_i]$  - the contents of the  $i$ th memory cell.

#### Procedure TRC

```

Input:  Memories block size  $n$ ;
        Starting address  $w$ ;
Read:    $[a_{w+n-1}]$  to  $D_0$ 
Shift:   $SR$ ;
Loop1:  For  $j = 1$  to  $2n$  Do
Read:    $[a_w]$  to  $D_0$ ;
Write:   $[D_1] \oplus 1$  to  $a_w$ ;
Shift:   $SR$ ;
Loop2:  For  $i = 1$  to  $n - 1$  Do
Read:    $[a_{w+i}]$  to  $D_0$ ;
Write:   $[D_1]$  to  $a_{w+i}$ ;
Shift:   $SR$ ;
End:    Loop2
Write:   $[a_{w+n-1}]$  to  $D_0$ 
Shift:   $SR$ ;
End:    Loop1
End:

```

In the above procedure, the *Write* and *Shift* operations in Loop1 and Loop2 can be accomplished simultaneously and do not need more than one control signal. It should be mentioned that we can start this procedure from any address  $w \in \{0, 1, 2, \dots, W - 1\}$ .

There are direct and inverse TRC procedure with opposite direction of simulated shifts of the twisted ring counter.

The complexity of the application Procedure TRC to a block with  $n$  cells is

$$L_{TRC} = 4n^2 + 2n + 2 \approx 4n^2 + 2n, \quad (4)$$

where  $n$  is the length of the RAM implemented twisted ring counter.

Procedure TRC can be easily extended to the case of the word-oriented memory with the size  $W \times m$ , with the same test complexity determined by (4).

## 5 Analysis of Fault Coverages

It is easy to show, that for any orbit (initial state) and for any  $l \leq n = 2^k$ , ( $k = 0, 1, 2, \dots$ ) the twisted ring counter algorithm provides for at least  $2l$  different binary codes at any  $l$  positions. Let us introduce the function  $f(l)$ , which represents the number of different binary test patterns out of  $2^l$  possible patterns at any  $l \leq n$  randomly chosen cells. Then,

$$f(l) \geq 2l. \quad (5)$$

To simplify interpretations in the following discussion,  $f(l)$  is normalized with respect to the size of the exhaustive test for  $l$  cells,

$$P(l) = \frac{f(l)}{2^l} \geq \frac{2 \cdot l}{2^l} = l \cdot 2^{-l+1}. \quad (6)$$

This function is the measure of a test quality for one test session. Then,  $P(l)$  is different for different orbits (initial states).

From (5) the test has good fault coverages for  $PS(n, l)$  faults for small  $l$ , and nonzero but decreasing coverages for growing  $l$ . It should be mentioned that  $f(l)$  presents the number of test patterns which have been achieved during only one test session according to the original twisted ring counter algorithm.

One can see that stuck-at fault  $a_i \equiv 0$  is not detectable iff the initial state  $a_0 a_1 \dots a_{i-1} a_i a_{i+1} \dots a_{n-1}$  of  $n$  memories cells is  $11 \dots 100 \dots 0$ . For any initial state, which is different from  $11 \dots 100 \dots 0$   $a_i \equiv 0$  is detectable by the direct TRC procedure, as well as for any state does not equal to  $00 \dots 001 \dots 1$  by the inverse TRC procedure.

As a measure of detectability for stuck-at faults by the direct or inverse TRC, let us introduce the probability  $P_s(n)$  to detect any given stuck-at fault within  $s$  test sessions. (We assume that all of the RAMs initial states are equally likely to occur.) Then,

$$P_s(n) = 1 - 2^{-sn}. \quad (7)$$

Every single stuck-at fault cannot be detected during one session of TRC procedure (direct or inverse) for only one initial state of  $n$  cells. For every initial state there is a set of undetectable stuck-at faults. If an initial state is equal, for example, to 0000 there are one single ( $a_0 \equiv 0$ ), three double ( $(a_0 \equiv 0, a_1 \equiv 0)$ ,  $(a_0 \equiv 0, a_2 \equiv 0)$  and  $(a_0 \equiv 0, a_3 \equiv 0)$ ), three triple and one fault with multiplicity four which are undetectable. At the same time, for initial state 0101 the set of undetectable faults contains only two faults i.e.  $(a_1 \equiv 1, a_2 \equiv 0, a_3 \equiv 1)$  and  $(a_0 \equiv 0, a_1 \equiv 1, a_2 \equiv 0, a_3 \equiv 1)$ .

