IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 4, April 2015.
www.ijiset.com
ISSN 2348 – 7968

Effective BIST architecture to reduce hardware overhead in digital circuits

B.DineshKumar M.P.Nirmala

***Abstract*—In VLSI testing technology the role of DFT is inevitable to reduce hardware overhead and latency. Input vector monitoring Built In Self Test (BIST) architecture execute the testing of normal operating circuit without enforce the circuit offline. In this paper, to achieve goals that area overhead (number of gates) and Concurrent Test Latency (CTL) i.e., amount of time taken to complete testing the circuit. The relative locations from examined window can be stored in SRAM which can consume less power. The proposed effective BIST architecture is shown as better than other BIST schemes which result in terms of low area overhead and CTL.**

***Index Terms*— Built-in self-test, design for testability, Testing, Very large scale integration (VLSI).**

I. Built In Self Test (BIST)

VLSI chip testing is done in several different places by several different types of people. When a new chip is designed and fabricated for the first time, testing should verify correctness of design and the test procedure. This frequently needs the involvement of the design engineer and the testing may even take place in the design laboratory rather than in a factory. Based on the result, both the design and the test procedure may be changed. This is called verification testing. Successful verification testing usually results in some good chips. These are the earliest chips and are normally used by the designers of systems that will use this design. A successful verification also signals the beginning of production. Production means large scale manufacturing. Fabricated chips are tested in the factory. This is called manufacturing testing. Finally, when the manufactured chips are received by a customer, they may be again tested to ensure quality. This testing, known as incoming inspection (or acceptance testing), is conducted either by the user or for the user by some independent testing house. If a fault is found, a part of the circuit (having the fault) is replaced with a corresponding redundant circuit part (by re-adjusting connections). Testing a circuit every time before they startup, is called BIST. Once BIST finds a fault, the readjustment in connections to replace the faulty part with a fault free one is a design problem. This paper presents central concepts of testing of VLSI circuits by BIST.

B.Dinesh Kumar is with the PG Scholar, Electronics and Communication Department, Adhiparasakthi Engineering College,Melmaruvathur, India
E-mail: dinesh.dazzler@gmail.com

II. Concurrent BIST architecture

Built-in self-test (BIST) techniques composed of a class of schemes that provide the potentiality of performing at speed testing with high fault coverage, whereas same time they unstaring the reliance on expensive external testing equipment. Hence, they include an charismatic solution to the problem of testing VLSI devices [1]. BIST schemes are divided 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 (CUT) and the responses are captured into a response verifier (RV). Therefore, to perform the test, the normal operation of the CUT is conked and, therefore 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 CUT simultaneously with its normal operation by exploiting input vectors appearing to the inputs of the circuit under test if the incoming vector belongs to a set called active test set, the RV is enabled to capture the CUT response. The block diagram of an input vector monitoring concurrent BIST architecture is shown in Fig. 1.

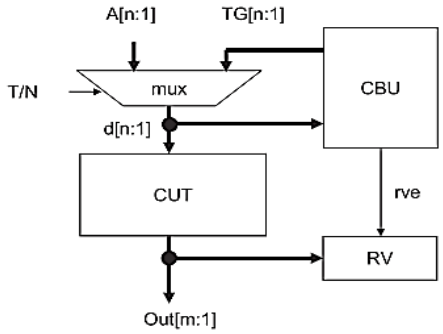

Fig. 1 Input vector monitoring concurrent BIST architecture

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. During normal mode, the vector that drives the inputs of the CUT (denoted by d[n:1] in Fig. 1) is impelled from the normal input vector (A[n:1]). A is also impelled 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

ISSN 2348 – 7968

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].When all input vectors have performed hit, the contents of RV are examined. During test mode, the inputs to the CUT are impelled from the CBU outputs denoted TG [n:1].

III. 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 following idea of monitoring a window of vectors, which has size W, with W = 2w, where w is an integer number w < n. Each time, 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 broke into two discrete sets comprising w and k bits, respectively, so that w + k = n. The k (high order) bits of the input vector show whether the input vector from the window under consideration. If the incoming vector from the current window and has not been received when 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 test the next window. The module implementing the idea is shown in Fig 2.

