© 2017 IJSRST | Volume 3 | Issue 8 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Input Vector Observing Simultaneous BIST Architecture Utilizing SRAM Cells

P. Kanimozhi*[1], S. Ramalingam[2], S. Murugesan[3], N. Ramya[4]
*[1]Electronics and Instrumentation Engineering, Trichy, Tamilnadu, India
[2]Electronics and Communication Engineering, Trichy, Tamilnadu, India
[3,4]Electrical and Electronics Engineering, K. Ramakrishnan College of Technology, Trichy, Tamilnadu, India

ABSTRACT

To perform the testing during the normal operation of the circuit without the set of the circuit in offline mode to perform the test by input vector observing simultaneous BIST engineering utilizing SRAM cells. These testing are evaluated on the basics of the hardware overhead and concurrent test latency (CTL), i.e., the time required for the test to complete, whereas the circuit operates normally. A novel input vector monitoring concurrent BIST architecture test is based on the idea of monitoring a set of vectors reaching the circuit inputs during the normal operation using static-RAM like structure to store the relative locations of the vectors that reach the circuit inputs in the appropriate set (window). The proposed scheme is to perform better than previously proposed schemes with respect to hardware overhead and CTL trade off.

Keywords: BIST architecture, Static-RAM cells, Concurrent test latency, Circuits under test

I. INTRODUCTION

Built –in self-test (BIST) techniques can provide the capability of performing the test in seed with high fault coverage, and also simultaneously they relax the reliance on expensive external testing equipment. Hence, they provide an attractive solution to the problem of testing VLSI circuits [1] BIST techniques are typically classified into two modes (i) offline mode (ii) online mode. Offline architecture can operate either in normal mode (i.e) during BIST circuitry is ideal or in test mode. During test mode, test generator module generates the inputs and they are given in the Circuits under Test (CUT) and the outputs are obtained through the Response Verifier (RV). To perform the test, the normal operation of the (CUT) is stalled and the circuits includes in the performance of the system is degraded.

Monitoring input vector concurrent BIST techniques [2]-[10] have been proposed to avoid the degradation performance. These architectures test the CUT 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 CUT response[11]-[19].

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 $N = 2n$. The technique can operate in either normal or test mode, depending on the value of the signal labeled *T/N*.

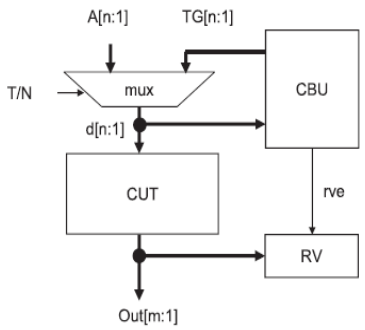

Figure 1: Input vector monitoring concurrent BIST

BIST is the capability of a circuit to test itself. BIST is an on-chip test logic that is utilized to test the functional logic of a chip[7]-[9]. A generic approach to BIST is shown in Figure BIST solution consists of a Test Pattern Generator (TPG), a circuit to be tested, a way to analyze

the results, and a way to compress those results for simplicity and handling.

With the rapid increase in the design complexity, BIST has become a major design consideration in Design-For-Testability (DFT) methods and is becoming increasingly important in today's state of the art SoCs. An achieving high fault coverage while maintaining an acceptable design, overhead and keeping the test time within limits is of almost importance. BIST help to meet the desired goals. The brief introductions of BIST architecture component are given below.

- Circuit under Test (CUT): It is the portion of the circuit tested in BIST mode. It can be sequential, combinational or a memory. It is delimited by their Primary Input (PI) and Primary Output (PO).
- Test Pattern Generator (TPG): It generates patterns for the CUT. It is a dedicated circuit or a microprocessor. The patterns may be generated in pseudorandom or deterministically.
- Test Response Analysis (TRA): It analyses the value sequence on PO and compares it with the expected output.
- BIST Controller Unit (BCU): It controls the test execution; it manages the TPG, TRA and reconfigures the CUT and the multiplexer.

