Input Vector Monitoring Concurrent BIST Techniques:A Survey

K.Priyadharshini,R.Prashidha

ECE Department,ME VLSI Design, SVS College of Engineering,coimbatore

Abstract - **Built-In Self-Test (BIST) techniques constitute a striking and practical solution to the problem of testing VLSI circuits and systems. In traditional methods it would be possible to generate test vectors off-line and store them in an on-chip ROM. This has not, however, been an attractive scheme it does nothing to reduce the cost of test pattern generation and it requires a very large ROM. Hence input vector monitoring concurrent built-in self test schemes is proposed which perform testing during the normal operation of the circuit without imposing a need to set the circuit offline to perform the test. Input vector monitoring concurrent BIST schemes can evade problems appearing separately in on-line and in off-line BIST techniques. 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. Many techniques such as C-BIST, MHSAT, OISAT, R-CBIST, w-MCBIST SWIM and eSCIMO have been emerged which performs efficiently in-terms of hardware overhead and CTL trade-off.**

Keywords: C-BIST, MHSAT, OISAT, R-CBIST, w-MCBIST SWIM, eSCIMO

I. INTRODUCTION

Built-in self test (BIST) techniques consist of a class of schemes that provides the ability of performing at-speed testing with high fault coverage, while at the same time they relax the reliance on expensive external testing equipment. Hence, they represent an attractive solution to the problem of testing VLSI devices [1]. BIST techniques are classified into off-line and on-line. Offline architectures operate in either normal mode or test mode. The traditional technique for off-line testing make use of test pattern generation and fault simulation algorithm to find a set of test vectors to spot the modeled fault in the circuit. These tests can be applied to the circuit either by external tester or on the other hand they can be stored on-chip and applied during the test mode. The method of storing tests on-chip and applying the test vector during the test phase can be sighted as an off-line BIST technique. In off-line BIST, the normal operation of the CUT is stopped in order to perform the test. Thus, if the CUT is vital for the function of the circuit, the total circuit performance is ruined. To avoid this, input vector monitoring concurrent BIST techniques have been proposed which make use of input vectors arriving at the inputs of the CUT during normal operation. Input vector monitoring concurrent BIST architectures test the CUT in parallel with its normal operation by utilizing input vectors appearing to the inputs of the CUT [2]-[10]; if the incoming vector belongs to a set called active test set, the RV is enabled to detain the CUT response.

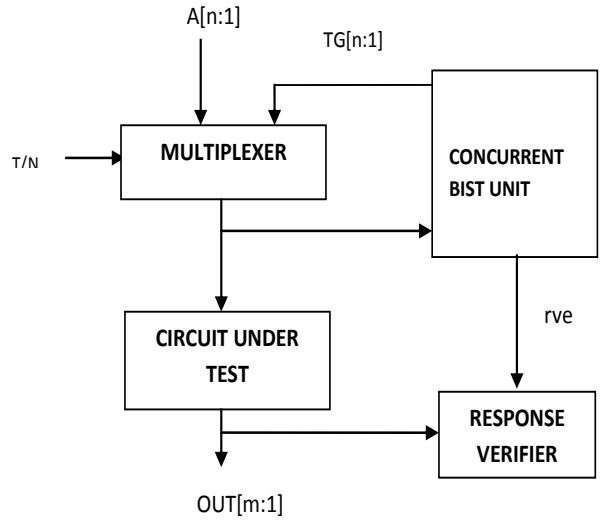

Fig.1. Input vector monitoring concurrent BIST

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 CUT (denoted by e [n: 1] in Fig. 1) is driven from the normal input vector (*A [n: 1]*). *A* is as well driven to a concurrent BIST unit (CBU), where it is compared with the active test set. If it is found that *A* matches one of the vectors in the active test set, we say that a hit has occurred. In this case, *A* is removed from the active test set and the signal response verifier enable (rve) is emanated to enable the m-stage RV to capture the CUT response to the input vector [1]. When all input vectors have executed hit, the contents of RV are observed. During test mode, the inputs to the CUT are driven from the CBU outputs denoted *TG [n: 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.

