

Test generation for VLSI chips with embedded memories

by E. K. Vida-Torku
J. A. Monzel
T. L. Bossis
C. E. Radke
D. M. Wu

An effective approach for generating patterns for testing memories embedded in logic is presented. Through circuit testability analysis, which is a study of the effect of process defects on memory circuits, unique algorithms can be derived for testing the memory. Circuit and logic designs for test features that are required to make the pattern generation process optimal are discussed. An analytical method is described which assesses the performance characteristics of the memory after functional test.

1. Introduction

As chip density increases, random access memories (RAMs) are embedded in logic portions of the chip. These memories can be register files or cache memories. Machine throughput is enhanced, because circuit signals need not cross off-chip drivers and receivers to access these memories. **Figure 1** shows a general form of a memory embedded in logic. The memory can be read only through the logic on-chip—the inputs and outputs are not available as primary pins. Various different test schemes have been suggested to make these memories testable [1, 2].

©Copyright 1990 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

One of the difficulties of testing embedded memories is that they are often tested as traditional memories, which implies subjecting them to numerous test patterns.

In this paper a different method is described, in which embedded memories are tested with an optimal test pattern set that is technology dependent. The application or test time can thus be greatly reduced.

Two approaches can be used when generating test patterns for memories embedded in logic: functional test and explicit memory test. In the former, functional patterns are generated to test the entire logic function of the VLSI chip [3]. For example, the correct operation of a microprocessor is tested through functional patterns. The memory embedded in the microprocessor may also be tested by functional patterns. This paper addresses the second approach, which is explicit memory test pattern generation for embedded memories. The test patterns are generated as if the memory were a separate entity, not embedded in logic.

Linked with the test pattern generation procedure is the test scheme of the embedded memory. If the memory is to be tested like any traditional memory chip, special design for test features must be incorporated. In [1], some possible solutions were presented to ensure traditional memory performance parameters such as access time.

Three possible test schemes were suggested:

1. Traditional memory tester approach. In this method, the inputs and outputs of the memory are brought to chip pads for dynamic or ac functional speed testing, ensuring direct measurement of access time, hold time, and other parameters.

2. Memory test with slow-speed testers through the logic on-chip. Since this is a static or dc test, no accurate performance parameters are measured. The performance of the memory must be ensured through statistical defect level projections.
3. Built-in self-test [4]. This is used to generate test patterns for testing the memory; the patterns can be applied in a range of speeds from slow to nearly the functional speed of the memory.

The scope of this paper is limited to the generation of explicit test patterns for testing memories embedded in logic, ensuring performance through statistical defect projections. It is assumed that the same tester is used to test both the memory and the logic on-chip.

This approach addresses the following problems:

- Generating an optimal test pattern set.
- Applying the test patterns to the memory through the on-chip logic and observing the output.
- Testing the memory in a reasonable amount of time.
- Assessing the ac performance specifications of the memory.

2. Pattern derivation based on testability analysis

To obtain optimal memory test patterns, the memory is characterized according to the failure modes that can affect it. This characterization depends on design, technology, and physical defects; it is an extension of the work reported in [5].

Five steps are involved in obtaining optimal test patterns for embedded memories. These steps, outlined in the flowchart of Figure 2, are described below.

1. Identification of physical defects

All possible manufacturing defects that can occur in the memory are identified and tabulated. Defects likely to affect memory functionality or having a high probability of occurrence are selected.

2. Circuit simulation

Defects selected in step 1 are simulated using a circuit simulation program such as SPICE or ASTAP to determine the impact of the defect on the memory. This simulation separates the defects into two classes, those affecting the functionality of the memory and those degrading the performance of the memory.

3. Mapping of faults into failure modes

The faults identified in step 2 are mapped into different types of failure modes, e.g., stuck cells or reading the wrong cell.

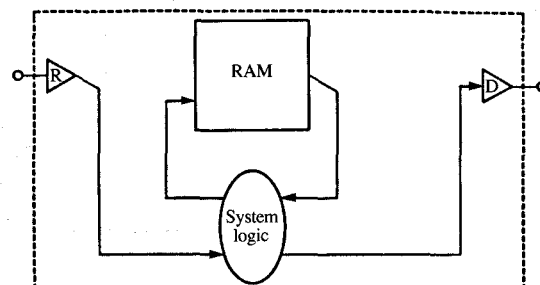


Figure 1

RAM embedded in logic on a chip. R = receiver, D = driver.

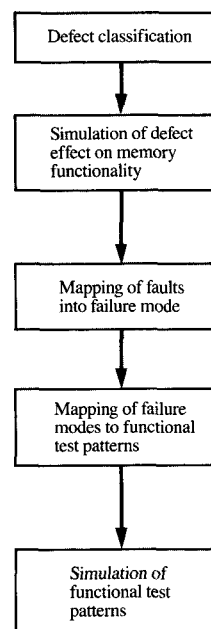


Figure 2

Test pattern derivation procedure.

4. Derivation of functional test patterns from failure modes

A set of algorithmic patterns are derived to cover all failure modes identified in the memory.

