5th International Conference on VLSI and CAD
October 13 - 15, 1997, Seoul, Korea

BUILT-IN SELF TEST FOR HIGH DENSITY SRAMS *

Jong-Cheol Lee, Yong-Seok Kang and Sungho Kang
Dept. of Electrical Engineering, Yonsei University
Shinchon-Dong 134, Seodaemoon-Gu, Seoul, Korea
E-mail: shkang@bubble.yonsei.ac.kr, Phone: 82-2-361-2775, Fax: 82-2-313-8053

Abstract

The paper shows a march algorithm which performed concurrently in each basic marching block which makes up whole memory cell array. The new parallel access method is very efficient in speed and a very tiny hardware overhead for BIST circuitry is required.

1 Introduction

With the progress in IC fabrication technology and the growing need for large memories, the density of memory circuits has increased rapidly. The test cost and the test time for semiconductor memories have grown significantly along with the growing density of memory chips. Several innovative test algorithms for RAMs have been developed. In these algorithms, either whole memory is read after changing the value of one cell or only the cell whose value is changed is read. And many researches about built-in self test(BIST) of RAMs based on the march algorithms have been reported [1,2]. However, each march test has a test complexity of $\mathcal{O}(n)$ for an n-bit RAM and requires a very large test time. As a result, the conventional march tests are no longer adequate for the large memories. Thus the technique of parallel testing has been proposed as an alternative to overcome a very large test time required for testing high density RAMs.

In this paper, a new SRAM BIST architecture with parallel testing technique has been designed, and a new test algorithm based on marching has been developed.

2 Test Methodology

The faults in SRAMs often occur in the memory cell array, the address decoder, and the read/write logic [3]. The major fault models in the memory cell array of SRAM can be categorized as follows: *stuck-at faults*(SAFs), *transition faults*(TFs) and *coupling faults*(CFs) [4]. *Address decoder faults*(AFs) may occur in the address decoder. In read/write logic, some output lines of the sense amplifier logic or write driver logic may be stuck-at 0 or stuck-at 1. In either case, this fault can be considered as the same as stuck-at 0 or stuck-at 1 fault in the memory cell array [2,3].

To detect these faults in memories, a lot of algorithms were developed and is still used in the industry. One of the most popular algorithm is the March tests. But with the conventional March test algorithms which have the complexity of $\mathcal{O}(n)$, it takes much time to test large size memories. In order to reduce the test time, multiple cells should be tested at a time by modifing the address decoder. With the modified decoder, a write operation results in parallel writing a logic value to all the locations selected by the decoder. In read mode, only one word line and multiple bit lines are selected, and the contents of the cells accessed by the selected word line and bit lines are read in parallel. The parallel comparator, which does not require a prior known reference value, is used to determine whether all the accessed cells have the identical content. If due to some fault a write operation on a cell fails or the contents of some cells change, all the inputs to the comparator are not identical. The comparator detects this anomalous input and indicates the occurrence of a fault. The modified address decoder has two input words; a normal address word and a mask address word. The decoder can be modified as in [5]. In this way, the mask address word allows one to access multiple cells at a time.

A parallel comparator is shown in Figure 1. The parallel comparator monitors the outputs of the sense amplifiers of the selected cells by comparing the output values. When any selected bit line has a different value from the other bit lines in the selected group, the output of the circuit will be 0 indicating the occurrence of a fault. In the normal mode of operation the comparator is isolated from the sense amplifiers.

*This research was supported by Samsung Electronics Co., Ltd.

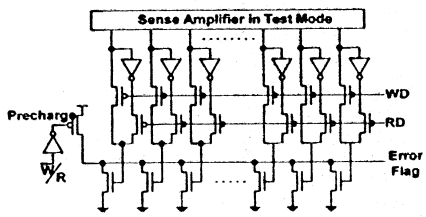

Figure 1: Parallel Comparator

3 Test Algorithm

Suppose that the memory cell array consists of $\sqrt{n}$ bit-lines and $\sqrt{n}$ word-lines such that the memory cells are organized into a $\sqrt{n} \times \sqrt{n}$ two-dimensional array. The memory plane can be thought of as a set of small blocks. Let the block consist of $\sqrt{k} \times \sqrt{k}$, where $0 < k < n$. With the mask address decoder the cells in the same position in each block can be accessed and tested simultaneously. Then the test time can be reduced to the time taken for testing a block. Let this block be a *basic marching block*. The size of a basic marching block is decided according to the fault coverage, test time, and the hardware overhead.

