Error Identification In RAM Using Input Vector Monitoring Concurrent BIST Architecture

Mr. L.RAJU
M.Tech Student
SR Engineering College, Warangal, India

Mr. V.RAJESH
Assistant professor
SR Engineering College, Warangal, India

Abstract— **Input vector monitoring concurrent built-in self test (BIST) schemes perform testing during the normal operation of the Random Access Memory without imposing a need to set the RAM offline to perform the test. These schemes are evaluated based on the hardware overhead and the concurrent test latency (CTL), i.e., the time required for the test to complete, whereas the circuit operates normally. In this brief, we present a novel input vector monitoring concurrent BIST scheme, which is based on the idea of monitoring a set (called window) of vectors reaching the circuit inputs during normal operation, and the use of a static-RAM-like structure to store the relative locations of the vectors that reach the circuit inputs in the examined window; the proposed scheme is shown to perform significantly better than previously proposed schemes with respect to the hardware overhead and CTL tradeoff.**

Index Terms— **Built-In Self-Test; Design for Testability; Testing;**

I. INTRODUCTION

Built-in self test (BIST) techniques constitute a class of schemes that provide the capability of performing at-speed testing with high fault coverage, whereas simultaneously they relax the reliance on expensive external testing equipment. Hence, they constitute an attractive solution to the problem of testing VLSI devices [1]. BIST techniques are typically classified into offline and online. Offline architectures operate in either normal mode (during which the BIST circuitry is idle) or test mode. During test mode, the inputs generated by a test generator module are applied to the inputs of the circuit under test (RAM) and the responses are captured into a response verifier (RV). Therefore, to perform the test, the normal operation of the CUT is stalled and, consequently, the performance of the system in which the circuit is included, is degraded.

Input vector monitoring concurrent BIST techniques [2]–[10] have been proposed to avoid this performance degradation. These architectures test the RAM concurrently with its normal operation by exploiting input vectors appearing to the inputs of the CUT; if the incoming vector belongs to a set called active test set, the RV is enabled to capture the RAM response. The block diagram of an input vector monitoring concurrent BIST architecture is shown in Fig. 1. The CUT has n inputs and m outputs and is tested exhaustively; hence, the test set size is $N = 2^n$. The technique can operate in either normal or test mode, depending on the value of the signal labeled T/N.

During normal mode, the vector that drives the inputs of the RAM (denoted by $d[n:1]$ in Fig. 1) is driven from the normal input vector ($A[n:1]$). A is also driven to a concurrent BIST unit (CBU), a hit has occurred. In this case, A is removed from the active test set and the signal response verifier enable (rve) is issued, to enable the m-stage RV to capture the CUT response to the input vector [1].

The concurrent test latency (CTL) of an input vector monitoring scheme is the mean time (counted either in number of clock cycles or time units) required to complete the test while the CUT operates in normal mode.

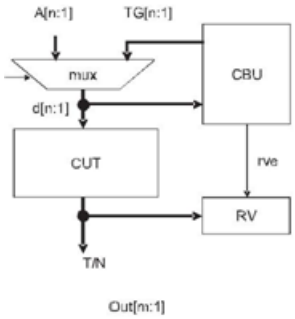

Fig. 1. Input vector monitoring concurrent BIST.

In this brief, a novel input vector monitoring concurrent BIST scheme is proposed, which compares favorably to previously pro-posed schemes [2]–[7] with respect to the hardware overhead/CTL tradeoff. This brief is organized as follows. In Section II, we introduce the proposed approach and in Section III, we calculate its hardware overhead. In Section IV, we compare the proposed scheme with previ-ously proposed input vector monitoring concurrent BIST techniques. A case study for the concurrent testing of ROM modules is presented in Section V. Finally, Section VI summarizes the conclusion of this brief.

II. PROPOSED SCHEME

Let us consider a combinational CUT with n input lines, as shown in Fig. 2; hence the possible input vectors for this CUT are 2n . The proposed scheme is based on the idea of monitor-ing a window of vectors, whose size is W , with W = 2w , where w is an integer number w < n. Every moment, the test vectors belonging to the window are monitored, and if a vector performs a hit, the RV is enabled.