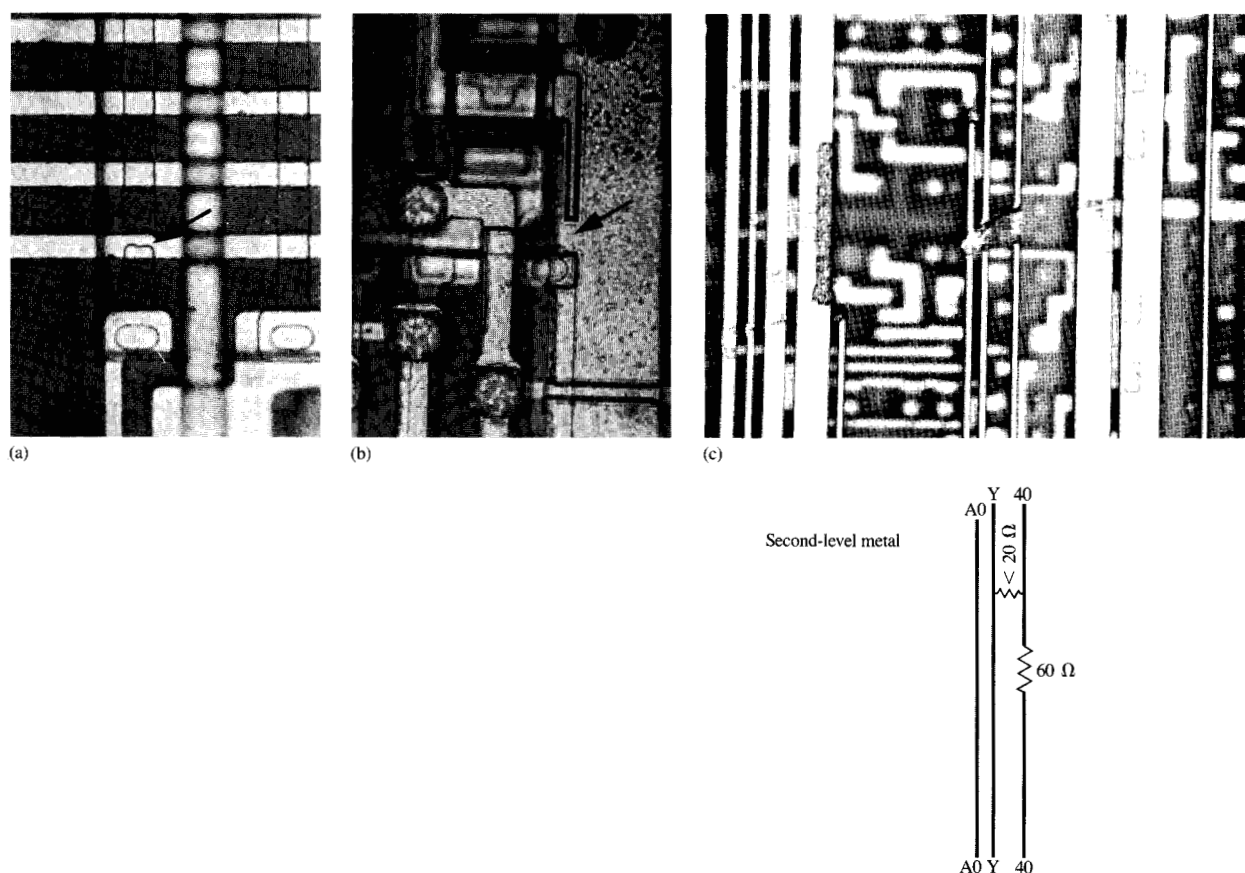


Figure 3

(a) Missing portion of resistor (base diffusion); (b) missing first-level metal; (c) second-level metal with scratch.

5. Fault simulation

The algorithmic patterns developed are simulated by a failure mode simulator. This step is necessary to ensure that any new test algorithm detects a set of failure modes that have been defined.

3. Failure modes

• Physical defects

Faults in a memory circuit are caused by physical defects (called point defects at the process level). **Figure 3** shows three examples of point defects. Figure 3(a) is a diffusion defect; a missing portion of a resistor results in a cell that cannot latch data. In Figure 3(b), a portion of the first-level metal is missing; the metal is supposed to extend beyond the arrow to the contact but does not. Figure 3(c) shows a scratch at the second-level metal that resulted in a short. In this case, nodes 40 and Y are shorted together.

Node A0 was not shorted electrically, although it is covered by the scratch.

Some possible point defects in a bipolar technology are listed in **Table 1**. This table is divided into two groups: defects that affect transistor manufacturing and those that affect intertransistor wiring. Most of these defects affect a circuit as a short between two nodes or an open contact.

Circuit simulation using ASTAP is performed with these defects on a cross section of the memory, as shown in **Figure 4**. This is a three-port memory with two independent read addresses and one write address. The defects injected in a bottom-up approach are component defects and intercomponent defects.

• Memory cell defects

Most defects in a memory cell result in a stuck cell—the output of the cell driving the sense amplifier is stuck at one or at zero. However, some defects result in other failure modes, such as the following:

- Defective cell dominates the output. This is not a stuck cell. Rather, in this failure mode the output of the memory is always a function of the defective cell.
- Defect in a memory cell appears at the memory output as a function of the state of the good cells on the same bit line. For example, if there are eight cells on one bit line and one of the cells has this particular failure mode, the memory output will be a logic 1 if four cells contain a logic 1.

• Decoder defects

Defects in a decoder result in conventional failure modes: Read or Write no memory word, Read or Write multiple words, and Read or Write the wrong word.

4. Test conditions for detecting failure modes

Most memories embedded on logic chips are designed such that crosstalk between cells and other "ac disturb" faults are negligible. The testability analysis in Section 2 ensures that most defects affect the function of the memory. As such, these defects can be detected by what we call dc test patterns. In this section, we present the algorithm for generating patterns. The approach is different from the traditional memory test [6-8]. Our objective is to be able to detect all defects in the memory with test patterns that require $O(n)$ Write/Read operations, where n is the total address space.

In the following section, a unique address location is represented by i . A Read operation at address i is represented by R_i . Similarly, a Write operation at address i is represented by W_i . Linear addressing is used in all the presented algorithms; i.e., the address is incremented or decremented by 1. Further, it is assumed that a divide-by- n counter is available for test, implying that an increment operation from the highest address ripples to address 0. Similarly, a decrement operation from address 0 ripples to the highest address. The address operator N increases the address counter by 1, and the address operator D decreases the address counter by 1. The other nomenclature used is as follows: $R_i(a)$, read pattern a from address i ; $W_i(a)$, write pattern a at address i ; $W'_i(a)$, write complement of pattern a at address i ; and $W_i \& R_i$, simultaneous read and write at address i . $R_i(a)W'_i(a)N$ represents a sequential operation of Reading address i followed by Write at i , followed by increment operation on the address counter.

The patterns represent logical 1s and 0s as follows:

- 00...: a string of 0s for the number of bits contained in the memory location.
- 11...: a string of 1s for the number of bits contained in the memory location.
- 0101...: alternating 0s and 1s, commencing with 0.
- 1010...: alternating 1s and 0s, commencing with 1.

