

Detection Of High Accuracy Bist-Based Fault Diagnosis For Embedded Read-Only Memories

Bingi.Vijaya Lakshmi(M.Tech), DEPARTMENT OF ECE,ASR COLLEGE,Tanuku

K.Rajasekhar M.Tech, Asst.Professor in ECE, ASR COLLEGE,Tanuku

ABSTRACT:

The main objective of this project is to design a fault diagnoses system for detection of any software or hardware or permanent failures in the embedded read only memories. BIST controller, along with row selector and column selector is designed to meet requirements of at speed test thus enabling detection of timing defects. The proposed approach offers a simple test flow and does not require intensive interactions between a BIST controller and a tester. The scheme rests on partitioning of rows and columns of the memory array by employing low cost test logic. It is designed to meet requirements of at-speed test thus enabling detection of timing defects

INTRODUCTION:

Non-volatile memories are among the oldest programmable devices, but continue to have many critical uses. ROM, PROM, EPROM, EEPROM, and flash memories have proved to be very useful in a variety of applications. Traditionally, they were primarily used for long-term data storage, such as look-up tables in multimedia processors or permanent code storage in microprocessors. Due to the high area density and new sub micrometer technologies involving multiple metal layers, ROMs have also gained popularity as a storage solution for low voltage/ low-power designs. Moreover, different methods such as selective pre-charging, minimization of non-zero items, row(s) inversion, sign magnitude encoding, and difference encoding are being employed to reduce the capacitance and/or the switching activity of bit and word lines. Most large

application-specific integrated circuits (ASICs) use scan as a fundamental design for test (DFT) methodology. It has been observed that the amount of test data required to test one gate in a large design can exceed 1 Kb. This depends on several factors, such as the design style, fault models used, and capabilities of the automatic test pattern generation (ATPG) tool used. However, even using state-of-the-art ATPG tools, several gigabits of test data may be required for a multi-million gate design. Testers have a limited number of channels designed to drive scan chains, typically around 8. The speed of loading is also limited by the maximum scan frequency, usually around 10 to 50 MHz. The large volume of test data creates two problems for testers: capacity and test application time. Very often, testers do not have enough memory to store the entire test set to cover stuck-at, transition, and path delay faults. In some cases, the available memory is not even large enough to store a complete test set for stuck-at faults. In this case, either very time consuming reloads are required, or only a subset of the test vectors is applied with the corresponding reduction of fault coverage. The volume of test data directly impacts the test application time. The increasingly large volume of test data and limited throughput of the scan interface between the tester and the design creates a bottleneck. Even today, test application time can be several seconds. The cost of tester time is typically 25 to 50 cents per second. For high-end testers used in testing state-of-the-art microprocessors, it has been reported that the tester amortization cost could as high as \$6000 per hour. Note that manufacturing test is applied to every device multiple times, at different voltage levels, at the wafer, packaged device, etc. The manufacturing test cost is incurred for every manufactured

device and might be as high as 25-30% of the total manufacturing cost. Logic built-in self-test (BIST) is based on scan as the fundamental DFT methodology. Initially, the predominant compelling reason for the adoption of BIST was the requirement to perform in-field testing. Recently, there has been growing interest in BIST as it can reduce the cost of manufacturing test as well as improve the quality of the test by providing at-speed testing capability.

TEST LOGIC ARCHITECTURE:

The following Fig shows the salient architectural features of a ROM. Every row consists of M words, each B -bit long. Bits belonging to one word can be either placed one after another or interleaved forming segments, as illustrated in the figure. Decoders guarantee the proper access to memory cells in either a fast row or a fast column addressing mode, i.e., with row numbers changing faster than word numbers or vice versa. Table I gives the main memory parameters that we will use in the next sections of this paper. It is worth noting that algorithms proposed in this paper do not impose any constraints on the addressing scheme so that the memory array can be read using either increasing or decreasing address order.