A. Fault models and test pattern generation

(i) Fault models

A fault refers to physical defect in a circuit. A logical fault causes the logic value at a point in a circuit to become opposite to specified value. Non-logical faults include the rest of the faults such as the malfunction of the clock signal, power failure, and etc. The duration of fault refers to whether the fault is permanent or temporary.

Memory fault models differ significantly from the fault models used for digital logic. Line stuck-at faults, bridging faults, open faults, and transistor stuck on/off fault models work fairly well for digital logic, 7 but they aren't sufficient even to determine the functional correctness of memory. In addition to line stuck-at, bridging, and open faults, memory faults include bit-pattern, transition, and cell-coupling faults. Parametric and timing faults are also considered. Timing faults include data access and retention time, which are very important memory parameters. Sometimes, transistor stuck on/off faults, address decoder faults, and sense-amp faults are also considered.

(ii) Test pattern generation

Test pattern generation approaches for BIST schemes can be classified into four categories. It includes Exhaustive testing, Pseudo exhaustive testing, pseudorandom testing and deterministic testing. Test main function of the test pattern generator is to apply test patterns to the unit under test. As shown in Fig. 2 the test of an electronic circuit is a stimulus/response measurement: In the first step a test pattern is applied to the circuit to bring it to a defined initial state or exercise some functionality. In the second step the test pattern is processed by the circuit, and in the third step the circuit's response is checked. This test procedure is repeated for different test patterns by a test controller.

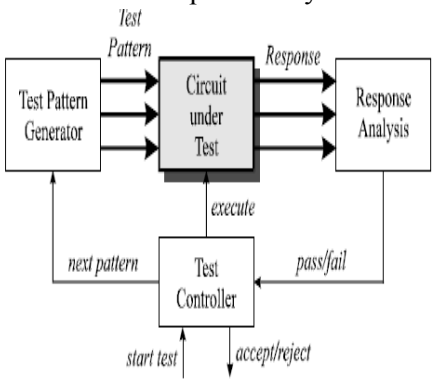

Figure 2 : Architecture of a typical test setup

(iii) Combinational versus sequential logic

The function of a combinational circuit, i.e. a circuit without storage elements, is completely described by its truth table. A combinational fault (a fault in a time-invariant logic truth table) can be detected with a single test stimulus. A straightforward check of the truth table requires $2n$ test patterns for a combinational circuit with n inputs.

With a more sophisticated strategy the number of patterns can often be substantially reduced. For the test of a combinational circuit an algorithmic solution exists (D-algorithm) that finds all detectable faults. While the runtime of this exhaustive algorithm may be prohibitive in complex applications, more recent techniques attempt to reduce test generation time or test

set size. The PODEM (Path Oriented Decision Making) algorithm is the basis for most of these modern test generation programs.

If the circuit under test, however, contains sequential logic (flip-flops and other digital storage elements), the test becomes much more complicated. Test patterns must be applied in different states; however, changing from one state into another may require a number of intermediate steps and test patterns. If a global reset of the circuit is not available, an initialization sequence or state identification is necessary, which makes the situation even worse.

(iv)Type of test sequence: Deterministic versus Pseudo-random testing

The determination of a test pattern usually involves feedback from a fault model: In a first step a list of all faults considered by the fault model is made. In the second step a test pattern is assumed and all faults detected by it are removed from the fault list. Repeating this step for new test patterns progressively reduces the fault list. Towards the end of the iteration process the contribution of new pattern decreases, since the fault list becomes small. One new pattern may be needed to remove one single fault from the list, while other new patterns do not make any contribution at all.

Although an exhaustive application of this deterministic algorithm promises the detection of all detectable faults, the duration of the search process and the length of the resulting test pattern may be excessive. A balance between coverage and cost must be found at this point. A special case of the deterministic test is the exhaustive test, for which all possible test patterns are applied. While this test produces the best possible coverage, it is impracticable for a complete VLSI chip. Considering the partitioning method described above a sum of exhaustive tests can be applied progressively to all parts of the circuit. This method is called pseudo-exhaustive test.