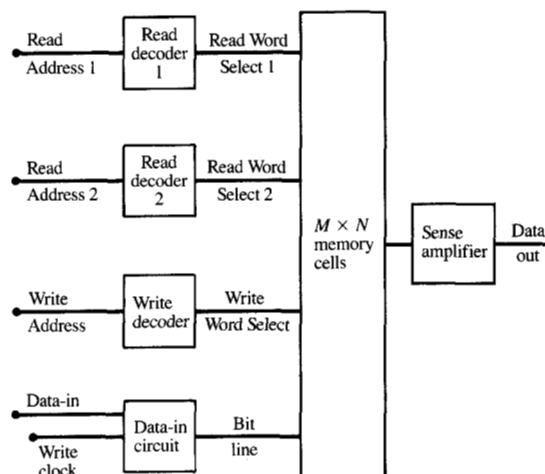


Figure 4

Cross section of three-port memory.

Table 1 Typical bipolar process defects

Process defects	Personalization defects
Isolation opens	1st metal opens
Emitter-base shorts	1st metal shorts
Collector-isolation shorts	2nd metal opens
Collector-emitter leakage	2nd metal shorts
Resistor opens	3rd metal opens
Resistor shorts	3rd metal shorts
Schottky diode leakage	4th metal opens
Collector contact integrity	4th metal shorts
Emitter contact integrity	1st-2nd metal shorts
Base contact integrity	2nd-3rd metal shorts
Sidewall integrity	3rd-4th metal shorts
	Contact integrity

Multi-port memories have two or more independent ways to address and operate on a single memory. A port is an address group plus either a Data-in or Data-out group and the controls needed to operate the grouping. These ports are usually referred to as Read or Write ports. For multi-port memories, $R_{ji}(a)$ denotes reading pattern a from address i of port j , and $W_{ji}(a)$ denotes writing pattern a at address i of port j .

For each of the defined failure modes, necessary and sufficient conditions are presented to detect it. It is assumed that one failure mode is present in the memory at a time. It is also assumed that there is at least one independent port for reading and a second port for writing.

- *Faults affecting only single ports*

Stuck cell

To detect a stuck-at-1 (0) in a memory cell, the cell must be initialized to 0 (1); the value 0 (1) must be read before a Write operation is performed on the cell.

No-select on Write

This is a Write decoder fault. Let f be the memory word not selected as a result of the fault. A necessary and sufficient condition for detecting this fault is

- Write pattern a at word f , followed by read.
- Write and read the complement of pattern a from f .

Since word f cannot be selected for a Write operation, the word contains a random pattern. Writing and reading a specific pattern is necessary to detect this failure mode. There still is a very small probability that the random pattern is a . Writing and reading the complement reduces this probability to zero.

No-select on Read

This is a Read decoder failure mode whose detection depends on the technology and special design for test features included in the memory design. A requirement for detecting a no-select condition is that the sense amplifier must be able to act as a three-value logic at its input, discriminating between logic 0, 1 and a third level that indicates no-select. This condition may be sufficient, depending on the technology. In one bipolar current switch design, for example, the output of the sense amplifier is a function of the relative number of 0s and 1s. Thus, if there are more 0s than 1s, the output is 0. The cells supply current equally down the bit line into the sense amplifier. Similarly, the output is 1 if there are more 1s than 0s. The three-value logic feature of the sense amplifier is valid only if there are equal numbers of 1s and 0s.

It is more difficult to determine the expected output of the memory based on the number of 1s or 0s on the sense amplifier than to include test circuits that correct the problem.

Write multiple words

Let i be the word being addressed and f the word being written as a result of the failure mode. To detect this failure,

1. f should contain the complement of the pattern written at word i .
2. The test sequence for each word i should include $R_i(a)W'_i(a)$, where a is the background pattern used to initialize the memory.

3. The linear addressing should ripple through the memory in both ascending and descending order.

If words i and f contain the same data, the fault will not be detected during read-out. Condition 1 is necessary to distinguish between the data at words i and f . The Read operation in condition 2 ensures that the complement data at word f are read before the value in f is destroyed. The Write operation satisfies condition 1.

If $f > i$ and conditions 1 and 2 are fulfilled, the fault is detected when addressing is in ascending order. However, if $f < i$, the fault can only be detected when addressing is in descending order. Conditions 1, 2, and 3 are sufficient to detect multiple select faults in a write decoder.

Read multiple words

The detection of this failure mode (also called multi-select) is also determined by the technology or special design. Again, let i be a word being written and f the word selected because of the defect. The necessary and sufficient conditions for detection are as follows:

1. Words i and f should have complementary data.
2. The sense amplifier should be able to discriminate between reading 0 and 1 on the same bit line simultaneously and reading normal 0 or 1.

Since two words are read simultaneously, the fault will not be detected if both words have the same value. Condition 1 is necessary to force both logic 0 and 1 into the sense amplifier, and condition 2 is needed to detect it.

Condition 2 is not valid in most differential sense amplifier designs. A multi-select condition produces input values to the differential amplifier that are almost identical, resulting in a reduced noise margin. There are two possible design options: First, the differential amplifier can be designed such that it favors one state over another. Alternatively, it can be designed such that it monitors the word lines and forces a known state on the output if more than one word line is active.

The word lines can be monitored by adding test circuits to every Word Select line and dotting them together. The output of the test circuits must have the following properties.

- Output must be 1 if one Word Select line is active.
- Output must be 0 if more than one Word Select line is active. This output can then be used to control one of the data output pins by gating the sense amplifier as illustrated in **Figure 5**.

These schemes transform a potential delay or noise-margin problem into an easy-to-detect stuck-at fault.

Defective cell dominates output

The output of a bit line depends on the value in the defective cell. The cell is always supplying current to the sense amplifier, whether or not it is being addressed. This is not a stuck-cell nor a multi-select condition. However, it can be shown that it is detectable by the conditions for detecting Read multiple word failure mode discussed previously.

• Multi-port faults

Metallization shorts between decoder lines of two ports can produce failure modes that require multi-port testing. Most of the test conditions for detection are dependent on the technology.

Shorts between two Read ports

Assume that a short between the decoders of the two ports places both ports in Read mode when one port is being read. Because of differences in layout, it is assumed that a short between ports can only occur on decoder lines selecting the same word. Finally, assume that during test, memory Read is done using one port at a time. Let f be the site of the fault. In addition to the conditions for detecting a multiple read in a single port, the following condition is required.