March C- algorithm shown in Fig. 2 is adopted to the proposed parallel BIST scheme. March C- requires $10 \times n$ operations and detects all SAFs, all AFs, all TFs and all CFs [4,6]. With the parallel test scheme, the test time can be reduced.

⇑(w0);	⇑(r0,w1);	⇑(r1,w0);	⇓(r0,w1);	⇓(r1,w0);	⇓(r0)
M_1	M_2	M_3	M_4	M_5	M_6

Figure 2: March C- Algorithm

In test mode, the write operation of march element M_1 can be carried out by increasing the normal address with the mask address fixed; all the mask addresses are 1's. The cells at the same position in each block are written simultaneously with the value 0. Thus the write operation required for march element M_1 is the same as the number of cells in the basic marching block. The number of operations required for the march elements M_2, M_3, M_4 and M_5 can be calculated easily because they require the identical number of operations. The read operation requires more accesses to the memory arrays. Note that in this test scheme, only the cells sharing the same word line can be read simultaneously. Thus the row normal addresses are not masked and the number of read operations are several times the number of write operations. The operations for the march element M_6 can be thought of as same as the read operation in the march element M_2. The total number of operations required to test a $\sqrt{n} \times \sqrt{n}$ memory which consists of $\sqrt{k} \times \sqrt{k}$ basic marching blocks is calculated as $5 \times \sqrt{k} \times \left(\sqrt{k} + \sqrt{n}\right)$. The complexity would be varied by the variable k.

The variable k also affects the fault coverage. The SAFs and TFs are 100% detected but in case of the AFs and CFs the fault coverage is slightly lower than that of the conventional March C- algorithm. This problem arises because the coupling between the cells which are in the same position in each block. Therefore the fault coverage is reduced to 100(1-1/*2*)%. However if the variable k is large enough then the fault coverage would be above 99.9%. For example, consider a 4M bit memory consists of 64k bit basic marching blocks. Then the fault coverage will be 99.998%.

4 BIST Architecture

The implemented BIST architectures to perform March C- in parallel should generate the appropriate address sequences and data to be read or written. To distinguish the normal address bits($log_2 n$) and the masked address bits $1/2(log_2(n/k))$, two up/down synchronous counters, *BMBAG* and *MdAG*, shown in Figure 3 and Figure 4 respectively, are used. In the test mode(TM), $1/2(log_2(n/k))$ bits of the column address bits are always masked. But the same number of bits of the row address are masked only in write operations. So *MdAG* which generates these row address bits for read operations in the test mode is required. *MdAG* generates the higher $1/2(log_2(n/k))$ row address bits and the lower $1/2(log_2 k)$ row address bits can still be generated by *BMBAG* in read operations. This approach using two counters solves the problem of the difference of the number of read and write operations which could result in a complex control logic in case of using only one counter. In Figure 3 and Figure 4, the added bits such as $\overline{T}/D$, A, B, and C are used to generate control signals. Only these signals and the generated clock signal are needed with the independence of the memory size. These signals indicate that which march element to be performed, that the current operation should be a read operation or a write operation, and that the next address to be a increased value or a decreased value. The mask address generator(MAG) is shown in Figure 5. All mask column address bits are always 1's in the test mode to mask $1/2(log_2(n/k))$ bits of the column address. However, the mask row address is in value of 0's in order not to mask $1/2(log_2(n/k))$ bits of the row address when the read operations(RM) are performed in the test mode as shown in Figure 5. The data generator and the fundamental signal gen-

erator to generate the BIST control signals can be easily designed.