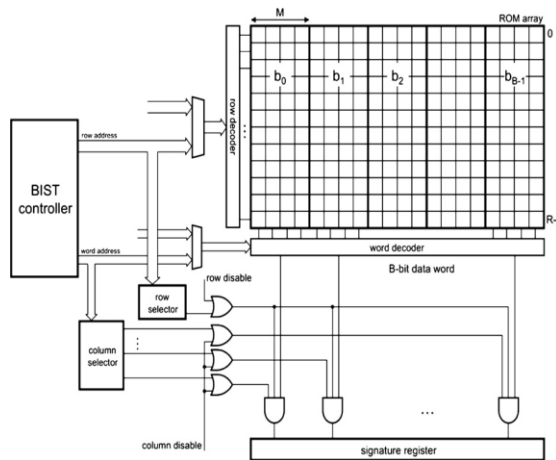


Fig: Schematic diagram

COLLECTION OF DIAGNOSTIC DATA:

The schematic diagram summarizes the architecture of a test environment used to collect diagnostic data from the ROM arrays. In addition to a BIST controller, it consists of two modules and gating logic that allow selective observation of rows and columns, respectively. Assuming permanent failures, the BIST controller sweeps through all ROM addresses repeatedly while the row and column selectors decide which data arriving from the memory rows and/or columns is actually observed by the signature register. Depending on a test scenario, test responses are collected in one of the following test modes.

1) Row disable = 0 and column disable = 1; the row selector may enable all bits of the currently received word, thereby selecting a given row; this mode is used to diagnose row failures and, in some cases, single cell faults.

2) Row disable = 1 and column disable = 0; assertion of the row disable signal effectively gates the row selector off; the column selector takes over as it picks a subset of bit lines to be observed (this corresponds to selecting desired columns and is recommended to diagnose column and single cell failures).

3) Row disable = 0 and column disable = 0; de-asserting both control lines allows observation of memory cells located where selected rows and columns intersect.

SIGNATURE REGISTER:

A signature register is used to collect all test responses arriving from selected memory cells. The register is reset at the beginning of every run (test step) over the address space. Similarly, the content of the register is downloaded once per run. A multiple input ring generator (MIRG) driven by the outputs of gating logic is used to implement the signature register. The design of Fig features the injector network handling the increasing number of input channels. It is worth noting that connecting each input to uniquely selected stages of the compactor makes it possible to recognize errors arriving from different input channels. This

technique visibly improves diagnostic resolution, as is demonstrated in the following sections.

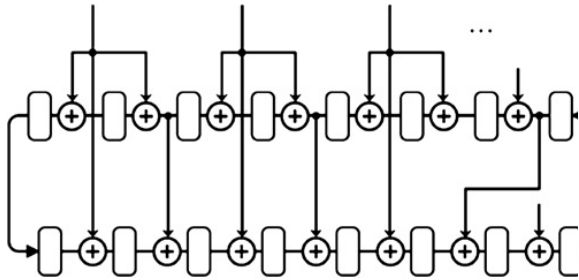


Fig: MIRG based signature register

ROW AND COLUMN SELECTOR:

In this section, we introduce several hardware solutions for row and column selection. In particular, after presenting separate row and column selectors that implement a deterministic partitioning of a ROM array, we introduce a scheme that allows one to partition rows and columns simultaneously.

Row Selection:

We start by introducing the general structure of the row selector shown in Fig(a). Essentially, it is comprised of four registers. The up counters *partition* and *group*, each of size $n = \lfloor 0.5 \log_2 R \rfloor$, keep indexes of the current partition and the current group, respectively. They act as an extension of the row address register that belongs to the BIST controller (the leftmost part of the counter in Fig. 4). A linear feedback shift register (LFSR) with a primitive characteristic polynomial implements a *diffractor* providing successive powers of a generating element of $GF(2^n)$, which are subsequently used to selectively invert data arriving from the partition register. The same register can be initialized when its input load is activated. Similarly, one can initialize a down counter called *offset* by asserting its input load. In principle, the circuit shown in Fig. 4 implements the following formula used to determine members *r* of partition *p* within group *g*: $r = S \cdot k + (p \cdot (g \otimes k))$, $k = 0, 1, \dots, P - 1$ (1) where *S* is the size of partition, *P* is the number of partitions, \cdot is a bit-wise addition modulo 2, and $g \otimes k$ is a state that the diffractor reaches after *k* - 1 steps assuming that its initial state was

g. If $k = 0$, then $g \otimes k = 0$. For example, (1) yields successive partitions of Fig. 3 for $S = 4$ and $k = 0, 1, 2, 3$, assuming that the diffractor cycles through the

following states: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$. Let $g = 3$ and $p = 2$. Then