Condition A

While one of the Read ports (e.g., port 1) is selecting word f , the other port (port 2) must be reading some word i , where i is not equal to f .

If both ports have the same address, the Read-out data at the outputs will not detect the fault. If port 1 is pointing to word f and port 2 is pointing to some word i (not equal to f), then port 2 will be forced to have both words i and f selected because of the fault, and is detected by the conditions of the *Read multiple words* section. These conditions, together with Condition A described above, are sufficient to detect the fault. Condition A requires independent control of both Read ports during test.

Other multi-port shorts

Depending on the layout, it is possible to have Read-to-Write-port shorts and Write-to-Write-port shorts.

Condition A is required to transform these failure modes to the appearance of single-port failure modes. Once this condition is fulfilled, the other conditions become trivial, and they are not discussed here.

• Test algorithms

The conditions discussed in the previous sections are used to derive a set of compact algorithms for detecting the various failure modes.

In the following steps, $a = 010101 \dots$, b is the complement of a , and n is the address space:

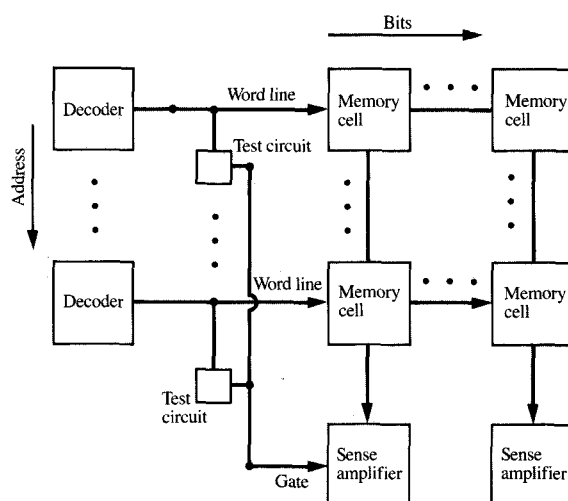


Figure 5

Test circuits to monitor multi-select condition in a memory.

1. Initialization

For $i = 0$ to $n - 1$, $W_i(a)$ (Write background pattern).

2. Forward ripple

For $i = 0$ to $n - 1$, $R_i(a)W_i(b)R_i(b)N$.

This step reads the background pattern and writes the complement, rippling through the address space from 0 to n .

3. Reverse ripple

For $i = n - 1$ to 0, $R_i(b)W_i(a)R_i(a)D$.

This step reads the new background pattern b and writes the complement a , rippling through the address space from $n - 1$ to 0.

For a two-port memory (separate read and write ports), these three steps are sufficient to provide all the defect-dependent test conditions discussed in this section. Note again that design for test features must be provided to detect some of the failure modes. For memories with more than two ports, a slight modification is required to satisfy the conditions discussed in the section on multi-port faults.

The memory is tested as defined in steps 2 and 3 using a pair of Read-port and Write-port combinations. For example, in a four-port memory, Read port 1 and Write

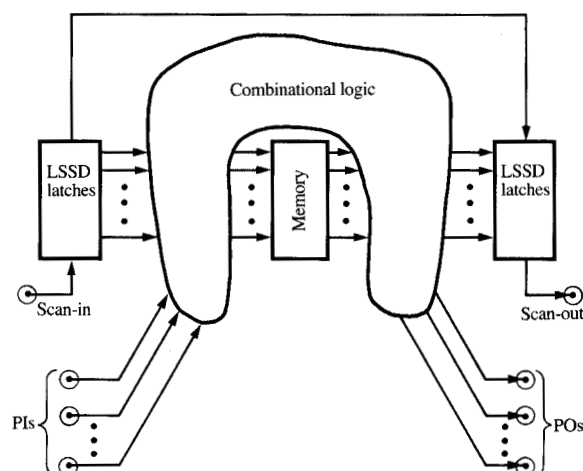


Figure 6

Embedded memory in a LSSD environment.

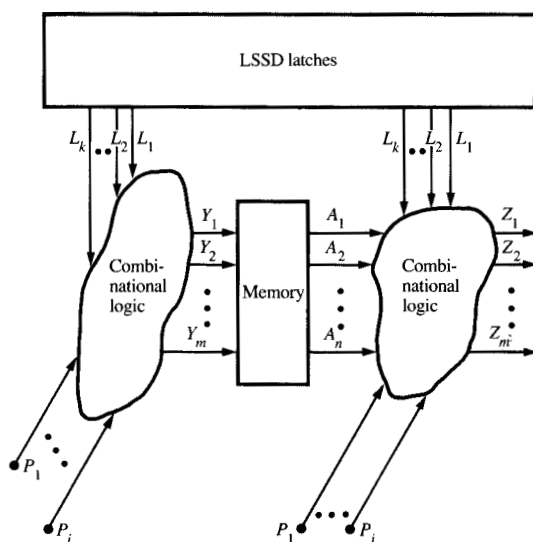


Figure 7

Controllability and observability of embedded memory by LSSD latches.

port 1 are used simultaneously. Similarly, Read port 2 and Write port 2 are used simultaneously for test. To

fulfill Condition A, Read port 2 should be rippled in steps 2 and 3 such that Read port 2 points one word behind Read port 1. Similarly, keeping Write port 2 one word behind Read port 1 fulfils the condition for detecting a short between these ports.

5. Test pattern generation for embedded memory chip

The general form of a chip with embedded memory shown in Figure 1 is redrawn in a level-sensitive scan design (LSSD) [9] environment, as shown in Figure 6. The shift register latches (SRLs) in Figure 6 can be used as primary inputs and outputs. Test generation for the chip is divided into two portions, logic test generation and memory test generation. Logic test generation generates test patterns for the combinational circuits and latches on the chip. The memory is used to test the logic. In general, two steps are involved:

1. (a) Set chip primary input values and load LSSD latches. (b) Write into the memory.
2. (a) Set new primary input values and reload LSSD latches. (b) Read the memory. (c) Take the results at the chip primary outputs and latches.