B. Rom testing

(i) ROM fault model

For the purpose of testing ROMs are commonly modeled either on the logical level or on the functional level:

- Logical Model: A ROM is a combinational circuit that transforms every address word applied at the input into a data word that can be read at the output. This mapping can be described by a function table.
- Functional Model: A ROM is a non-volatile memory. Reading from one given address must always produce the same data. These data are known a priori. This makes the ROM test much easier than a RAM test and allows checking data integrity in addition to the proper functionality of the device.

(ii)Structural ROM testing

Like any other combinational logic an embedded ROM can be tested structurally by test patterns via the scan path. Due to its simplicity this method is very usual. It is, however, not capable of detecting sequential faults (i.e. faults making combinational logic behave like a sequential one) that might be induced by open defects.

(iii)Signature computation

An exhaustive functional test of the ROM can be performed quite easily by reading the ROM contents and to compacting them to a signature by means of a multiple input shift register. Multiple bit faults originating from defects in decoder or output register may escape detection if aliasing occurs.

II. PROPOSED SYSTEM

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 monitoring 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[1]-[7]. 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.

We propose a novel technique for online testing, which we call Built-In Concurrent-Self- Test (BICST). BICST assumes the presence of underlying BIST resources for off-line testing. These resources are modified in such a way that they can be used for both off-line and on-line testing. By carrying out testing concurrently with the normal operation of a circuit/system, we shall show that BICST circumvents the performance degradation caused by periodic maintenance testing.

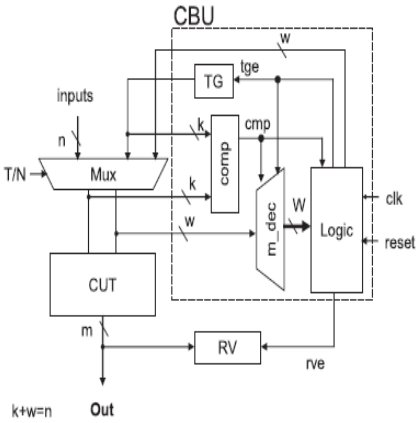

Figure 3. Proposed architecture.

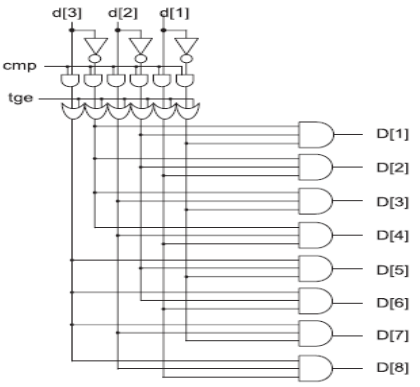

Figure 4 : Modified decoder design used in the proposed architecture.

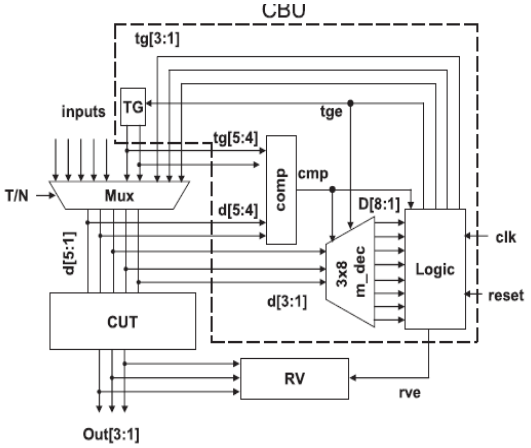

Figure 5. Proposed architecture

The modified comparator, compared to the original design, has one additional stage; the one input to the new ('high-order') stage is OUT[w]. Also, the inputs driven by in the original design are driven by two-way multiplexers. The select input of the multiplexers is driven by the RAM_BIST signal. The one input of the multiplexers is driven by A[i], and the other by the signal read-out. Especially for the 'high-order' comparing stage, the one input is driven from OUT[w], while the other input from read-out. The module operates as follows. During the testing of the RAM , each one of the signals OUT[i] is compared against the signal read-out; therefore, if some output of the RAM differs from the expected output, we conclude that a fault has occurred. During the testing of the CUT , the comparator operates identically to its original design, i.e. OUT[i] is compared against A[i] ; OUT[w] is compared against itself. In Fig. 3 we present the comparator (a), and modified comparator (b), for an AGC with a RAM consisting of bits per word.