It should be mentioned, that if we use both direct and inverse TRC procedures all stuck-at faults with the multiplicity less than  $n$  are detectable.

Any coupling fault from a cell  $i$  to a cell  $j$  can be detected when the cells contain a particular pair of binary values  $a_i$  and  $a_j$  and  $\bar{a}_i$  is written into cell  $i$ . Then cell  $j$  changes its state  $a_j$  to the erroneous state  $\bar{a}_j$ , which ensures a manifestation of the coupling fault. For the manifestation of all coupling faults during one test session four different states 010, 011, 100, 101 for the cells  $i-1, i$ , and  $j$  should be generated for any  $i$  and  $j$ . For  $n = 8$  the probability of this event, for randomly selected equiprobable initial states, is  $1 - 1/16 = 0.9375$ . As follows from procedure TRC there is no masking for the coupling faults when  $j$  does not equal to  $n - 1$  for direct TRC procedure and to 0 for inverse.

If we use both direct and inverse TRC procedures then for any  $i$  and  $j$  of all four different states will be generated, i.e. all 2-coupling faults will be detected.

For the general case, the self-testing session consists of several subsessions, overlapped by  $\Delta$  bits, where the same memory cells are tested during more than one subsession. Thus, the above estimation (7), obtained for  $\Delta = 0$ , can be considered as a lower bound for the general case ( $\Delta \neq 0$ ). To generalize (7) it is easy to show that the probability  $P_s(n, \Delta)$  to detect any given stuck-at fault within  $s$  sessions is given by

$$P_s(n, \Delta) = 1 - 2^{-sz}, \quad (8)$$

where  $z = \lfloor \frac{n}{n-\Delta} \rfloor (n - \Delta) + \Delta$ , and  $n \leq z \leq 2n - 1$ .

The approach provides for a high fault coverage for pattern-sensitive faults  $PS(n, l)$  due to a high probability,  $P(l)$ , to generate a large portion of all possible test patterns of length  $l \leq n$  within any  $l$  cells.

The average values of  $P(l)$  for different  $n$  and  $l$  are shown in Table 1.

(All experimental results in Table 1 have been obtained for  $10^5$  randomly chosen initial states of the block-under-test.)

One can see from Table 1 that  $P(l)$  is very close to 1 for  $l \leq \log_2 n$ . We note that  $P(l)$  does not depend on selection of  $l$  positions within a block of  $n$  cells.

Thus, the deterministic circular test TRC has a good fault coverage for pattern sensitive faults for a small size,  $l$ , of the neighborhood and a nonzero but decreasing coverage for growing  $l$ .

The analysis of effectiveness of fault detection shows that for  $n \geq 16$  all simple (not pattern sensitive) faults will be detectable with probabilities very close to 1. The approach provides for a high fault coverage for pattern-sensitive faults  $PS(n, l)$  due to

Table 1: Probability  $P(l)$  for deterministic tests (TRC) based on simulation of a twisted ring counter on a  $n$ -bit block-under-test

$n$	$l=2$	$l=4$	$l=6$	$l=8$	$l=10$	$l=12$
4	1.00	0.687	-	-	-	-
8	1.00	0.893	0.524	0.216	-	-
16	1.00	0.989	0.790	0.393	0.143	0.045
32	1.00	0.999	0.954	0.637	0.267	0.089
64	1.00	1.000	0.997	0.862	0.464	0.168

a high probability to generate a large portion of all possible test patterns within any  $l$  memory cells.

Transparent testing of RAMs based on simulation of twisted ring counters on blocks of  $n$  cells has a simple hardware implementation and a low test complexity, but a number,  $f(l)$ , of different combinations which are generated at any  $l$  bits in a block for this approach depends on the initial state of the cells in the block. In the worst case for one test session this number is equal to  $2l$ , which may be not sufficient for detection of  $PS(n, l)$  faults. To overcome this difficulty one can simulate LFSRs [6] instead of twisted ring counters on blocks with  $n$  cells. This approach provides for any  $f(l)$ , ( $f(l) \leq 2^l$ ) but it will require an increase in a testing time.