The bits of the input vector are separated into two distinct sets comprising w and k bits, respectively, such that $w + k = n$. The k (high order) bits of the input vector show whether the input vector belongs to the window under consideration. The w remaining bits show the relative location of the incoming vector in the current window. If the incoming vector belongs to the current window and has not been received during the examination of the current window, we say that the vector has performed a hit and the RV is clocked to capture the CUT's response to the vector. When all vectors that belong to the current window have reached the CUT inputs, we proceed to examine the next window.

The module implementing the idea is shown in Fig. 2. It operates in one out of two modes, normal, and test, depending on the value of the signal T /N. When T /N = 0 (normal mode) the inputs to the CUT are driven by the normal input vector. The inputs of the CUT are also driven to the CBU as follows: the k (high order) bits are driven to the inputs of a k-stage comparator; the other inputs of the comparator are driven by the outputs of a k-stage test generator TG. The proposed scheme uses a modified decoder (denoted as m_dec in Fig. 2) and a logic module based on a static-RAM (SRAM)-like cell, as will be explained shortly.

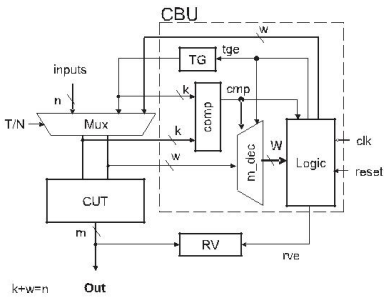

Fig. 2. Proposed architecture.

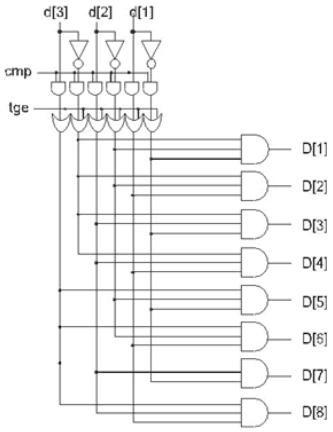

Fig. 3. Modified decoder design used in the proposed architecture.

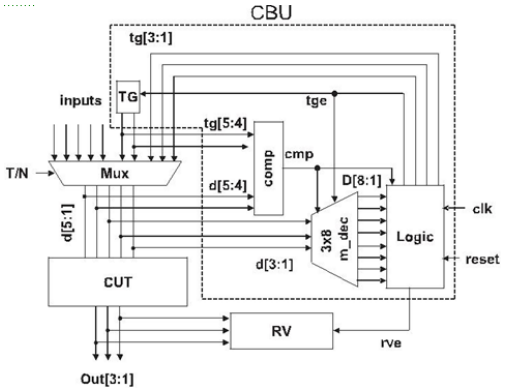

Fig. 4. Proposed architecture for n = 5, w = 3, and k = 2.

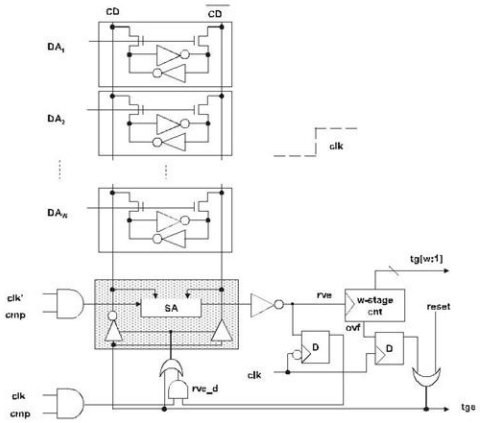

Fig. 5. Design of the logic module.

tge is disabled and cmp is enabled, the module operates as a normal decoding structure.

The architecture of the proposed scheme for the specific case n = 5, k = 2, and w = 3, is shown in Fig. 4.

The module labelled logic in Fig. 4 is shown in Fig. 5.

It comprises W cells (operating in a fashion similar to

the SRAM cell), a sense amplifier, two D flip-flops, and a w-stage counter (where w = log2 W). The overflow signal of the counter drives the tge signal through a unit flip-flop delay. The signals clk_ and clock (clk) are enabled during the active low and high of the clock, respectively. In the sequel, we have assumed a clock that is active during the second half of the period, as shown in Fig. 5.