we have

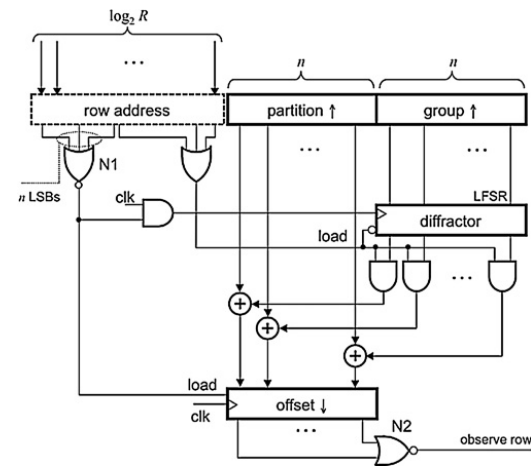
$$k = 0: r = 4 \cdot 0 + (2 \cdot (3 \otimes 0)) = 0 + (2 \cdot 0) = 2$$

$$k = 1: r = 4 \cdot 1 + (2 \cdot (3 \otimes 1)) = 4 + (2 \cdot 3) = 5$$

$$k = 2: r = 4 \cdot 2 + (2 \cdot (3 \otimes 2)) = 8 + (2 \cdot 1) = 11$$

$$k = 3: r = 4 \cdot 3 + (2 \cdot (3 \otimes 3)) = 12 + (2 \cdot 2) = 12.$$

With the ascending row address order, selection of rows within a partition, a group, and finally the whole test is done as follows. The offset counter is reloaded periodically every time the *n* least significant bits of the row address register become zero (this is detected by the NOR gate N1). Once loaded, the counter is decremented to reach the all-0 state after $p \oplus (g \otimes)$ cycles. This is detected by the NOR gate N2 associated with the counter. Hence, its asserted output enables observation of a single row within every *S* successive cycles. As indicated by (1), the initial values of the offset counter are obtained by adding the actual partition number to the current state of the diffractor.



Fig(a): Row selector

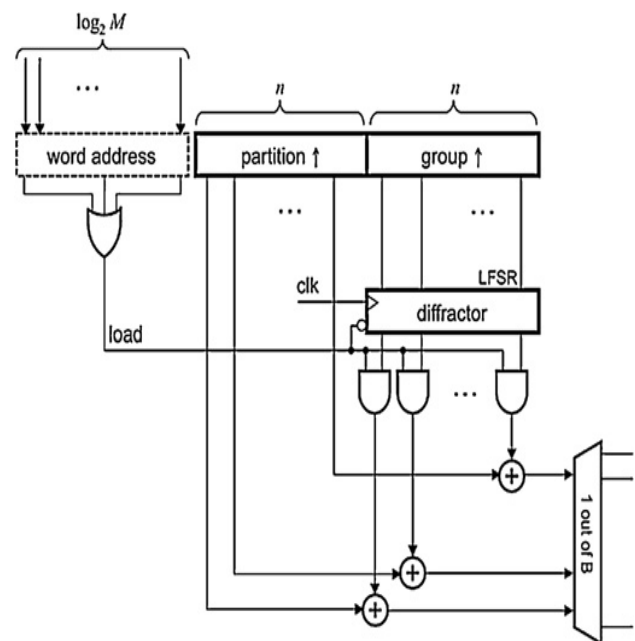
The diagram depicts the output of gate N2. As can be seen, the diffractor is loaded at the beginning of a test run (*row address* = 0) with the group number 3 (112), and then it changes its state every four cycles by following the trajectory $1 \rightarrow 2 \rightarrow 3$. At the same cycles, i.e., 0, 4, 8, and 12, the offset counter is reloaded with the sum of the partition number 2 (102) and the previous state of the diffractor, except for the first load, when only the partition number goes to the offset counter as the outputs of the AND gates are set to 0. After initializing, the offset register counts down and reaches zero at cycles 2, 5, 11, and 12, which yields an active signal on line "observe row" resulting, in turn, in observing data from the memory rows with addresses 2, 5, 11, and 12, respectively.

Column Selection:

Fig(b) shows the column selector used to decide, in a deterministic fashion, which columns should be observed. Its architecture resembles the structure of the row selector as both circuits adopt the same selection principles. The main differences include the use of a BIST column address register and a diffractor clocking scheme. Moreover, the offset counter is now replaced with a combinational column decoder, which allows selection of one out of B outputs of the word decoder. It is worth noting that the diffractor advances every time the column address increments. Its content added to the partition number yields a required column address in a manner similar to that of the row selection.