## 6 Hardware Implementation

The procedure for simulation of the twisted ring counter over  $n$  bit(word) block of the memory-under-test, has a simple hardware implementation. It takes two( $2m$ ) flip-flops and one( $m$ ) Exclusive-OR gates for the test pattern generator, one( $m$ ) multiplexers for mode control and one signature analyzer per memory unit to implement Procedure TRC. For a small  $n$  and  $m = 1$  the signature analyzer is an  $n$ -bit shift register and for the general case it consists of an  $r \leq 32$  bit signature analyzer, a  $r$ -bit register and a comparator. In many cases one can take  $r = m$ . Details of these implementations are described in following subsections.

### 6.1 Bit-Oriented RAM

Hardware implementation for a bit-oriented RAM with transparent testing is shown in Fig 2. The memory contains  $W$  words of length 1 ( $N = W \times 1$ ). For this case the transparent self-testing RAM consists of the memory array, the control unit, the test pattern generator (TPG), the multiplexer (Mux), the signature analyzer (SA) and the XOR gate for response

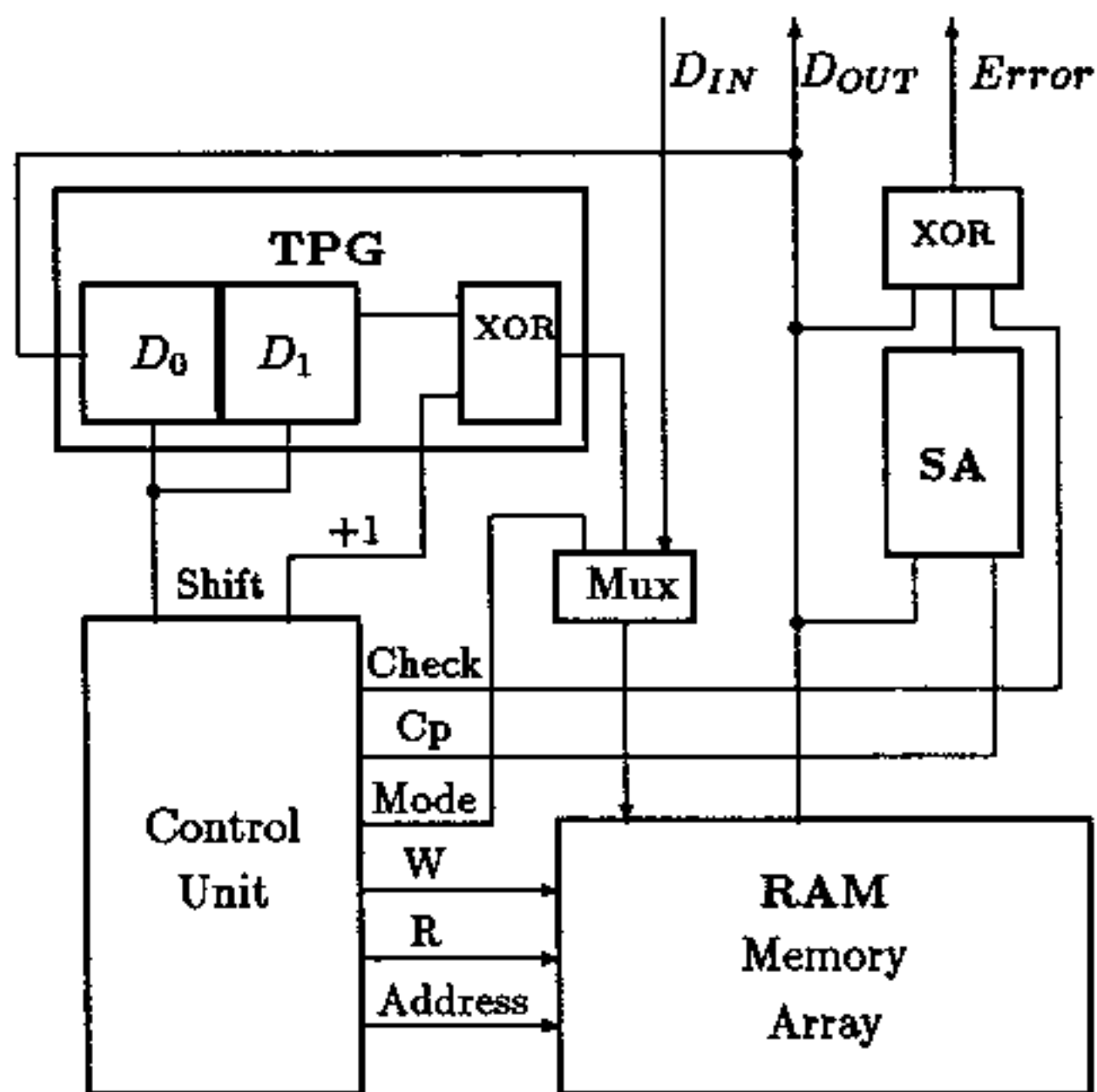


Figure 2: State diagram of the transparent self-testing RAM

evaluation. The RAM has one data input line  $D_{IN}$  and one data output line  $D_{OUT}$ . The control unit generates *Write* ( $W$ ) and *Read* ( $R$ ) control signals, addresses, *Mode* signal to change the mode, the clock pulses ( $Cp$ ) for SA, control signal *Check* for comparator and two types of signals for TPG i.e., *Shift* and  $+1$ . The RAM in Fig. 1 may operate in the normal computing mode and in the transparent self-testing mode.

Transparent periodic testing of the RAM of Fig 2 is divided into  $T(n, \Delta)$  subsessions, where  $n \ll N$  is the size of a RAM's block used for one subsession. Every subsession consists of the following stages.

1. Compute the signature for an initial state of the  $n$  memories cells. For small  $n$  (say  $n < 32$ ) cell's contents are written into a shift register (SA) (see Fig. 2) and for larger  $n$  SA is replaced by a primitive  $r$ -bit LFSR. This stage requires  $n$  memory cycles.