II. TECHNIQUES

A. C-BIST

In off-line BIST organization, TG and RV remain unused during normal mode and in the test mode the normal operation of the CUT is stopped and performance gets deteriorated. Hence the hardware related with this organization is modified and such organization is said to be Comparative Concurrent-Built-in-Self Test (C-BIST) [2]. In this modified organization an equality comparator is added (shown in Fig.2) to compare the normal inputs to the CUT to the output of TG. The Active test set Generator and Comparator of C-BIST is a single Linear Feedback Shift Register (LFSR) and a comparator and thus the active test set consists of only one active test vector, which is the present value of the LFSR. During normal mode, the input vector is compared with the distinctive active test vector. If the two vectors match, the LFSR shift to the next state changing the active test vector and the Response Verifier (implemented as a Multiple Input Shift Register, MISR) is enabled. In C-BIST, the value of the concurrent test latency is very high since in every clock cycle the active test set consists of only one active test vector.

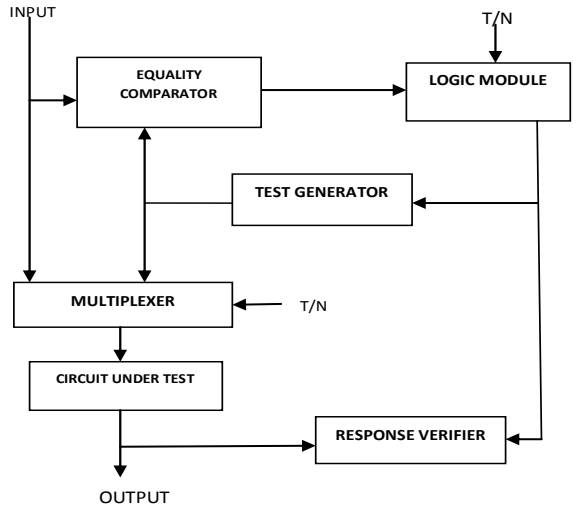

Fig.2.C-BIST Organization

C-BIST suffers from long test latency due to the following reasons:

- During normal circuit operation some inputs are given to the CUT very often, while some others not quite. This depends on the specific application of the circuit.
- Every instant only one specific vector is expected, i.e. the order in which the vectors are expected is fixed.

The first problem is solved [2] by utilizing timeout indicators, which, if too much time has passed for an active test vector, halt the normal system operation and force a test vector injection. With respect to the second problem, two techniques have been proposed [3], [4] that reduce the test latency by increasing the probability of HIT, namely Multiple Hardware Signature Analysis Technique (MHSAT) and Order-Independent Signature Analysis Technique (OISAT).

B. MHSAT AND OISAT

The Active test set Generator and Comparator of MHSAT [3], consists of L *(L>1)* LFSRs and L comparators and thus the active test set consists of L active test vectors. Also, there is L order-dependent Response Verifiers (implemented as MISRs) each one corresponding to an LFSR. During normal mode, the input vector is compared to the L active test vectors. A hit occurs if the input vector matches any one of the active test vectors. In this case, the LFSR that performed the hit continues to the next state changing the specific active test vector, and the corresponding MISR is enabled. The test is complete when all the LFSRs have completed their cycles.

OISAT [4] realizes an Active test set Generator and Comparator similar to the one used by MHSAT, but instead of L order-dependent Response Verifiers, it employs only one order-independent Response Verifier (the Accumulator-Based Compaction of the responses, ABC).The Response Verifier is enabled every time an LFSR performs a hit.

C. R-CBIST

RAM-based Concurrent BIST which is more efficient than the other input vector monitoring concurrent BIST techniques projected so far in terms of Hardware Overhead and Concurrent Test Latency trade-off. Read-only Memories are commonly embedded into current VLSI chips and form critical parts in complex circuits. When the operation of a ROM module is stopped, the whole circuit performance is deteriorated. As a result, a BIST scheme for ROMs should meet two requirements: (a) very high fault coverage (b) no needs to stop the normal operation. So as to meet the first requirement, practical BIST schemes for ROMs apply exhaustive testing. This kind of testing is enough to cover all logically testable faults. By means of concurrent BIST techniques, testing is performed throughout normal operation of the module. Thus R-CBIST also meets the second requirement and thus manifests itself as a promising solution in ROM testing.

R-CBIST [5] is based on a separate data bus RAM with 2^k *x (w+l)*-bit memory locations (shown in Fig.3.), where $k+w=n$. The pinout of the RAM consists of the address bus *(A [k: l])* the input data bus (*IN [w+l: l])* the output data bus *(OUT [w+l: l])* and two control signals: clock enable bar (*ceb*) and write enable bar (*web*). Every instant, the address bus point out the address of a memory word called enabled word. The address of the enabled word is the enabled address. The signal ceb is used to synchronize the read write access operation and web is used to differentiate between a read and a write access cycle. In R-CBIST, the Response Verifier is order-independent. The accumulator-based compaction (ABC) [6] of the responses is an order-independent Response Verification technique that requires only one D-type flip-flop and one full adder at each output of the CUT. Furthermore the aliasing characteristics of ABC

are similar to the characteristics of the best compactors based on Multiple Input Shift Registers and/or Cellular Automata (CA) [7].