In the sequel, we describe the operation of the logic module, presenting the following cases: 1) reset of the module; 2) hit of a vector (i.e., a vector belongs in the active window and reaches the CUT inputs for the first time); 3) a vector that belongs in the current window reaches the CUT inputs but not for the first time; and 4) tge operation (i.e., all cells of the window are filled and we will proceed to examine the next window).

A. Reset of the Module

At the beginning of the operation, the module is reset through the external reset signal. When reset is issued, the tge signal is enabled and all the outputs of the decoder (Fig. 3) are enabled. Hence, DA1, DA2, . . . , DAW are one; furthermore, the CD_ signal is enabled; therefore, a one is written to the right hand side of the cells and a zero value to the left hand side of the cells.

B. Hit of Vector (i.e., Vector Belongs in the Active Window and Reaches the CUT Inputs for the First Time)

The design of the m_dec module for w = 3 is shown in Fig. 3 and operates as follows. When test generator enable (tge) is enabled, all outputs of the decoder are equal to one. When comparatot (cmp) is disabled (and tge is not enabled) all outputs are disabled. When during normal mode, the inputs to the CUT are driven from the normal inputs. The n inputs are also driven to the CBU as follows: the w low-order inputs are driven to the inputs of the decoder; the k high-order inputs are driven to the inputs of the comparator. When a vector belonging to the current window reaches the inputs of the CUT, the comparator is enabled and one of the outputs of the decoder is enabled. During the first half of the clock cycle (clk_ and cmp are enabled) the addressed cell is read; because the read value is zero, the w-stage counter is triggered through the NOT gate with output the response verifier enable (rve) signal. During the second half of the clock cycle, the left flip-flop (the one whose clock input is inverted) enables the AND gate (whose other input is clk and cmp), and enables the buffers to write the value one to the addressed cell.

Table I Calculation Of The Hardware Overhead Of The Proposed Scheme

Module	Hardware Overhead	gates
n-stage multiplexer	n × MUX_2	3 × n
m-stage ABC	m× (DFF + FA)	18 × m
k-stage comparator	k×(XOR2) + k-stage AND	5×k
w-to-W decoder	2 × W	2×W + 4×w
Logic (W)	1,5×W + SA + 2×Buf + 2×DFF + 2	1,5×W + 23
TG	8×k	8×k
w-stage counter	8×w	8×w
Total		15×n +18×m + k + 3,5×W + 23

Table II Calculation Of The Hardware Overhead Of Competing Schemes

Technique	modules	calculation
C-BIST(n,m) [4]	Mux(n) + Comp(n) + NFSR (n) + MISR(m)	17n+12m
MHSAT(n,m,K) [5]	K × (Mux(n) + Comp(n) + NFSR(n)) + K × MISR(m)	17Kn+12Km
OISAT(n,m,K) [6]	K × (Mux(n) + Comp(n) + NFSR(n)) + ABC(m)	17Kn+18m
R-CBIST(n,m,R) [2]	Comp(w) + NFSR(n-w)+ INC (w)+ RAM	11k + 25 w+18m + RAM
w-MCBIST(n,m,W) [3]	Mux(n) + Comp(n) + NFSR(n) + Logic(W) + Dcc (W) + ARC(m)	15n + 9W + 18m
SWIM(n,m,W1,W2) [7]	n × MUX21 + m× (DFF + FA) + k×(XOR2) + k-stage AND + 2 × W1 + 2 × W2 + OR(W2) + W2 × ((W1 × 4) + AND(W1) + OR(W1)) + 8×k + 8×(w1+w2)	11×n +18×m + 2×W1 + 3×W2 + 6×W1×W2

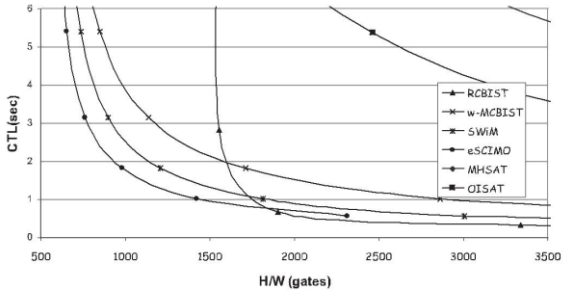

Fig. 6. Input vector monitoring techniques: comparison (n = 16, m = 16, and 100-MHz clock).