If the size B of the memory word is equal to M (the number of words per row), it suffices to select one out of B columns at a time to cover all columns of the memory array for one partition group. Typically, however, we observe that $B > M$. This requires in a certain case certain columns would always be observed together, thereby precluding an effective partitioning. Consequently, the output column decoder is divided into t smaller "1 out of B " decoders fed by phase shifters (PS), and then the diffractor, as shown in Fig. 7. The phase shifters transform a given input combination in such a way that the resultant output values are spread in regular intervals over the diffractor state trajectory. Fig.

8 demonstrates this scenario for a 3-bit diffractor driving three phase shifters and using primitive polynomial $x^3 + x + 1$. Let the diffractor be initialized to the value of 1. re than one column of each word to be selected at a time, as far as the single test run is concerned for every partition. The number t of columns observed simultaneously can be determined by dividing the maximal number of columns in a partition, which is $2n$, by the number M of memory words per row $\tau = 2n/M$. It is important to note that columns observed in parallel cannot be handled by a single "t out of B " selector, as in such

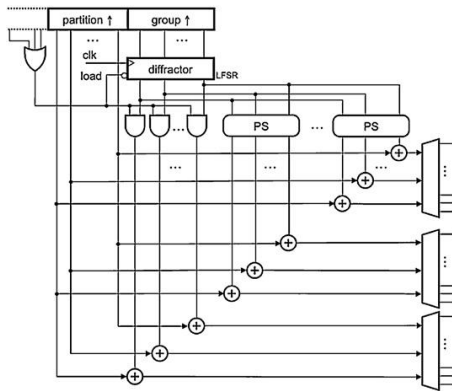


fig(b). column selector.

The phase shifters PS1, PS2, and PS3 are then to output states of the original trajectory, but starting with the values of 4, 6, and 5, respectively. When various partition groups are examined, the diffractor traverses the corresponding parts of its state space while the phase shifters produce appropriate values that ensure generation of all $2n - 1$ combinations. The missing all-0 state is again obtained by means of AND gates.

Word Address	Partition	Group	Column Decoder 0	Column Decoder 1	Observed Columns
0	00	01	00	11	0, 14
1	00	01	10	01	5, 11
0	01	01	01	10	2, 12
1	01	01	11	00	7, 9
0	10	01	10	01	4, 10
1	10	01	00	11	1, 15
0	11	01	11	00	6, 8
1	11	01	01	10	3, 13
0	00	10	00	01	0, 10
1	00	10	11	10	7, 13
0	01	10	01	00	2, 8
1	01	10	10	11	5, 15
0	10	10	10	11	4, 14
1	10	10	01	00	3, 9
0	11	10	11	10	6, 12
1	11	10	00	01	1, 11

COLUMN PARTITIONING



Fig(C).

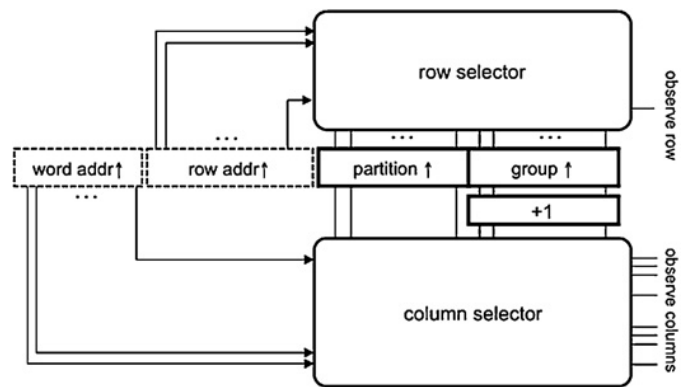
ENHANCED COLUMN SELECTOR

Combined Row and Column Selection:

In order to reduce the area overhead, some components of the row selector and the column selector can be shared. The circuit by which this concept is implemented is shown in Fig(d) where the partition and group registers feed both selectors. Since the word address increments prior to the row address, the memory array is read in the fast column.

As no interaction between control signals arriving from the word and row address registers is needed, the scheme enables reading the memory array in the fast row mode as well, after exchanging the row and column address registers. Furthermore, the combined row and

column selector is designed in such a way that none of the components require clock faster than the one used to increment either the word or row address register. As a result, the proposed scheme allows reading memory at-speed, and thus detection of timing defects. Finally, as the combined selector makes it possible to collect the row and column signatures in parallel, such an approach allows one to reduce the diagnostic time by half. In this mode, however, two signature registers are required.