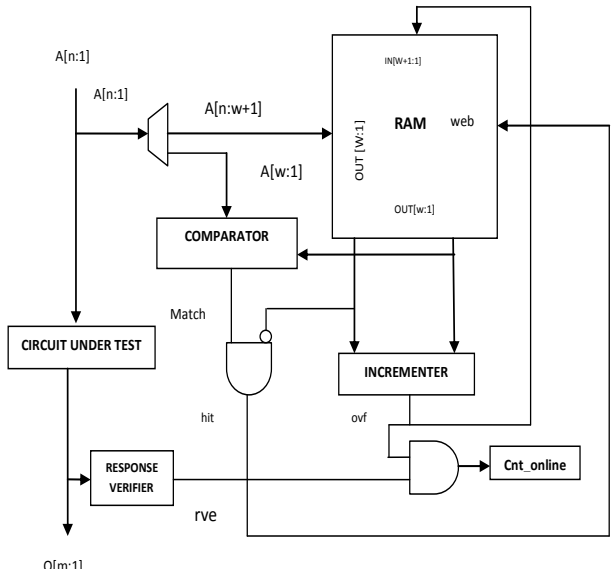

Fig.3.R-CBIST Technique

D. W-MCBIST

The Window-Monitoring Concurrent BIST scheme is shown to be better than other input vector monitoring concurrent BIST schemes for low values of the hardware overhead. The test set size is $Ts=2^n$. W-MCBIST is based on the partition of the test set into non-overlapping subsets (called windows) each one of size $Ws=2^w$, where $0 < w < n$; and N_w is the total number of windows; thus, $N_w = Ts/Ws = 2^n/2^w = 2^{n-w}$.

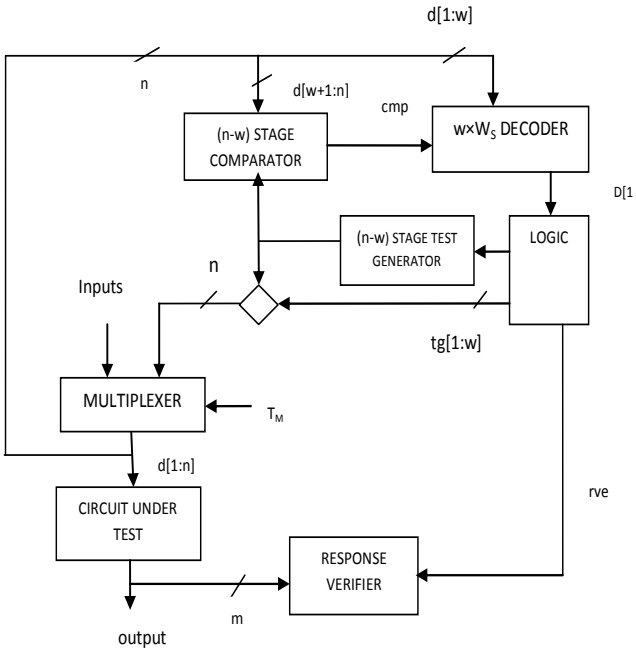

Fig.4. W-MCBIST technique

A block diagram of w-MCBIST is presented in Fig 4. During normal operation, w-MCBIST utilizes the arrival of any test vector belonging to a specific window, called active window [8]. The input vector is compared against a set of *Ws* vectors that comprise the active window as follows. The *(n-w)* bits of the n-bit input vector are compared with the outputs of the *(n-w)*-stage test generator and, if they match, cmp is enabled. The remaining w bits of the input vector are driven to the inputs of a $w \times Ws$ decoder whose enable input is driven by cmp. Therefore, if the input vector belongs to the active window, one of the *D[i]* signals is enabled. When all vectors of a window have performed a hit, the signal test generator enable (tge) triggers the test generator to the next state, in order to observe a new window. After the test generator has generated all its 2^{n-w} states, we inspect the signature captured in RV and decide whether the CUT is faulty.

E. SWIM

Square Windows Monitoring (SWIM) [9] concurrent BIST is based on monitoring input vectors using a square window; it is shown in Fig.5. The SWIM is better than formerly proposed input vector monitoring schemes, with regard to concurrent test latency and hardware overhead trade-off.