C. Vector That Belongs in the Current Window Reaches the CUT Inputs But Not for the First Time

If the cell corresponding to the incoming vector contains a one (i.e., the respective vector has reached the CUT inputs during the examination of the current window before), the rve signal is not enabled during the first half of the clock cycle; hence, the w-stage counter is not triggered and the AND gate is not enabled during the second half of the clock cycle.

D. tge Operation (i.e., All Cells of the Window are Filled and We Will Proceed to Examine the Next Window)

When all the cells are full (value equal to one), then the value of the w-stage counter is all one. Hence, the activation of the rve signal causes the counter to overflow; hence in the next clock cycle (through the unit flop delay) the tge signal is enabled and all the cells (because all the outputs of the decoder of Fig. 3 are enabled) are set to zero.

When switching from normal to test mode, the w-stage counter is reset. During test mode, the w-bit output of the counter is applied to the CUT inputs. The outputs of the counter are also used to address a cell. If the cell was empty (reset), it will be filled (set) and the RV will be enabled. Otherwise, the cell remains full and the RV is not enabled.

III. CALCULATION OF HARDWARE OVERHEAD

The hardware overhead of the proposed scheme is calculated using the gate equivalents as a metric. One gate equivalent or gate is the hardware equivalent of a two-input NAND gate. The parameters that affect the hardware overhead of the proposed scheme are n (the number of CUT inputs), m (the number of CUT outputs), and w (representing the window size) with k = n − w and $W = 2^w$.

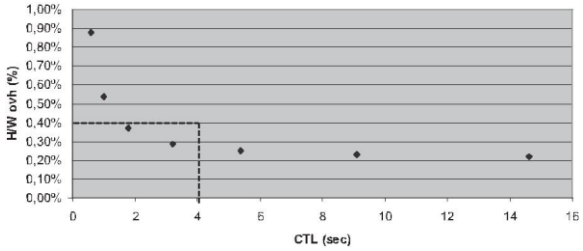

Fig. 7. Hardware overhead versus CTL for the proposed scheme (64 k × 16 ROM operating at 100 MHz).

Table III Comparison Of The Schemes For The Concurrent Testing Of Various Rom Sizes

16K ROM	CTL (secs)	0,20	0,11	0,06	0,04	0,02
	w-MCBIST	1,64%	2,52%	4,28%	7,79%	14,82%
	SWIM	1,32%	1,94%	3,14%	5,56%	10,29%
	Proposed	1,14%	1,48%	2,16%	3,53%	6,26%
	Decrease1	30,49%	41,27%	49,53%	54,69%	57,76%
	Decrease2	13,64%	23,71%	31,21%	36,51%	39,16%
64K ROM	CTL (secs)	3,18	1,82	1,03	0,57	0,31
	w-MCBIST	0,42%	0,64%	1,08%	1,96%	3,72%
	SWIM	0,34%	0,49%	0,79%	1,40%	2,58%
	Proposed	0,30%	0,38%	0,55%	0,89%	1,58%
	Decrease1	28,57%	40,63%	49,07%	54,59%	57,53%
	Decrease2	11,76%	22,45%	30,38%	36,43%	38,76%
256K ROM	CTL (secs)	50,94	29,17	16,44	9,15	5,04
	w-MCBIST	0,11%	0,16%	0,27%	0,49%	0,93%
	SWIM	0,09%	0,13%	0,20%	0,35%	0,65%
	Proposed	0,08%	0,10%	0,14%	0,23%	0,40%
	Decrease1	27,27%	37,50%	48,15%	53,06%	56,99%
	Decrease2	11,11%	23,08%	30,00%	34,29%	38,46%

The implementation of the scheme requires the n-stage multiplexer at the inputs of the CUT, and an m-stage order-independent RV. The necessity to have an order-independent response verification scheme stems from the fact that, during the examination of any window of vectors, the order that the vectors will perform hit, is not fixed. The accumulator-based compaction of the responses is an order-independent response verification technique [11], [12] that has been shown to have aliasing properties similar to the best compactors based on cellular automata and multiple input signature registers. Furthermore, the accumulator-based compaction requires only a one-bit full adder (FA) and a D-type flip-flop (DFF) for each CUT output. Therefore, the accumulator-based compaction of the responses is used for the implementation of the proposed scheme. In Table I, the hardware overhead of the various modules of the proposed scheme has been calculated, following Figs. 2–5. We have estimated the overhead of one cell as 1.5 gate equivalents, the overhead of a tristate buffer as one gate, and the overhead of a sense amplifier as three gates.