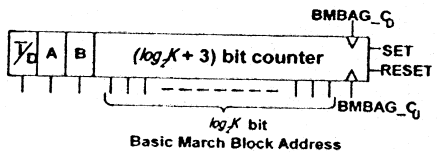

Figure 3: Basic March Block Address Generator

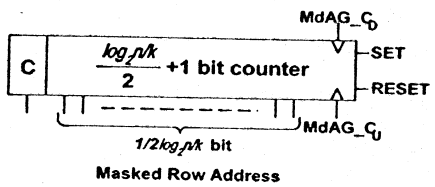

Figure 4: Masked Row Address Generator

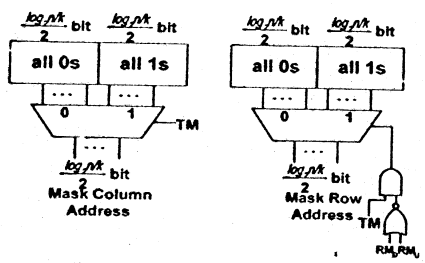

Figure 5: Mask Address Generator

5 Results and Conclusion

In the previous sections, the parallel testing methodology and the BIST architecture has been presented. By using parallel testing technique, the test time can be reduced to a great extent. The proposed algorithm can test an $\sqrt{n} \times \sqrt{n}$ RAM which consists of $\sqrt{k} \times \sqrt{k}$ basic marching blocks in $O\left(5 \times \sqrt{k} \times \left(\sqrt{k} + \sqrt{n}\right)\right)$ time. Table 1 shows the number of operations required for different block sizes.

The number of the additional transistors and the fault coverage loss for different block sizes are shown in Table 2. The total hardware overhead becomes $(20 \log_2 k + 23.5 \log_2 n + 291)$. The fault coverage loss results from the undetectable coupling faults between the cells at the same position in each block. Note that the proposed parallel test method is much more faster than the conventional March C- algorithm and as the RAM size increases by a factor of 4, the test complexity increases by a factor of 2 and the hardware overhead in the BIST architecture increases only by 47 transistors.

Thus this architecture can be efficiently used for ultra high density memories and can be modified a little to speed up the other march algorithms. The fault coverage loss which is proportional to the inverse of the block size is very small and can be negligible.

Block Size	The Number of Operations			
	4M	16M	64M	256M
64K	2.93M	5.57M	10.81M	21.29M
256K	6.55M	11.79M	22.28M	43.25M
1M	15.73M	26.23M	47.18M	89.12M
March C–	41.94M	167.77M	671.08M	2.68G

Table 1: The Number of Operations

Block Size	The number of Added TRs				Fault Coverage Loss
	4M	16M	64M	256M	
64K	1128	1175	1222	1269	0.001526 %
256K	1168	1215	1262	1309	0.000381 %
1M	1208	1255	1302	1349	0.000095 %

Table 2: The Hardware Overhead and the Fault Coverage Loss

References

[1] K. K. Saluja, S. H. Sng and K. Kinoshita, "Built-In Self-Testing RAM: A Practical Alternative," in *IEEE Design and Test of Computers*, Feb. 1987, pp. 42–51.

[2] S. Jain and C. Stroud, "Built-in Self Testing of Embedded Memories," in *IEEE Design and Test of Computers*, October 1986, pp. 27–37.

[3] R. Nair, S. M. Thatte and J. A. Abraham, "Efficient Algorithms for Testing Semiconductor Random-Access Memories," in *IEEE Trans. on Computers*, June 1978, pp. 572–576.

[4] A. J. Goor, *Testing Semiconductor Memories; Theory and Practice*. Singapore: John Wiley and Sons, 1995.

[5] Y. S. Kang, J. C. Lee and S. Kang, "Parallel BIST architecture for CAMs," in *Electronics Letters*, January 1997, pp. 30–31.

[6] R. Dekker, F. Beenker and L. Thijssen, "A Realistic Fault Model and Test Algorithm for Static Random Access Memories," in *IEEE Trans. on CAD*, June 1990, pp. 567–572.