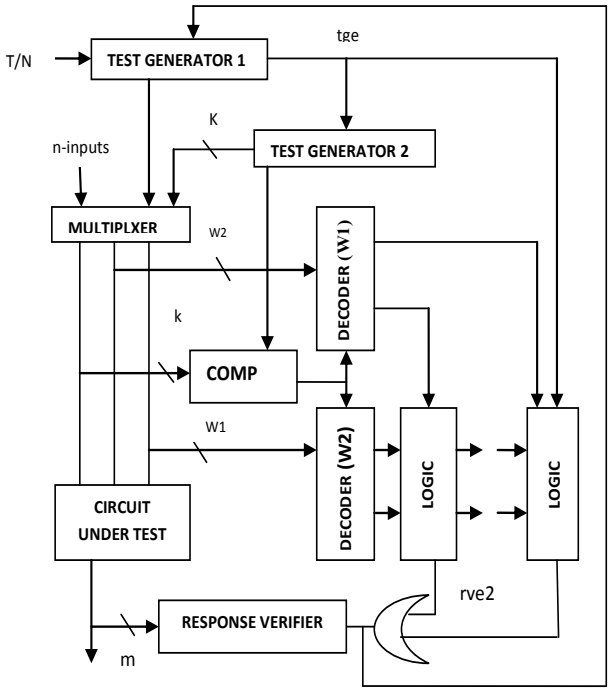

Fig.5.The SWIM architecture

The basic idea of SWIM is to monitor a "square" window of vectors, whose size is *W1xW2*, with $W1=2^{W1}$ and $W2 = 2^{W2}$. Every instant, the test vectors belonging to the window are monitored, and if any one vector performs a hit, the response verifier is enabled. The bits of the input vector are divided into three distinct sets comprising w1, w2 and k bits respectively, such that $w1+w2+k=n$. The *k* (high order) bits of the input vector specify whether the input vector

belongs to the window under consideration. The *w1+w2* remaining bits specify 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 assessment of the current window, we say that the vector has performed a hit and the response verifier is clocked in order to capture the CUT's response to the vector. When all vectors that belong to the current window have reached the CUT inputs, we continue to inspect the next window.

F. eSCIMO

A combinational CUT with *n* input lines are considered, as shown in Fig. 6; therefore the possible input vectors for this CUT are 2^n. This scheme is based on the plan of monitoring a window of vectors, whose size is *W*, with $W = 2^w$, where *w* is an integer number $w < n$. The bits of input vector is partitioned into *k* and w bits where $k+w=n$. Every instant, the test vectors belonging to the window are monitored, and if a vector carry out a hit, the RV is enabled.

Logic module used here is SRAM-cell like structure for storing the information of whether an input vector has emerged or not during normal operation [10]. There are four cases considered in the operation of logic module: 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. In addition, here a decoder is modified in-order to improve the latency. This scheme is more efficient than previously proposed input vector monitoring concurrent BIST techniques in terms of hardware overhead and CTL trade-off.

III. COMPARISON

C-BIST [2] was the first input vector monitoring concurrent BIST technique projected, and suffers from long CTL; as a result modifications have been suggested namely Multiple Hardware Signature Analysis Technique (MHSAT) [3], Order Independent Signature Analysis Technique (OISAT) [4], RAM-based Concurrent BIST (R-CBIST) [5], Window-Monitoring Concurrent BIST (w-MCBIST) [8], Square Windows Monitoring Concurrent BIST (SWIM) [9] and eSCIMO [10]. The comparisons will be performed with respect to the value of the CTL and the hardware overhead.

In **Table I**, we present the formulas that we used to calculate the hardware overhead of MHSAT, OISAT *($K = 2^k$)*, R-CBIST, w-MCBIST, SWIM and the eSCIMO 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*)*. Concurrent test latency for exhaustive testing is the time it takes for the TG to go through all possible 2^n states during normal operation. The hardware overhead is calculated using the gate equivalents as a metric. In **Fig. 7**, the CTL is presented (in time units, i.e., seconds) as a function of the hardware overhead (in gate equivalents) for MHSAT, OISAT, R-CBIST, w-MCBIST and SWIM. A CUT with $n = 16$ inputs and $m = 16$ outputs has been considered. Thus, we concluded that eSCIMO is more efficient than MHSAT, OISAT, w-MCBIST and SWIM with regard to the hardware overhead—CTL trade-off.