Fig(d) combined row and column selector

Trellis Selection:

Given $x + 1$ groups of signatures, the selection schemes presented earlier allow one to identify correctly up to either x failing rows or x failing columns, exclusively. The actual failure may comprise, however, faults occurring in rows and columns at once. Fig.(a) illustrates a failure that consists of a single stuck-at column and a single stuck-at row. The black dots indicate failing cells assuming a random fill—note that some cells of the faulty row and column store the same logic values as those forced by the fault. If diagnosed by using separate selection of rows and columns, such a fault would affect most of signatures as cells belonging to the failing column make almost all row signatures erroneous, and cells of the failing row would render almost all column signatures erroneous, as well. Collecting signatures in so-called trellis mode provides a solution to this problem by partitioning rows and columns simultaneously. Selecting rows and columns in parallel

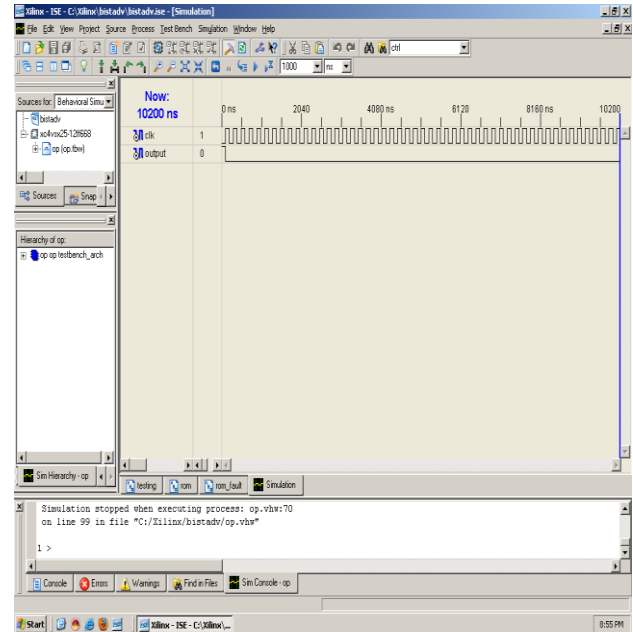
substantially reduces the number of observed cells, thereby increasing a chance to record fault-free signatures and to sieve successfully failing rows and columns.

Fig.(b) and (c) are examples of trellis compaction in the presence of a single row-single-column failure. Observed are memory cells located at the intersections of rows and columns only. The resultant signatures are therefore likely to be error-free, as shown in Fig.(b). Consequently, the selected rows and columns can be declared fault free. When the selected cells come across the failing row or the failing column, one may expect to capture at least one error, as in Fig(c). There is an intrinsic rows-to-columns correlation in the trellis selection mode. In particular, using the same characteristic polynomial for both diffractors of the combined selector, and initializing them with the same group number causes predictable changes in this dependency—many row-column pairs always end up in the same partitions. As a result, the diagnostic algorithm is unable to distinguish fault-free rows and columns from defective ones since they are permanently paired by the selection scheme.

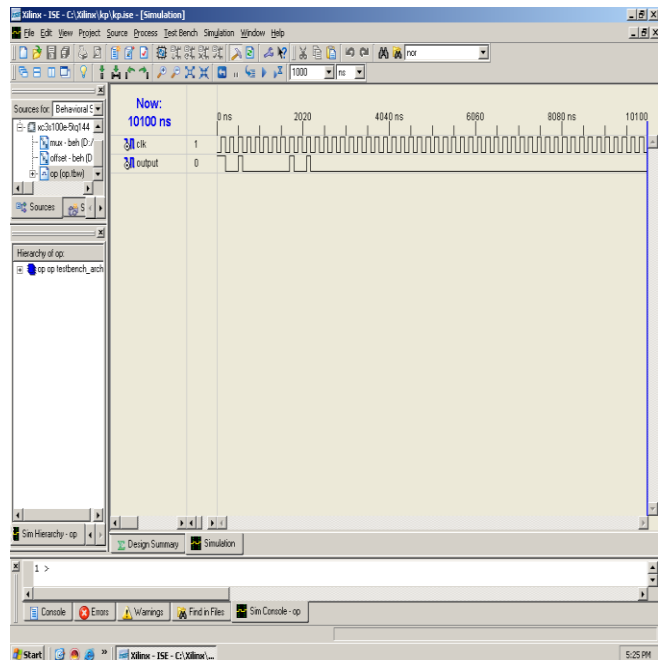
The row and column selectors employ identical diffractors with a primitive polynomial $x^5 + x^2 + 1$. Each entry to the table provides the number of row-column pairs (out of total 10242) that occur k times within the same partitions for arbitrarily chosen 3, 4, 5, and 32 partition groups. As shown in the table, 1024 rows and columns get always to the same partition regardless of the number of partition groups. A thorough analysis of these results has further revealed that every row is permanently coupled with a certain column due to this particular selection mechanism. It appears, however, that a simple n -bit arithmetic incremter (a module “+1” in Fig(d) placed between the group register and one of the diffractors alters this row-column relationship so that the resultant correlation is significantly decreased. This is confirmed by the experimental data gathered in the lower part of Table III. We assume here that the column diffractor is initialized with the group number increased arithmetically by 1. As can be seen, the enhanced selection technique clearly reduces the number of the row-column pairs that always end up in the same partitions. Interestingly, the number of such pairs is equal to the number of partitions in a group (32). This is due to the zero