2.2. CONCURRENT TESTING OF ROM

ROMs are commonly embedded into current VLSI chips, and constitute critical parts in complex circuits. When the operation of a ROM module is stalled, the whole circuit performance is degraded. Therefore, a scheme for built-in testing for ROM should meet two requirements: 1) very high fault coverage, and 2) no need to stop the normal operation. To meet the first requirement, practical BIST schemes for ROM exercise exhaustive testing this kind of testing is enough to cover all logically testable faults.

With input vector monitoring concurrent BIST techniques, testing is performed during normal operation of the module. Thus, R-CBIST also meets the second requirement, and therefore manifests itself as a promising solution in ROM testing. To evaluate the applicability of R-CBIST, we shall use as a metric the percentage of hardware overhead, i.e., the ratio of the BIST hardware overhead over the hardware overhead of the ROM. In practical off-line BIST schemes, 15% has been considered an upper limit for the percentage hardware overhead. To calculate the percentage hardware overhead of R-CBIST, we have also implemented ROM of typical sizes, and have calculated their hardware overhead in gate equivalents.

2.3 C-BIST CONCEPT

To illustrate the C-BIST concept, let us consider a BIST scheme, in which the combinational part of a circuit with *n* inputs is tested exhaustively by using the circuit organization shown in Fig.4. In this organization, we assume that the TG and the RV stay idle during the normal mode of operation of the circuit. We have chosen this representation for clarity of presentation. An actual organization may not employ multiplexers, or only parts of TG and RV may form idle test logic. We shall comment on these details in later sections. We modify the hardware associated with the idle test logic as shown in Fig.4.3**.** We have added an equality comparator to compare the normal inputs to the CUT to the outputs of the TG. The structures shown within dotted lines form the CBU. The two modes of testing proceed as follows.

2.3.1 Test Mode

This is off-line testing mode. During this mode, the multiplexer is set such that the tests generated by the TG are applied to the CUT and the responses are compressed using the RV. After the application of all tests, contents of the RV are used to determine the status of the CUT. In this configuration of the circuit, the equality comparator plays no role at all.

2.3.2 Normal Mode

This is the system operation mode and the testing of the CUT proceeds concurrently with the normal operation of the system. In order for the RV to obtain a result identical to that for the test mode, it must process only those responses that correspond to the test input vector set or test input sequence applied during the test mode. Thus the RV must be enabled only if a vector from a particular set or a vector at a particular point in a test sequence occurs. (The choice of a vector set versus a vector sequence depends on the type of RV employed.) The enabling of the RV is accomplished as follows. The TG produces, at its outputs, vectors from the test set or sequence for the CUT. During normal operation, the outputs of the TG are constantly compared, bit-by-bit, with the normal inputs to the CUT by the equality comparator. The particular output vector with which the normal inputs are compared is known as the active test vector.

A signal HIT is generated if and only if the normal inputs to the CUT are the same as an active test vector of the TG. When a HIT is generated, the RV is enabled to compress/record the normal outputs of the circuit. Also, the TG is advanced to the next state to produce the next vector only when a HIT occurs. It is possible to have multiple active test vectors; when the HIT signal is activated, the particular active test vector present is identified. When the TG has gone through the appropriate set of states, i.e., all test vectors in the set or sequence have been applied to the inputs of the CUT, then the contents of the RV are examined to determine the status of the CUT. From the above conceptual explanation, it should be evident that the testing of the circuit can proceed concurrently with the normal operation of the circuit.

2.4 Determination of test latency

An important evaluation measure for the performance of the CBU's is the time required for test completion. Let ***T*** be the set of test vectors to completely test the CUT. Clearly the fault-free status of the CUT cannot be verified until all the test vectors from ***T*** have been applied to the CUT. For the CBU discussed in the previous section**,** the status of the circuit cannot be determined until the TG has gone through all the required states. The question we need to address **is** the time required for the TG to go through all the states necessary to completely test the circuit.