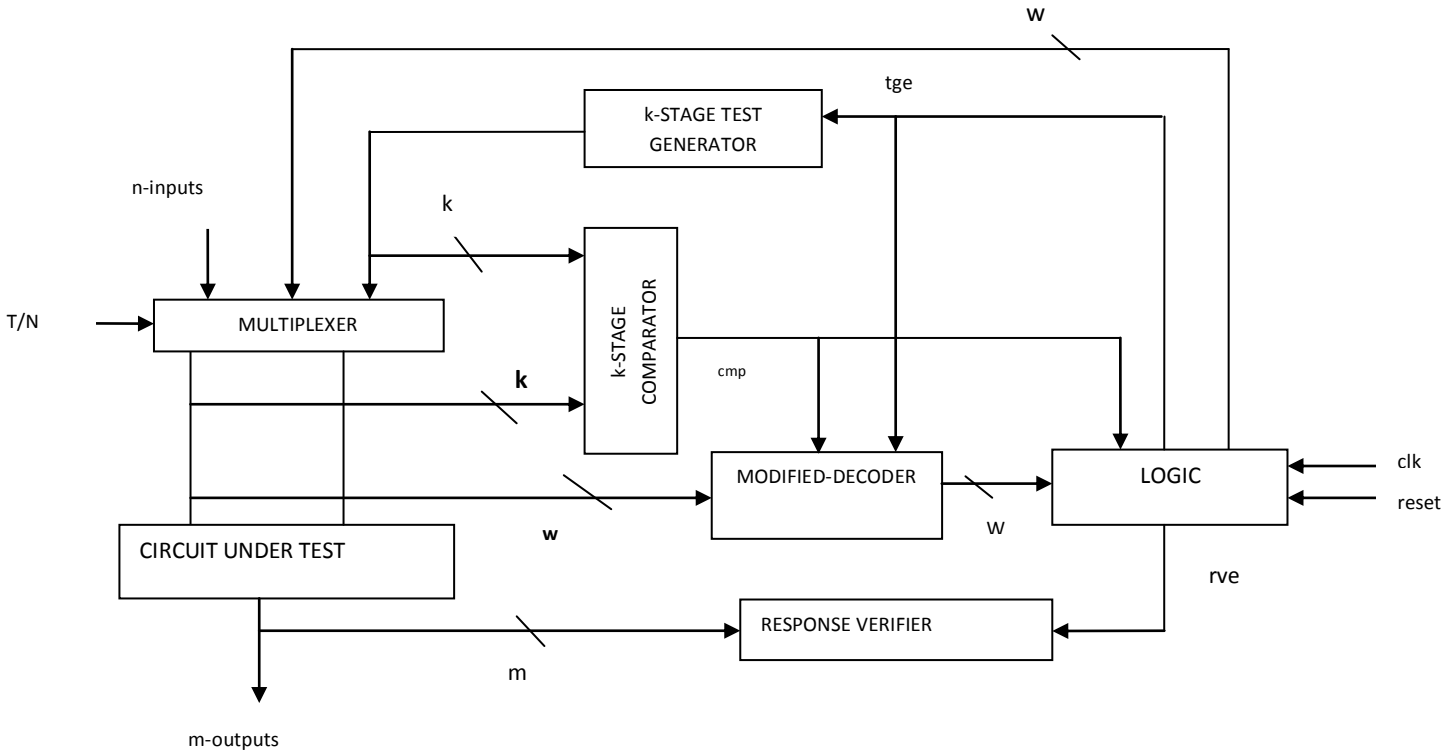

Fig .6.SRAM based concurrent input vector monitoring BIST

TABLE I
CALCULATION OF THE HARDWARE OVERHEAD

TECHNIQUE	MODULE	CALCULATION
C-BIST [2]	Mux(n)+comp(n)+NFSR(n)+MISR(m)	17n+12m
MHSAT(n,m,K) [3]	K× Mux(n)+comp(n)+NFSR(n)+K×MISR(m)	17Kn+12Km
OISAT(n,m,K) [4]	K× Mux(n)+comp(n)+NFSR(n)+ABC (m)	17Kn+18m
R-CBIST(n,m,R) [5]	Comp(w)+NFSR(n-w)+INC(w)+RAM	11k+25w+18m+RAM
w-MCBIST(n,m,w) [8]	Mux(n)+comp(n)+NFSR(n)+Logic (W)+dec(W)+ABC(m)	15n+9W+18m
SWIM(n,m,W1,W2) [9]	n×MUX_{21}+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
eSCIMO(n,m,k,W)[10]	n×MUX_{21}+m×(DFF+FA)+k×(XOR2)+k-stage AND+2×W+1,5×W+SA+2×Buf+2×DFF+2+8×k+8×W	15×n+18×m+k+3,5×W+2 3