• Memory test generation

The algorithms presented in the previous section were defined in terms of the memory input, not in terms of the logic surrounding the memory. Hence, to generate the test patterns for the memory in terms of the primary inputs, one must trace back through the logic from the memory inputs to the chip primary inputs or SRLs. Similarly, the memory output test patterns must also be propagated through logic to chip outputs or SRLs. This is the classical controllability and observability problem applied to embedded memories.

The back-tracing and propagation problem can be simplified if logic design for test features is used. It should be possible to isolate the inputs and outputs of the memory logically from the on-chip logic, as illustrated in Figure 7. For the m outputs of the combinational logic that feed the memory, there are m chip input pads [primary inputs (PI)] and/or SRLs such that there is a logical path from each PI or latch to the memory. Similarly, there should be a logical path from the N outputs of the memory to chip primary outputs or SRLs. Logically, in order to isolate the inputs of the memory in Figure 7, for every Y_k ($k = 1, M$), there must exist simultaneously a unique L_i or P_j such that

$$Y_k = L_i \vee P_j,$$

where $i = 1, K$ and $j = 1, J$. Similarly, at the output, for every A_n there should exist simultaneously a unique Z_m for some set of values at L_k and P_i such that

$$Z_m = A_n,$$

where $n = 1, N$; $m = 1, M$; $k = 1, K$; and $i = 1, I$.

We refer to controllability and observability for embedded memories achieved by LSSD latches as shift register isolation. This is in contrast to chip-input-pad-chip-output-pad isolation schemes, in which all the memory data input, output, and control lines are brought to chip pads [1].

- *Memory test time and test data volume considerations*
One of the problems that must be solved when memories are embedded in VLSI chips is how to minimize test time—the time it takes to test the logic on-chip and the embedded memory. Similarly, the volume of test patterns to be applied is of significant concern.

The embedded memory test time is a function of the test strategy implemented for each design and the test patterns.

Test patterns

One of the primary features of the outlined procedure is to develop a set of compact test patterns based on the process defects. The test algorithms derived in Section 4 are of order n [$O(n)$], where n is the address space of the memory. This contrasts sharply with the usual $O(N^2)$ or N^2 patterns that are used to test memories [6–8], where N is the number of cells in the memory. By deriving test patterns following the procedure of Section 4, test time (and test data volume) can be greatly reduced, regardless of the test strategy used.

Test strategy

Given a set of compact test patterns generated in Section 4, the test methodology that is used can affect the test time and test data volume. While a detailed test strategy discussion for embedded memories is beyond the scope of this paper, the following observations are presented:

- Shift register isolation, as described in the subsection on memory test generation, saves input and output pins. During test, the shift registers must be loaded and unloaded in order to write and read from the memory. This process can be time-consuming and can generate considerable test data volume, depending on the length of the shift register chains.
- Bringing all memory inputs and outputs to chip pads to obtain primary-input-to-primary-output isolation can reduce test time and test data volume. It can be used to perform traditional memory test. Although these pads need not be dedicated to the memory [1], the use of chip pads becomes more complicated when there are multiple memories on a chip.

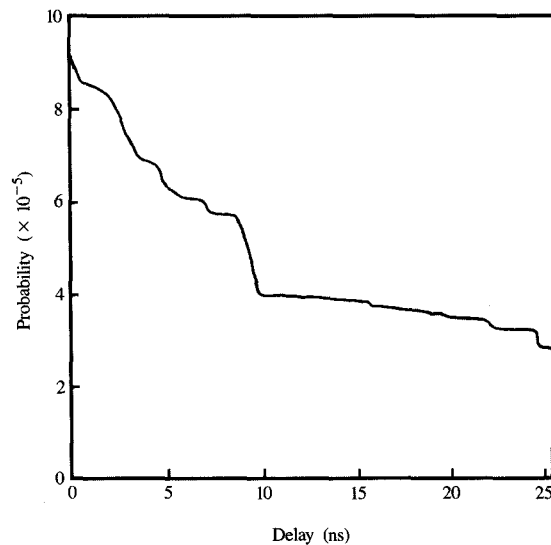


Figure 8

Defect density delay distribution function.

- Built-in self-test for memories offers a solution for the test data volume problem, since patterns are generated on-chip [2]. Test time can also be minimized, depending on the clocking scheme used.

6. Performance fallout projection

The functional test patterns generated in Section 5 can only detect those defects that affect the functionality of the memory at the speed of the tester. However, when a memory is operating in a system, some manufacturing defects are likely to cause it to fail due to an increase in access time beyond the specifications of the memory. Defects that affect only the ac specifications of the memory are referred to as “performance defects.” To estimate the number of embedded memories with performance-related defects, a performance fallout projection is made. For ease of computation, assume that a memory can be partitioned into two mutually exclusive paths, Read path and Write path. The total fallout of the RAM in a system is the fallout due to each of these paths.

• Read path fallout

To project the fallout in the Read path of a RAM, the cumulative defect density delay distribution function (Figure 8) of the Read path is calculated [10]. Two steps are involved in the derivation of the curve of Figure 8:

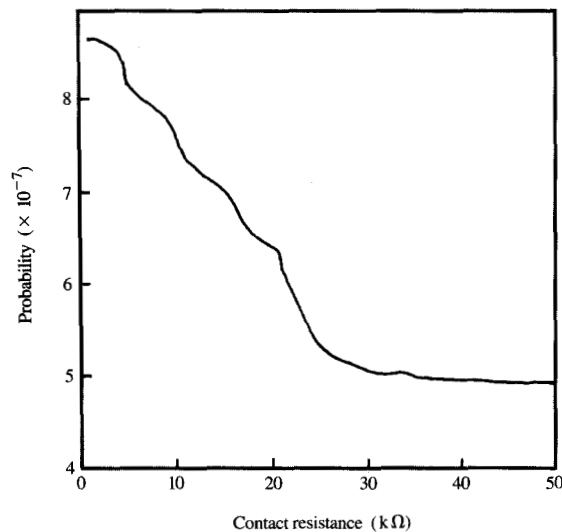


Figure 9

Probability of contact defect vs. contact resistance.