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.4) and a logic module based on a static-RAM (SRAM)-like cell, as will be explained shortly. The design of the m_dec module for $w = 3$ is shown in Fig.5 and operates as follows. When test generator enable (tge) is enabled, all outputs of the decoder are equal to one. When comparator (cmp) is disabled (and tge is not enabled) all outputs are disabled.

When 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.3 The module labeled logic in Fig.4 is shown .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 $w = \log 2W$). 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 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).

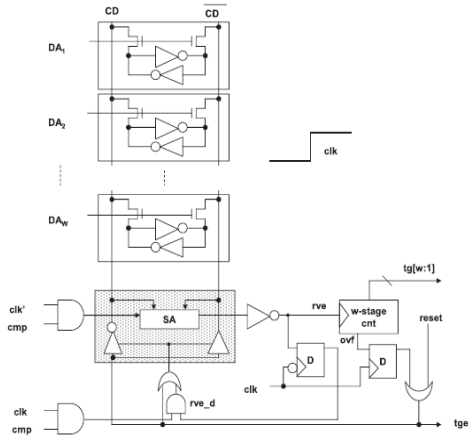

Figure 6. Design of logic module

2.4.1 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.

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.

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 $W = 2w$. 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 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.

TABLE 1. Calculation of 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) + Dec (W) + ABC(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

TABLE II. Calculation of Hardware Overhead of proposed scheme

Module	Hardware Overhead	gates
n-stage multiplexer	n × MUX_{21}	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	

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 Signature Analysis Technique (MHSAT) [5], Order Independent Signature Analysis Technique (OISAT) [6], RAM-based Concurrent 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 = 2k)*, R-CBIST, w- CBIST, and the SWIM scheme. The cells used are two- input XOR gate (XOR2*)*, *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 (MUX21*)*. 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

TABLE III. Comparison of the schemes for the concurrent testing of various ROM sizes

Scheme	TPG	RV	H/W #gates	% H/W over c6288	Offline test patterns	Latency (sec)*		Detected faults
						of fault detection	of Concurrent test	
MICSET		ACC	1904	78,8%	64	85	85	Modeled
		CMP	2384	98,7%	64	0		
BICST	LFSR	CMP	2116	87,6%	2^{32}	0		
	ROM	CMP	3268	135,3%	28	0		
DWC	LFSR	CMP	2768	114,6%	2^{32}	0	$>2^{34}$	All
	ROM	CMP	4296	177,8%	28	0	$>2^{34}$	

*A 200 MHz clock is assumed

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 ¼ 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. 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 3.3, 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 favorable results, as shown in Fig. 6. For the calculations, we have considered ROMs with 16-bit words and a 100-MHz clock. In Table 3.3, 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). 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 3.3, 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]. A. Manikandan, V.Jamuna, "Fault Tolerant Parallel Filters Based On Error Correction Codes", Journal of Advanced Research in Dynamical and Control Systems, ISSN:1943-023X , pp 1399-1404, 2017.

[2]. A. Manikandan, P.Nirmal Kumar, "Network-On-Chip by using Power Reduction Technique" International Journal of Control Theory and Applications, ISSN:0974-5572, Volume 10(12), pp 265-269, 2017.

[3]. A.Manikandan, P.Nithya, "Low-Power Content Addressable Memory Based on Sparse Clustered Networks " Journal of Chemical and Pharmaceutical Sciences, ISSN:0974-2115, pp 302-304, Feb 2017.

[4]. A.Manikandan, P.Nirmal Kumar , "OFDM Technique Decoding Accelerator" Journal of Chemical and Pharmaceutical Sciences, ISSN:0974-2115, pp305-307, Feb 2017.

[5]. K.Bommaraju, A. Manikandan, S.Ramalingam, "Aided System for Visually Impaired People in

Bus Transport using Intel Galileo Gen-2" International Journal of Vehicle Structures & Systems, Online ISSN: 0975-3540, Volume 9, Issue 2, pp 110 112, 2017.