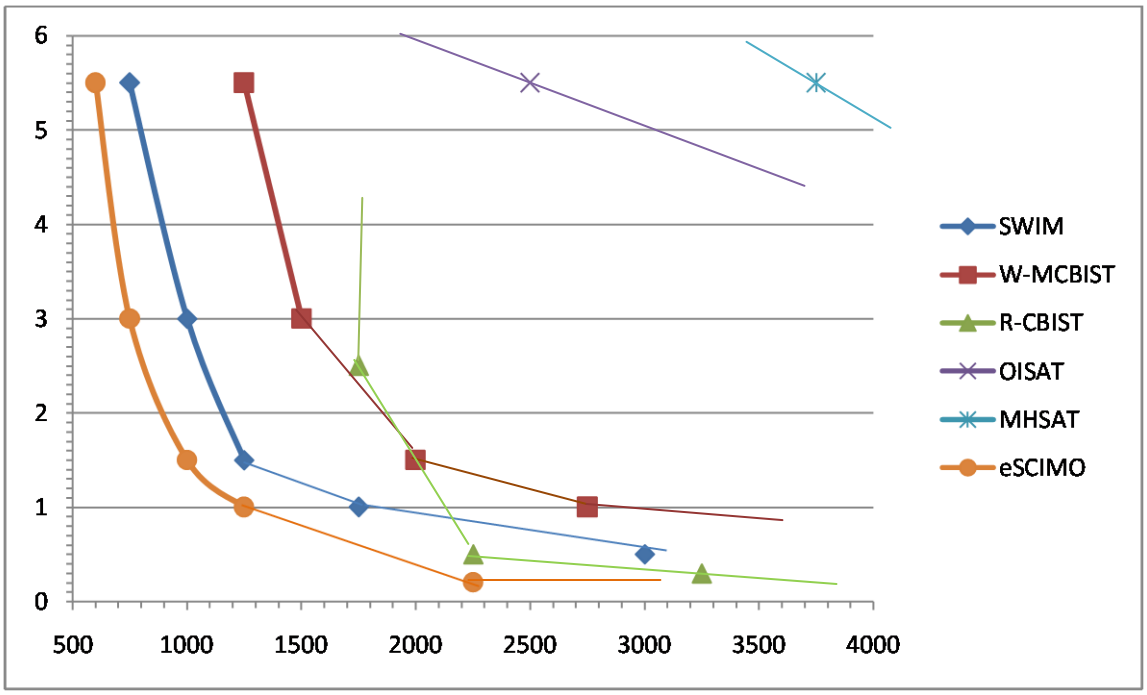

Fig.7.Input vector monitoring techniques: comparison (n = 16, m = 16, and 100-MHz clock)

(Courtesy: IOANNIS VOYIATZIS)

IV. CONCLUSION

A built-In Self-Test (BIST) technique represents a striking and practical solution to the problem of testing VLSI circuits and systems. During the circuit normal operation,Input vector monitoring concurrent BIST scheme do testing without imposing a need to set the circuit offline to perform the test, consequently they can avoid problems appearing in offline BIST techniques. The evaluation criteria for this class of schemes are the hardware overhead and the concurrent test latency, i.e., the time required for the test to complete; while the circuit operates normally. Many input vector monitoring concurrent BIST techniques has been presented which are efficient in terms of hardware overhead and CTL trade-off. An input vector monitoring concurrent BIST named eSCIMO is recently emerged technique which is 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 [10]. This scheme is more efficient than previously proposed input vector monitoring concurrent BIST techniques in terms of hardware overhead and CTL trade-off.

REFERENCES

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

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

[3] 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.

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

[5] 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.

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

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

[8] 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

[9] Voyiatzis, T. Haniotakis, C. Efstathiou, and H. Antonopoulou, "A concurrent BIST architecture based on monitoring square windows," in *Proc. 5th Int. Conf. DTIS*, Mar. 2010, pp. 1–6.

[10] Ioannis Voyiatzis and Costas Efstathiou, "Input Vector Monitoring Concurrent BIST Architecture Using SRAM Cells" IEEE Trans. On very large scale integration (vlsi) systems, vol. 22, no. 7, JULY 2014.