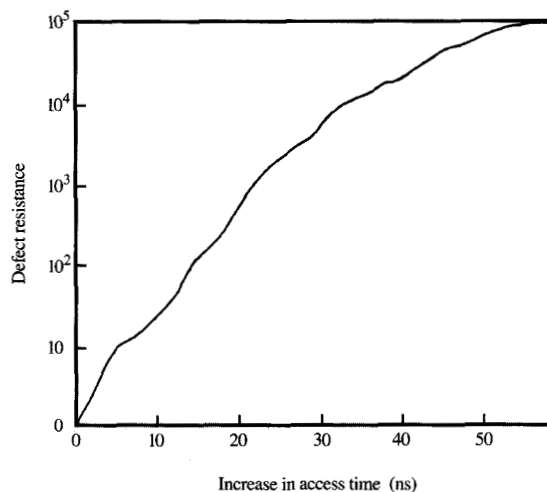


Figure 10

Defect resistance as a function of access time.

1. The probability of occurrence of each process defect type is calculated using yield data [11]. Figure 9 is an example of the probability curve of a metal contact

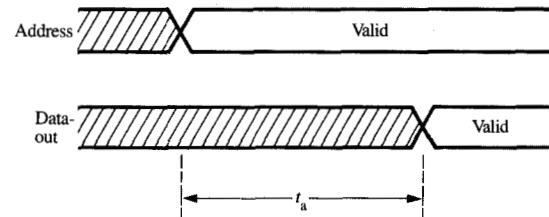


Figure 11

Read timing diagram.

open. A corresponding probability curve is obtained for each defect type.

2. The sensitivity curve of Figure 10 is combined with the probability curve of Figure 9 for each defect size. The cumulative effect of all possible defects, their probabilities of occurrence and the sensitivity of the memory to these defects is the defect density delay distribution (D4) function shown in Figure 8. Figure 11 is the timing diagram of the Read operation in a memory. When the Read address becomes stable, the output of the memory is valid after t time units. The read access time is t ; it is usually specified as a worst-case value, t_{wc} . Assume that the read access time of the memory has a normal distribution, $f_p(x)$. The effect of defects on the access time can be found by the convolution of the derivative of the D4 curve, called $f_d(x)$, and $f_p(x)$.

As an example, a Rayleigh distribution function is used for $f_d(x)$ to perform a qualitative analysis, and can be expressed by

$$f_d(x) = \left(\frac{x}{a}\right)^2 \exp -\left(\frac{x}{2a}\right)^2, \quad (1)$$

where a represents the degree of random defects. It is assumed that the random defects are independent of process variations.

From the statistical theory [12], the read access distribution function, $f_1(x)$, of memories with defect can be obtained as

$$f_1(x) = \int_{-\infty}^{\infty} f_p(x-y)f_d(y)dy, \quad (2a)$$

$$m_1 = m_d + m_p, \quad (2b)$$

and

$$\sigma_1^2 = \sigma_d^2 + \sigma_p^2, \quad (2c)$$

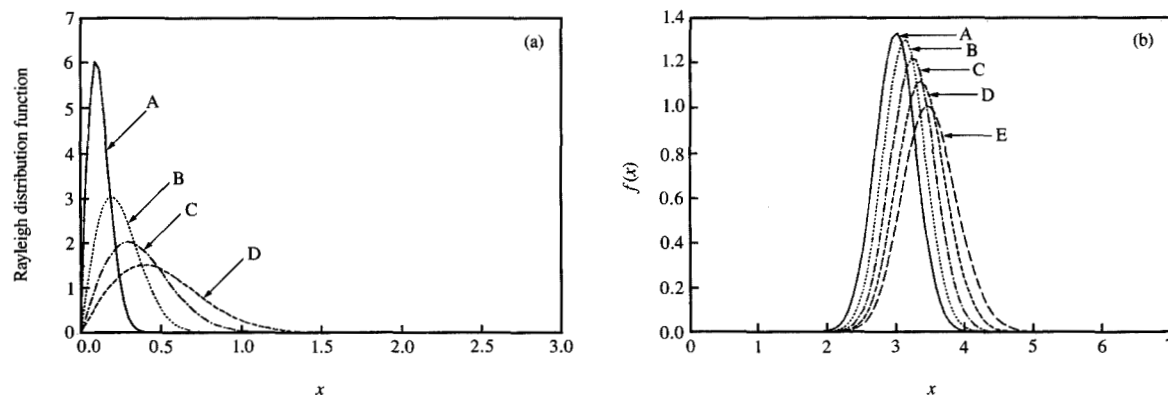


Figure 12

(a) Rayleigh distribution function: $\alpha = 0.1$ (Curve A), $\alpha = 0.2$ (Curve B), $\alpha = 0.3$ (Curve C), and $\alpha = 0.4$ (Curve D). (b) Process distribution function for memory with and without random defects: Curve A, without defects; Curves B, C, D, E, with defects and $\alpha = 0.1, 0.2, 0.3, 0.4$.

where m_i , m_p , and m_d represent the mean of $f_i(x)$, $f_p(x)$, and $f_d(x)$, respectively; here, σ_i , σ_p , and σ_d represent the standard deviation of $f_i(x)$, $f_p(x)$, and $f_d(x)$ individually. Equations (2a) and (2b) indicate that the existence of defects provides a combined distribution function, $f_i(x)$, which is shifted and spread to the right of a defect-free process distribution function. Figure 12(a) denotes the Rayleigh distribution function; Figure 12(b) depicts the shifting and spreading of the process distribution function due to defects. Curve A denotes the defect-free process, and Curves B–E represent the degree of defect, with α equal to 0.1, 0.2, 0.3, and 0.4. Figure 12(b) can be used to determine the percentage of memory chips that will have access time greater than the specified worst-case time t_{wc} .

Write path fallout

The Write path fallout projection is more complicated than that for the Read path. A cross section of the Write path of a static RAM is given in Figure 13. Four main classes of defects in the Write path can cause performance-type failures in a system:

- Delay in Word Select line switching.
- Delay in bit line switching.
- Delay of the Write pulse.
- Shrinking of the Write pulse.