IV. COMPARISONS

To evaluate the presented scheme, we compare it with the input vector monitoring concurrent BIST techniques proposed hitherto. Because for the same window size *W* , the CTL is equal to the scheme proposed in [3] and [7] for the same window size, in the sequel, we proceed using the CTL calculated in these publications.

C-BIST [4] was the first input vector monitoring concurrent BIST technique proposed, and suffers from long CTL; therefore modifications have been proposed, Multiple Hardware Sig-nature Analysis Technique (MHSAT) [5], Order Independent Signature Analysis Technique (OISAT) [6], RAM-based Concur-rent BIST (R-CBIST) [2], Window-Monitoring Concurrent BIST (w-MCBIST) [3], and Square Windows Monitoring Concurrent BIST (SWIM) [7]. The comparisons will be performed with respect to the value of the CTL and the hardware overhead.

In Table II, we provide the formulas that we used to calculate the hardware overhead of MHSAT, OISAT ($K = 2^k$), R-CBIST, w-MCBIST, and the SWIM scheme. The cells used are two-input XOR gate (XOR_2), n-input AND gate(AND_n), n-input NAND gate ($NAND_n$), n-input OR gate (R_n), n-input NOR gate (NOR_n), DFF, FA and two-to-one multiplexer (MUX_{21}).

In Fig. 6, the CTL is presented (in time units, i.e., seconds) as a function of the hardware overhead (in gate equivalents) for R-CBIST, w-MCBIST, SWIM, and the proposed architecture. A CUT with n = 16 inputs and m = 16 outputs has been considered. The points have been connected with power trend lines. From Fig. 6, we can observe that for the same hardware overhead, the proposed scheme achieves shorter CTL than the previously proposed schemes. Thus, we conclude that the proposed scheme is more efficient than MHSAT, OISAT, w-MCBIST, and SWIM with respect to the hardware overhead—CTL tradeoff. For example, if a CTL of 3 s is required, then the proposed scheme requires 761 gates, whereas SWIM (the second better scheme) requires 898 gates, i.e., 16% more and w-MCBIST requires 1136 gates, i.e., 33% more.

Furthermore, if the demand for CTL is not < 0.8 s, the proposed scheme achieves the same CTL with R-

CBIST with significantly less hardware overhead. For example, for CTL = 3 s, the hardware overhead required by the proposed scheme is 761 gates, whereas the same number for R-CBIST is 1553, i.e., 102% more.

V. CASE STUDY: COMPARATIVE CONCURRENT TESTING OF ROM MODULES

ROM modules require high-quality testing because they constitute critical parts in complex circuits, therefore testing schemes for ROMs use exhaustive application of input patterns, which has been proved to cover all logically testable combinational faults [14]. For the calculations, we have considered a ROM cell to be equivalent to 1/4 gate (as in [13]). For the case considered in Fig. 6 (a 64 k × 16 word memory), the overhead of the ROM is calculated by multiplying the number of cells (64 k × 16 = 65 536 × 16 = 1 048 576) with $^{1/4}$, giving 262 144 gates. Fig. 7 shows the percentage of hardware overhead of the proposed scheme as a function of the CTL assuming a 100-MHz clock. From Fig. 7, we can observe that the concurrent test can be completed within < 4 s with < 0.4% overhead, as shown with the dashed line. It should be noted that, because of problems with layout, the actual area overhead may be higher; however, the above calculations give an indicative order of magnitude for the relative hardware overhead of the proposed scheme.

In Table III, we compare the w-MCBIST, SWIM, and the proposed scheme for the concurrent testing of ROMs with representative sizes. We have not considered R-CBIST in these comparisons, because for these values of the CTL, the R-CBIST scheme does not give favourable results, as shown in Fig. 6. For the calculations, we have considered ROMs with 16-bit words and a 100-MHz clock. In Table III, for every ROM size, we present a group of six rows. In the first row of each group, we present the CTL (s); in the three following rows of each group, we present the hardware overhead of each scheme as a percentage of the hardware overhead of the ROM module. In the last two rows of every group, we present the decrease of the proposed scheme over the w-MCBIST scheme (denoted Decrease1) and over the SWIM scheme (denoted Decrease2).