2. The  $n$  memory cells operate as the twisted ring counter. The TPG generates test patterns by simulation of Procedure TRC. The complexity  $L_{TRC}$  of this stage is equal to  $4n^2 + 2n + 2 \approx 4n^2 + 2n$  memory cycles (see equation 4).
3. Compute the signature of a new state of the  $n$  memory cells and compare it with computed at

Table 2: Test complexities (in seconds) for a  $10^6 \times 1$  RAM

$\Delta$	$n$			
	16	64	256	1024
1	7.25	26.4	103	410
7	12.0	28.9	105	412
15	108	33.6	108	415
31		50.0	116	422
63		$1.6 \times 10^3$	135	436
127			203	467
255			$2.6 \times 10^4$	545
511				817

stage 1. For the RAM shown in Fig. 2 this stage requires the comparison of the contents of the memory with the contents of SA (see Fig. 2). This stage requires  $n$  memory cycles.

For the complexity  $L(n, \Delta)$  of one test session we have

$$L(n, \Delta) = (2n + L_{TRC})T(n, \Delta) \approx 4n(n+1) \left[ \frac{W}{n-\Delta} \right], \quad (9)$$

where  $L_{TRC}$  is the complexity of Procedure TRC. The complexity  $L(n, \Delta)$  is  $O(N)$  for  $n \ll W = N$  and  $O(N^2)$  for  $n = W$  and  $T = 1$ , ( $\Delta = 0$ ) as follows from (9). There is a possibility to generate the test with the complexity  $O(N^3)$ , for the case when  $n = W$  and using Procedure TRC for all addresses  $w \in \{0, 1, 2, \dots, W-1\}$ , with  $\Delta = n-1$ .

As we can see the test complexity for the circular test depends on  $n, W$  and overlapping  $\Delta$  between two consecutive subsessions. Let us consider the following three cases: 1.  $\Delta = 0$ , then  $L(n, 0) = 4(n+1)W$ . 2.  $\Delta = n/2$ , then  $L(n, n/2) = 8(n+1)W$ . 3.  $\Delta = n-1$ , then  $L(n, n-1) = 4n(n+1)W$ .

In Table 2 we present the time in seconds required for the proposed test for  $N = W \times 1 = 10^6$  and access time to the RAM  $100ns$ , for different  $n$  and  $\Delta$ .

## 6.2 Word-Oriented RAM

A self-testing procedure for a word-oriented ( $W \times m$ ) RAM, as in the previous case requires  $T(n, \Delta)$  subsessions and test complexity  $L(n, \Delta)$  is determined by (9). The only difference is that during stages 1 and 2 the compaction of the contents of  $n$  memory words is implemented by the MISR [6].