Since the analysis of all four categories is similar, an outline of the procedure for the first category is given. For this type of defect, a simplified timing diagram is

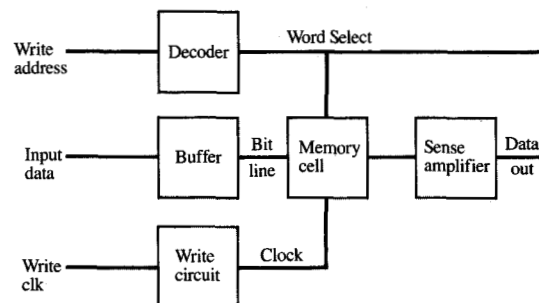


Figure 13

RAM Write path.

shown in Figure 14. Assume that the data are stable before the write clock becomes active. When the data are valid, the corresponding Word Select line becomes active after t time units. A defect in the write circuitry, for example, can cause the write clock to be late or early, causing multiple words to be written into the memory.

Let t_s be the address setup time. For a given value of t_s , the effect of the defect in the write circuit is to increase t , the Word Select time, by dt time units. As in the case of the Read path, the defect density delay distribution function of the Write path for a given t_s can be found.

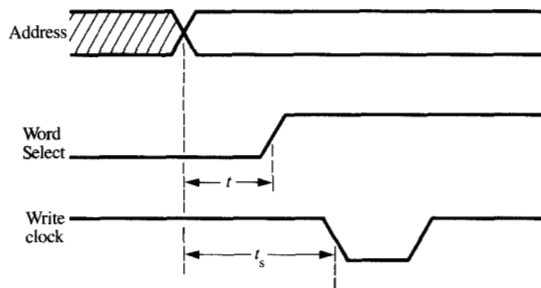


Figure 14

Write timing diagram.

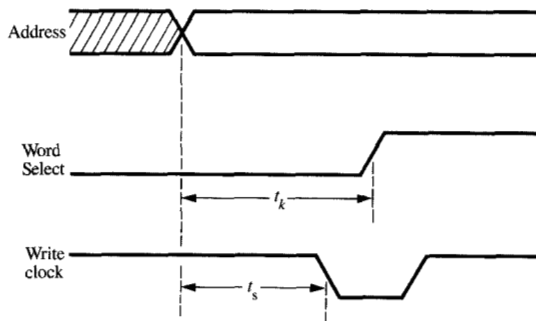


Figure 15

Write timing diagram showing write error.

The effect of process defects on t can also be found by convolution; that is,

$$P[(t = t_i) | t_s] = \int_{-\infty}^{\infty} f_p(x - t) f_w(t | t_s) dt, \quad (3)$$

where $P[(t = t_i) | t_s]$ is the distribution of the Word Select time of the memory (with defects) for a given t_s , $f_p(x)$ is the process distribution of the Word Select time, and $f_w(t | t_s)$ is the derivative of the D4 curve for a Write path at a given value of t_s . **Figure 15** shows an example of a write error occurring because the Word Select line is too long; t_k is the value of t that produces a write error. The probability of write error in the memory after a dc functional test is the Write path fallout. In Appendix A, the Write path fallout is calculated as

$$P(t \geq t_k) = \int_{-\infty}^{\infty} \int_{t_k}^{\infty} P[(t = t_i) | t_s] f(t_s) dt_s, \quad (4)$$

where $f(t_s)$ is the distribution of the setup time t_s and $P[(t = t_i) | t_s]$ is defined in Equation (3). Using Equation (4), the percentage of memories that will have Write-path-related problems can be estimated.

Summary

The development of a set of tests for the testing of memories embedded in very dense chips can be time-consuming and may still result in unknown quality. The work discussed was directed toward generating tests with defined failure modes. These failure modes are identified directly from the anticipated manufacturing defects. The tests generated for the explicit memory are applied to the memory embedded directly within an LSSD logic structure.

In an effort to reduce the failure modes and the sophistication of the tester environment needed to test the memories, the memories and their associated peripheral circuits are enhanced. Examples of design enhancements to detect multiple selects and no-selects are used to illustrate the value of the approach.

The work discussed here makes possible a lower-performance tester environment than that normally used for memory testing, while still achieving the quality objectives. Through the circuit testability analysis used to establish the failure modes, and hence define the tests, an assessment of the memory quality can be made.

Appendix A

To calculate the probability of a write error caused by a delay in the switching of a Word Select line, use the relation

$$P_x(x) = \int P_{xy}(x, y) dy. \quad (A1)$$

Let $x = (t \geq t_k)$ be the time at which a write error occurs, and $y = t_s$ the address setup time. Then

$$P(t \geq t_k) = \int P[(t \geq t_k), t_s] dt_s. \quad (A2)$$

By Bayes' theorem,

$$P[(t \geq t_k), t_s] = \int P[(t \geq t_k) | t_s] P(t_s) dt_s. \quad (A3)$$

Hence, Equation (A2) becomes

$$P(t > t_k) = \int P[(t \geq t_k) | t_s] P(t_s) dt_s. \quad (A4)$$

For a given value of t_s , the effect of process defects on the distribution of the Word Select can be found using Equation (2) of the text. For the Write path, because this

distribution is dependent on t_s , we have a conditional probability; that is, for a given t_s the distribution of Word Select delay is

$$P[(t = t_i) | t_s] = \int f_p(t - y)f_w(y)dy, \quad (A5)$$

where $P[(t = t_i) | t_s]$ is the distribution of the Word Select time of the memory (with defects) for a given t_s , $f_p(x)$ is the process distribution of the Word Select time, and $f_w(y)$ is the derivative of the D4 curve for a Write path at a given value of t_s . Since an error occurs for each t_s if $t \geq t_k$, the probability of write error for a given t_s is

$$P[(t > t_k) | t_s] = \int_{t_k}^{\infty} \int_0^{\infty} f_p(t - y)f_w(y)dy dt. \quad (A6)$$

Substituting (A6) into (A4), we obtain the probability of a write error as

$$P(t > t_k) = \int_0^{\infty} \int_{t_k}^{\infty} \int_0^{\infty} f_p(t - y)f_w(y)dy dt f(t_s)dt_s. \quad (A7)$$

Acknowledgments

Many people have contributed to our embedded memory test strategy; we wish to acknowledge particularly the contributions of C. Beh, M. Graf, and T. Kwap. The photographs of Figure 3 were provided by R. Mundoon of Quality Physical Analysis and T. Joseph of Product Assurance Physical Analysis; we appreciate their contribution.