states that are always contributed by the AND gates at the beginning of each partition

SIMULATION REPORT FOR A FAULT FREE MEMORY:



SIMULATION REPORT FOR FAULT MEMORY:



CONCLUSION:

In this paper, we proposed a new fault diagnosis scheme for embedded read-only memories. It reduces the diagnostic data that needs to be scanned out during ROM test such that the minimum information to recover the failure data is preserved, and the time to unload the data is minimized. The presented approach allows an uninterrupted collection and processing of test responses at the system speed. This has been achieved by using low-cost on-chip selection mechanisms, which are instrumental in very accurate and time efficient identification of failing rows, columns, and single memory cells. In particular, the scheme employs the original designs of row and column selectors with phase shifters controlling the way the address space is traversed.

Furthermore, the new combined selection logic allows the scheme to collect test results in parallel (leading to shorter test time) without compromising quality of diagnosis. Results of experiments performed on several memory arrays for randomly generated failures clearly confirm high accuracy of diagnosis of the scheme provided the signature registers and the proposed selection logic are properly tuned to guarantee a desired diagnostic resolute

REFERENCES:

- [1] R. D. Adams, *High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test*. New York: Kluwer, 2003.
- [2] D. Appello, V. Tancorre, P. Bernardi, M. Grosso, M. Rebaudengo, and M. Sonza Reorda, "Embedded memory diagnosis: An industrial workflow," in *Proc. ITC*, 2006, paper 26.2.
- [3] S. Barbagallo, A. Burri, D. Medina, P. Camurati, P. Prinetto, and M. Sonza Reorda, "An experimental comparison of different approaches to ROM BIST," in *Proc. Eur. Comput. Conf.*, 1991, pp. 567–571.
- [4] I. Bayraktaroglu and A. Orailoglu, "The construction of optimal deterministic partitioning in scan-based BIST fault diagnosis: Mathematical foundations and cost-effective implementations," *IEEE Trans. Comput.*, vol. 54, no. 1, pp. 61–75, Jan. 2005.
- [5] T. J. Bergfeld, D. Niggemeyer, and E. M. Rudnick, "Diagnostic testing of embedded memories using BIST," in *Proc. DATE*, 2000, pp. 305–309.
- [6] T. Boehler and G. Lehmann, "Using data compression for faster testing of embedded memory," U.S. Patent 6 950 971, Sep. 27, 2005.
- [7] J. T. Chen and J. Rajski, "Method and apparatus for diagnosing memory using self-testing circuits," U.S. Patent 6 421 794, Jul. 16, 2002.
- [8] J. T. Chen, J. Rajski, J. Khare, O. Kebichi, and W. Maly, "Enabling embedded memory diagnosis via test response compression," in *Proc. VTS*, 2001, pp. 292–298.
- [9] J. T. Chen, J. Khare, K. Walker, S. Shaikh, J. Rajski, and W. Maly, "Test response compression and bitmap encoding for embedded memories in manufacturing process monitoring," in *Proc. ITC*, 2001, pp. 258–267.
- [10] D. W. Clark and L.-J. Weng, "Maximal and near-maximal shift register sequences: Efficient event counters and easy discrete logarithms," *IEEE Trans. Comput.*, vol. 43, no. 5, pp. 560–568, May 1994.
- [11] X. Du, N. Mukherjee, W.-T. Cheng, and S. M. Reddy, "Full-speed fieldprogrammable memory BIST architecture," in *Proc. ITC*, 2005, paper 45.3.