[6]. P.Nirmal Kumar,A.Manikandan, "Zigbee Based Online Air Pollution Monitor" Journal of Chemical and Pharmaceutical Sciences, ISSN:0974-2115, pp 230-232, Feb 2017.

[7]. A.Manikandan, S.Ramalingam, V.Aathi bhagavan, "Potholes Alert System for Riders" in International Journal of Advances in Natural and Applied Sciences, ISSN:1995-0772, pp 440-444, July 2016.

[8]. Murugesan.S, R. Senthilkumar. "Design of Single Phase Seven Level PV Inverter Using FPGA." International Journal of Emerging Technology in Computer Science & Electronics, 2016, Vol.20, No.2, pp.207-2012.

[9]. S. Murugesan, C. Kalavalli," FPGA Based Multilevel Inverter With Reduce Number of Switches For Photovoltaic System", International Journal of Scientific Research in Science, Engineering and Technology(IJSRSET), Print ISSN : 2395-1990, Online ISSN:2394-4099, Volume 3 Issue 6, pp.628-634, September-October 2017.

[10]. S. Arunraj, S. Murugesan, T. Uma Nandhini, K. Keerthana, 'A Novel Zeta Converter with Pi Controller for Power Factor Correction in Induction Motor', International Journal of Scientific Research in Science and Technology(IJSRST),Print ISSN : 2395-6011, Online ISSN:2395-602X,Volume3,Issue8,pp.230-234, November-December2017. URL : http://ijsrst.com/IJSRST173845.php

[11]. M.G.Rani Gowshalya, S.Murugesan, "Improved Fault Ride Through Capability Using LSC And An ESS For PMSG Wind Turbine System", International Journal of Applied Engineering Research (IJAER), ISSN:0973-4562, Volume 10, Issue 9, pp. 7344-7349, 2015. Research India Publications.

[12]. K.Umamaheswari, S.Murugesan, "Reactive Power Pricing Methods Based On Consumption & Power Loss", International Journal of Applied Engineering Research (IJAER), ISSN:0973-4562, Volume 10 Issue 55, pp. 3309-3314, 2015. Research India Publications.

[13]. A.T. Sankara Subramanian, P. Sabarish and R Jai Ganesh. "An Improved Voltage follower Canonical Switching Cell Converter with PFC for VSI Fed BLDC Motor." Journal of science and technology. ISSN:2456-5660 http://jst.org.in/Vol-2Issue-10October2017.html.

[14]. M. D. Udayakumar, A. T. Sankara Subramanian, 'Harmonic Analysis on Single Phase Symmetrical Multilevel Inverter for various classes of Variable Frequency PWM strategies', International Journal of Scientific Research in Science and Technology(IJSRST), Print ISSN : 2395-6011, Online ISSN :2395-602X,Volume 3, Issue 7, pp.749-755, September-October 2017. URL : http://ijsrst.com/IJSRST1737149.php

[15]. R.Senthilkumar, M.Vigneshwaran, S.Murugesan, "Hierarchical Hybrid Protocols for Wireless Sensor Network: A Survey", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume 2, Issue 11, pp.1401-1406, November 2013.

[16]. A.T. Sankara Subramanian, R.Sooriya Priya Dharshini, R.siranjeevi and S.Joshua smith, "A Novel Reduced Switch Multilevel Inverter Using FPGA", Journal of science and technology. ISSN:2456-5660 http://jst.org.in/Vol-2Issue-10October2017.html

[17]. P.Sabarish, R.Sneha, G.Vijayalakshmi and D.Nikethan, "Performance Analysis of PV- Based Boost Converter using PI Controller with PSO Algorithm", Journal of science and technology. ISSN:2456-5660 http://jst.org.in/Vol-2Issue-10October2017.html

[18]. Udayakumar, M. D., and M. Bhavani. "Design of Robust Power System Stabilizer using Neuro-Fuzzy controller."

[19]. Nazar, A. Ali, R. Jayabharath, and M. D. Udayakumar. "An ANFIS Based Advanced MPPT Control of a Wind-Solar Hybrid Power Generation System." International Review on Modelling and Simulations (IREMOS) 7.4 (2014): 638-643.