References

1. K. Torku, J. Monzel, and C. Radke, "Performance Assurance of Memories Embedded in VLSI Chips," *Proceedings of the 1986 IEEE International Test Conference*, Washington, DC, September 8-11, 1986, pp. 154-160.
2. S. K. Jain and C. E. Stroud, "Built-In Self Testing of Embedded Memories," *IEEE Design & Test of Computers* 3, No. 5, 27-37 (October 1986).
3. K. Lai, "Functional Testing of Digital Systems," *Proceedings of the ACM IEEE 20th Design Automation Conference*, Miami Beach, FL, June 27-29, 1983, pp. 207-213.
4. K. Kinoshita and K. K. Saluja, "Built-In Testing of Memory Using On-Chip Compact Testing Scheme," *IEEE Trans. Computers* C-35, No. 10, 862-870 (October 1986).
5. C. C. Beh, K. H. Arya, C. E. Radke, and K. E. Torku, "Do Stuck Faults Reflect Manufacturing Defects?," *Proceedings of the 1982 IEEE International Test Conference*, Philadelphia, PA, November 15-18, 1982, pp. 148-156.
6. D. Suk and S. Reddy, "A March Test for Functional Faults in Semiconductor Random Access Memories," *IEEE Trans. Computers* C-30, 982-985 (December 1981).
7. M. Marinescu, "Simple and Efficient Algorithms for Functional RAM Testing," *Proceedings of the 1982 IEEE International Test Conference*, Philadelphia, PA, November 15-18, 1982, pp. 236-239.
8. C. A. Papachristou and N. B. Saghal, "An Improved Method for Detecting Functional Faults in Semiconductor Random Access Memories," *IEEE Trans. Computers* C-34, No. 2, 110-116 (February 1985).
9. E. Eichelberger and T. Williams, "A Logic Design Structure for LSI Testability," *Proceedings of the ACM IEEE 14th Design Automation Conference*, New Orleans, LA, June 20-22, 1977, pp. 462-468.
10. D. Wu and C. E. Radke, "The Effect of Random Defects and Tester Accuracy on the Quality and Yield of VLSI Circuits," *Proceedings of the 14th IEEE International Conference on Fault-Tolerant Computing*, Kissimmee, FL, June 20-22, 1984, pp. 56-63.
11. D. S. Cleverley, "Product Quality Monitoring and Control for Logic Chips and Modules," *IBM J. Res. Develop.* 27, No. 1, 4-10 (January 1983).
12. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill Book Co., Inc., New York, 1974, pp. 201-205.

Received June 16, 1989; accepted for publication October 24, 1989

E. Kofi Vida-Torku IBM General Technology Division, East Fishkill Facility, Route 52, Hopewell Junction, New York 12533. Dr. Vida-Torku is an advisory engineer in the Logic Application and Modeling Group. Since joining IBM in 1981, he has worked in the areas of developing test strategies for VLSI chips and fault modeling for logic and memories. He received his Ph.D. from the University of Oklahoma, his M.Sc. from the University of Toronto, Canada, and his B.Sc. from Kumasi University, Ghana, all in electrical engineering. Dr. Vida-Torku is a senior member of the Institute of Electrical and Electronics Engineers.

James A. Monzel IBM General Technology Division, East Fishkill facility, Route 52, Hopewell Junction, New York 12533. Mr. Monzel is a staff engineer in the Test Methodology and Applications Group. He has been involved in design for testability and developing test strategies for logic and memory circuits. Mr. Monzel received a B.A. in physics from Marietta College in 1983 and a B.Sc. in electrical engineering in 1983 from Case Western Reserve University; he is currently working on his Master's degree in electrical engineering at Syracuse University. In 1989 Mr. Monzel received an IBM General Technology Division Award.

Theodula L. Bossis IBM General Technology Division, East Fishkill facility, Route 52, Hopewell Junction, New York 12533. Ms. Bossis is an engineering technician specialist in the Logic Application and Modeling Group at IBM East Fishkill. She joined IBM in 1981, working on the testing and qualification of non-IBM products. She is currently involved with modeling and testing of VLSI chips. Ms. Bossis received an Associate degree in electronics from the Cleveland Institute of Electronics.

Charles E. Radke IBM General Technology Division, East Fishkill facility, Route 52, Hopewell Junction, New York 12533. Dr. Radke is currently a Senior Engineer Manager in the Semiconductor Laboratory. He manages a development group which is working on circuit design for testability and test methods to improve semiconductor product quality. Dr. Radke joined IBM in 1964 in Endicott; he has been involved in algorithm development and mathematical modeling relating mainly to design automation technology. More recently, he has concentrated on semiconductor

final test. In 1979 he formed the Test Methodology and Applications Group within the semiconductor product development laboratory to build and plan test into East Fishkill products. In 1964 Dr. Radke received a Ph.D. in electrical engineering from Case Institute of Technology, Case Western Reserve University. He is a member of the Computer Society of IEEE, ACM, Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

David M. Wu *IBM Systems Technology Division, VLSI Silicon Group, Boca Raton, Florida 33432.* Dr. Wu received his B.S. in physics from the National Taiwan Normal University in 1971, and his M.S. and Ph.D. in electrical engineering from the Polytechnic Institute of New York in 1977 and 1980. In 1981, he was a member of the technical staff at the COMSAT laboratory, Gaithersburg, Maryland. From 1982-1983, he was a member of the research staff at the AT&T Research Center, Princeton, New Jersey. In 1983, Dr. Wu joined IBM at East Fishkill, New York, working on the development of test strategies and techniques for bipolar VLSI chips. In 1987, he moved to IBM Research Triangle Park, North Carolina, to work on techniques and algorithms for testing CMOS products. In January 1989, Dr. Wu joined the Florida Institute of Technology as an associate professor. Since September of that year, he has been working with the IBM VLSI Silicon Group at Boca Raton, Florida, where he is exploring new algorithms and techniques to improve VLSI design and test. Dr. Wu was a session chairman at the IEEE VLSI workshops in 1987 and 1988. In 1989, he received an IBM Invention Award and a Technical Work Merit Award. He is a member of Sigma Xi and the Institute of Electrical and Electronics Engineers.