The efficiency of the transparent circular test for a word-oriented RAM depends on the initial states of

all  $n$  memory words of the block-under-test. For the worst case, when the contents of  $n$  words are equal, we will get only  $2n$  different test patterns during one sub-session. If data in any word is different from the data stored in another word and its negation, then the number of test patterns is equal to  $2n^2$ . If we assume, that all possible memory contents are equally likely to occur, then the probability  $P\{NTP = 2n^2\}$ , that the number of test patterns (NTP) is equal to  $2n^2$  is determined by

$$P\{NTP = 2n^2\} = \prod_{i=1}^{n-1} (1 - i2^{m-1}) \geq (1 - n2^{1-m})^{n/2},$$

and for  $n \ll 2^m$

$$P\{NTP = 2n^2\} \geq 1 - n^2 2^{-m}. \quad (10)$$

For  $n = 32$  and  $m = 4$  we have  $P\{NTP = 2n^2\} = 1 - 2^{-19}$ .

The hardware implementation requires an additional  $m$ -bit MISR,  $m$ -bit register, a comparator and two  $m$ -bit registers with  $m$  Exclusive-Or gates for a TPG. The relative overhead, decreases as a size of RAM-under-test increases.

### 6.3 RAM with On-line Testing

An important feature of our approach is the ability to combine off-line and on-line testing. If for any codeword  $X$ , which belongs to the code used for on-line error detection, the negation of the word  $\bar{X}$  also belongs to the same code, then there is the possibility to explore the on-line testing circuits (syndrome evaluators) for off-line testing with circular test patterns, based on twisted ring counter sequences. For the case of parity check codes and even  $m$  our approach has a simple implementation. During the off-line test mode the presented design implements the transformation of the memory word contents for the fault-free case within the parity code codewords.

For a faulty RAM, a memory word will be changed to a noncode word. A manifestation of any fault through the cyclic test pattern application can be detected in the *Read* mode by the on-line parity check circuit.

Thus, if the RAM has on-line test circuits, it can be used for the off-line transparent self-testing procedure and which results in reduction of hardware and time overheads for BIST. In this case we have to generate the circular test patterns for all columns of the RAM.

We note that for the RAM with on-line testing capability we do not need an  $m$ -bit MISR,  $m$ -bit register and comparator, which simplifies the hardware implementation.

Test complexity  $L(n, \Delta)$  will be determined by the complexity  $L_{TRC}$  of Procedure TRC and a number  $T = (n, \Delta)$  of test sub-sessions. It has the following form

$$L(n, \Delta) = L_{TRC} T(n, \Delta) \approx 2n(2n + 1) \left[ \frac{W}{n - \Delta} \right].$$

## 7 Conclusions

In this paper we have presented an efficient transparent testing approach based on circular test patterns generated by twisted ring counters, which allows to combine on-line and off-line testing. It was shown that the test sequences have good fault coverages for simple faults and for pattern sensitive faults. The proposed approach provides for preservation of initial contents of the memory, which is useful for periodic testing. For the case of manufacturing testing the approach allows to generate the tests with the complexities from  $O(N)$  up to  $O(N^2)$ . The proposed approach can be used for manufacturing and periodic testing of RAMs.

## Acknowledgment

The authors wish to thank Prof. Lev B. Levitin of the Boston University, Boston, for his valuable suggestions.

## References

- [1] Goor A.J., Verruijt C.A. "An overview of deterministic functional RAM chip testing", *ACM Computing Surveys*, Vol.22, N1, March, 1990.
- [2] Mazumder P. "An On-chip Double-bit Error-correcting Code for Three-dimensional Dynamic Random-access Memory", *Proceedings International Test Conference*, Washington, September 1988, pp.279-288.
- [3] Goor A.J. "Using March Tests to Test SRAMs", *IEEE Design and Test of Computer*, Vol.10, N1, March 1993, pp.8-14.
- [4] Nicolaidis M. "Transparent BIST for RAMs", *Proceedings International Test Conference*, Baltimore, September 1992, pp. 598-607.
- [5] Bleickardt W. "Multimoding and Its Suppression in Twisted Ring Counters", *The Bell System Technical Journal*, November, 1968, pp.2029-2050.
- [6] Bardell P.H., McAnney W.H., Savir J. "Built-in Test for VLSI: Pseudorandom Techniques", John Wiley and Sons, New York, 1987.