From Table III, the following conclusions can be drawn.

1) The hardware overhead of the proposed scheme is lower than the other schemes for all the entries of the table.

2) The decrease in hardware overhead obtains higher as the CTL decreases; for example, in the 256-k ROM group, the decrease (compared with SWIM) is 11.11% when a CTL = 50.94 s is required, it climbs up to 38.46% when the required CTL is ∼5 s.

VI. CONCLUSION

BIST schemes constitute an attractive solution to the problem of testing VLSI devices. Input vector monitoring concurrent BIST schemes perform testing during the circuit normal operation without imposing a need to set the circuit offline to perform the test, therefore they can circumvent problems appearing in offline BIST techniques. The evaluation criteria for this class of schemes are the hardware overhead and the CTL, i.e., the time required for the test to complete, while the circuit operates normally. In this brief, a novel input vector monitoring concurrent BIST architecture has been presented, based on the use of a SRAM-cell like structure for storing the information of whether an input vector has appeared or not during normal operation. The proposed scheme is shown to be more efficient than previously proposed input vector monitoring concurrent BIST techniques in terms of hardware overhead and CTL.

VII. REFERENCES

[1] E. J. McCluskey, "Built-in self-test techniques," IEEE Design Test Comput., vol. 2, no. 2, pp. 21–28, Apr. 1985.

[2] I. Voyiatzis, A. Paschalis, D. Gizopoulos, N. Kranitis, and C. Halatsis, "A concurrent BIST architecture based on a self-testing RAM," IEEE Trans. Rel., vol. 54, no. 1, pp. 69–78, Mar. 2005.

[3] I. Voyiatzis and C. Halatsis, "A low-cost concurrent BIST scheme for increased dependability," IEEE Trans. Dependable Secure Comput., vol. 2, no. 2, pp. 150–156, Apr. 2005.

[4] K. K. Saluja, R. Sharma, and C. R. Kime, "A concurrent test-ing technique for digital circuits," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 7, no. 12, pp. 1250–1260, Dec. 1988.

[5] R. Sharma and K. K. Saluja, "Theory, analysis and implementation of an on-line BIST technique," VLSI Design, vol. 1, no. 1, pp. 9–22, 1993.

[6] K. K. Saluja, R. Sharma, and C. R. Kime, "Concurrent com-parative built-in testing of digital circuits," Dept. Electr. Comput. Eng., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. ECE-8711, 1986.

[7] I. Voyiatzis, T. Haniotakis, C. Efstathiou, and

H. Antonopoulou, "A con-current BIST architecture based on monitoring square windows," in Proc. 5th Int. Conf. DTIS, Mar. 2010, pp. 1–6.

[8] M. A. Kochte, C. Zoellin, and H.-J. Wunderlich, "Concurrent self-test with partially specified patterns for low test latency and overhead," in Proc. 14th Eur. Test Symp., May 2009, pp. 53–58.

[9] S. Almukhaizim and Y. Makris, "Concurrent error detection methods for asynchronous burst mode machines," IEEE Trans. Comput., vol. 56, no. 6, pp. 785–798, Jun. 2007.

[10] S. Almukhaizim, P. Drineas, and Y. Makris, "Entropy-driven parity tree selection for low-cost concurrent error detection," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 25, no. 8, pp. 1547–1554, Aug. 2006.

[11] J. Rajski and J. Tyszer, "Test responses compaction in accu-mulators with rotate carry adders," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 12, no. 4, pp. 531–539, Apr. 1993.

[12] I. Voyiatzis, "On reducing aliasing in accumulator-based compaction," in Proc. Int. Conf. DTIS, Mar. 2008, pp. 1–12.

[13] L. R. Huang, J. Y. Jou, and S. Y. Kuo, "Gauss-elimination-based generation of multiple seed-polynomial pairs for LFSR," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 16, no. 9, pp. 1015–1024, Sep. 1997.

[14] Y. Zorian and A. Ivanov, "An effective BIST scheme for ROM's," IEEE Trans. Comput., vol. 41, no. 5, pp. 646–653, May 1992.

