Memory Chip BIST Architecture

Jacob Savir
ECE Dept.
New Jersey Institute of Technology
University Heights
Newark, New Jersey 07102-1982
Ph. (973) 596-5681, Fx. (973) 596-5680, e-mail savir@oak.njit.edu

Abstract

This paper describes a random access memory (RAM, sometimes also called an array) test scheme that has the following attributes:

1. *Can be used in both built-in mode and off chip/module mode.*
2. *Can be used to test and diagnose naked arrays.*
3. *Fault diagnosis is simple and is "free" for some faults during test.*
4. *Is never subject to aliasing.*
5. *Depending upon the test length, it can detect many kinds of failures, like stuck-cells, decoder faults, shorts, pattern-sensitive, etc.*
6. *If used as built-in feature, it does not slow down the normal operation of the array.*
7. *Does not require storage of correct responses. A single response bit always indicates whether a fault has been detected. Thus, the storage requirement for the implementation of the test scheme is zero.*
8. *If used as a built-in feature, the hardware overhead is very low.*

1 BIST Architecture

The density of memory chips has been increasing at a fast pace. Memory chips having 1GB and higher are in the horizon. More and more digital designs comprise numerous embedded arrays. This trend is expected to continue in the foreseeable future.

Testing these memories has become a major concern because algorithms of complexity $O(n^2)$, where n is the number of bits in the memory chip, are no longer acceptable because of their extensive cost. On the other hand, the linear algorithms in use today fall short of detecting considerable number of failure mechanisms.

Memory tests that are typically used include tests such as GALPAT, checkerboard, sliding diagonal, [1, 3] etc. These tests are not geared to detect newly emerging manufacturing defects. Consequently, their quality, in terms of fault coverage, is not adequate for detecting non-traditional faults. Their main reason for existence is that they have performed satisfactorily in screening out bad products in the early days, where failure mechanisms were not that "sophisticated". One can argue that better test strategies can be devised at a much lower cost.

Built-in self-test (BIST) has become a major design for testability vehicle in the last fifteen years. BIST has been employed at all levels of digital designs [2]. Today, chips, cards, modules, or even full computer systems employ BIST.

The purpose of this paper is to provide a unique testable design for RAMs. The testable design uses linear feedback shift registers and a comparator to assist the test. These test aids can be placed on or off product, depending upon the design constraints. The testable design proposed in this paper offers many benefits in terms of simplicity, modularity, fault coverage, and fault diagnosis.

The idea of this paper can be simply described as follows. If the data to be stored inside the array could be algorithmically generated, and if the same data can be independently regenerated during the readout time, then a simple comparison circuitry can tell whether the word currently being read is in error. A miscompare at any given time will not only detect a failure, but will also pinpoint to the affected address.

We illustrate the method of test and diagnosis on a single-port naked array, but the scheme may be employed on modern memory chips as well. Embedded arrays may be treated similarly, except that a one-to-one correspondence has to be established between primary inputs (PIs) or latches and address lines before read and

0-7695-0104-4/99 $10.00

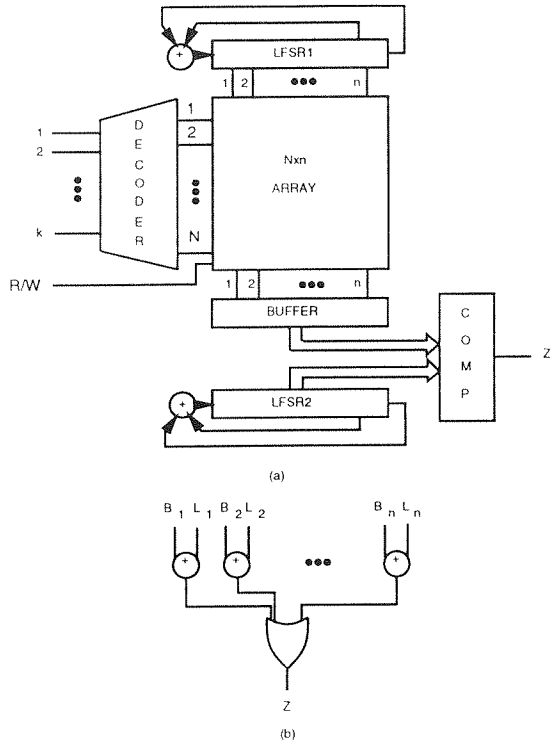

Figure 1: (a) A naked array in test mode (b) The comparator expansion

write operations. Multi-port arrays can also be treated in a similar fashion. Fig. 1 is a block diagram of the hardware involved in test mode.

The single port array consists of N words of n bits each. The decoder associated with the array has k inputs such that $2^k \geq N$. The array is assumed to have a buffer that holds the contents of the accessed word in read mode. The buffer is not essential, though, and the method will also apply when the array has no such buffer. The R/W line is the read/write control. LFSR1 and LFSR2 are two identical linear feedback shift registers implementing a primitive polynomial of degree n. The comparator (COMP) is shown in block diagram form in Fig. 1(a), and in expanded form in Fig. 1(b). The output of the comparator indicates whether a fault has been detected at any time during the test. The inputs to the comparator are fed from the buffer $(B_i, i = 1, 2, ..., n)$ and from the stages of LFSR2 $(L_i, i = 1, 2, ..., n)$.

The test is conducted in the following way. The R/W control is set to write mode. The address space is stepped from 1 to N (address stepper not shown in Fig. 1). LFSR1 and LFSR2 are seeded with the same nonzero seed (any nonzero seed will do). LFSR1 and the address space are synchronously advanced writing all addresses with LFSR1 contents. The address stepper is then initialized to point to address 1. The R/W control is then set to a read mode, while LFSR1 is frozen in its last state. Walking through the address space again, the stored information is read and dumped into the buffer word by word. During this read operation LFSR2 comes into play, cycling synchronously with every read operation. Since LFSR2 was seeded with the same seed as LFSR1, each time a word is read from the array, its correct version is generated by LFSR2. The comparator outputs a value 0 each time the contents of the buffer matches that of LFSR2. A fault is detected when the comparator outputs $Z = 1$. Whenever a miscompare occurs the decoder still points to the affected bad word. As a result, information about bad words is accumulated "for free" during the test. After a full write/read cycle through the memory is complete, a second cycle is initiated in the same manner. Both LFSR1 and LFSR2 still have the same seed in them (different from the first seed, though), and they are ready for the next cycle.

Note that in most practical applications $2^n > N$. Thus, the data written into the addresses are all different in any given write/read cycle. The number of cycles needed to test for faults depend on which faults are being targeted.

It is instrumental to mention what is the hardware overhead associated with this scheme. For BIST applications LFSR1 and LFSR2 can be made one because they are needed at different times. LFSR1 is frozen when LFSR2 is working, and vice versa. This, however, will require scanning in the previous seed when the LFSR switches from its LFSR1 role to LFSR2 role. The comparator constitutes an overhead (n 2-input Exclusive-Or gates and one n-input Or gate). If external test is used, the LFSRs and the comparator can be placed inside the tester without affecting the array product.

References

[1] M. Abramovici, M.A. Breuer, and A.D. Friedman. *Digital systems testing and testable design.* IEEE Press, Piscataway, New Jersey, 1994.

[2] P. H. Bardell, W. H. McAnney, and J. Savir. *Built-In Test for VLSI: Pseudorandom Techniques.* Wiley Interscience, New York, 1987.

[3] A. J. van de Goor. *Testing semiconductor memories theory and practice.* John Wiley & Sons, 1991.