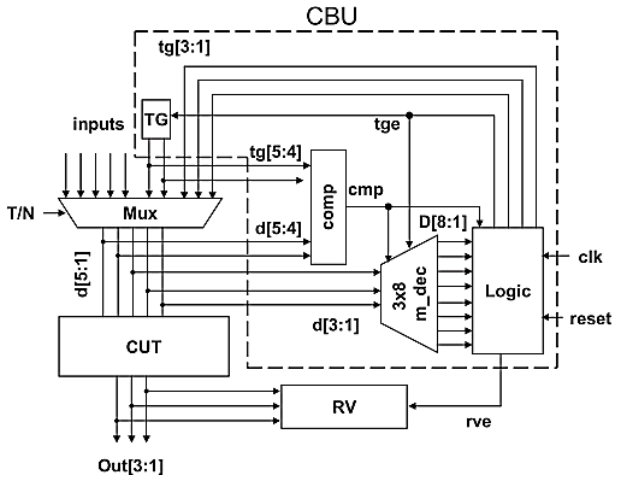

Fig. 2 C-BIST architecture for $n = 5$, $w = 3$, and $k = 2$

It works in one out of two modes, normal, and test, contingent on the value of the signal T/N. When T/N = 0 (normal mode) the inputs to the CUT are impelled by the normal input vector. The inputs of the CUT are also impelled to the CBU as follows: the k (high order) bits are impelled to the inputs of a k-stage comparator; the other inputs of the comparator are impelled by the outputs of a k-stage test generator TG. The proposed scheme uses a modified and a logic module following on a SRAM like cell, for power consumption purpose. The design of the m_dec module for w = 3 is shown in Fig 2 and operates as follows. When test generator enable (tge) is enabled, all outputs of the decoder are equal to 1. 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 2. The module labeled logic in Fig 2 is shown in Fig 3. 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 = log2W). 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.

A Reset of the Module

At the initial stage of the operation, the module is reset through the external reset signal. When reset is given, the tge signal is enabled and all the outputs of the decoder are enabled. Hence, DA1 , DA2, . . , DAW are one; not only that the CD signal is enabled; therefore, a 1 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

Vector Belongs in the Active Window and Reaches the CUT Inputs for the 1st Time. During normal mode, the inputs to the CUT are impelled from the normal inputs. The n inputs are also impelled to the CBU as follows the w low-order inputs are impelled to the inputs of the decoder the k high-order inputs are impelled 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 due to 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.

C Vector that belongs in the current window reaches the cut inputs but not for the first time

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

D tge signal operation

All Cells of the Window are Filled and to continue to test the Next Window. When all the cells are full (value equal to one),

ISSN 2348 – 7968

then the value of the w-stage counter is all one. Therefore the activation of the rve signal causes the counter to overflow, thus 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 2 are enabled) are set to zero. When switching from normal to test mode, the w-stage counter is reset. At 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. elsewhere, the cell remains full and the RV is not enabled.

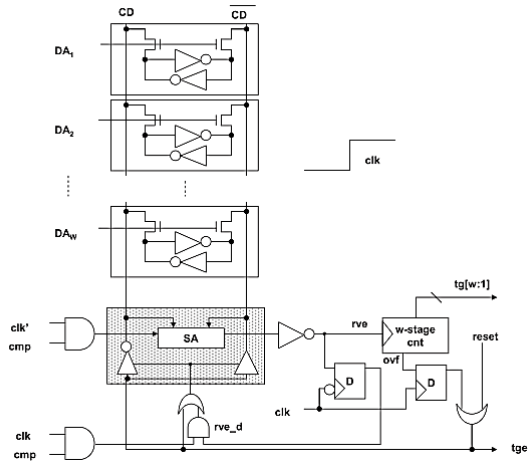

Fig. 3 Design of the logic module

A. 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 $W = 2w$.

IV. SIMULATION RESULTS

BIST is mainly used to reduce the complexity, and thereby decrease the cost and reduce reliance upon external (pattern-programmed) test equipment. BIST reduces cost in two ways, by reduces test-cycle duration reduces the complexity of the test/probe setup, by reducing the number gates used in circuit, I/O signals that must be driven/examined under tester control. From here the functionality of Concurrent BIST that it can test the digital circuits with low area overhead and less power consumption.

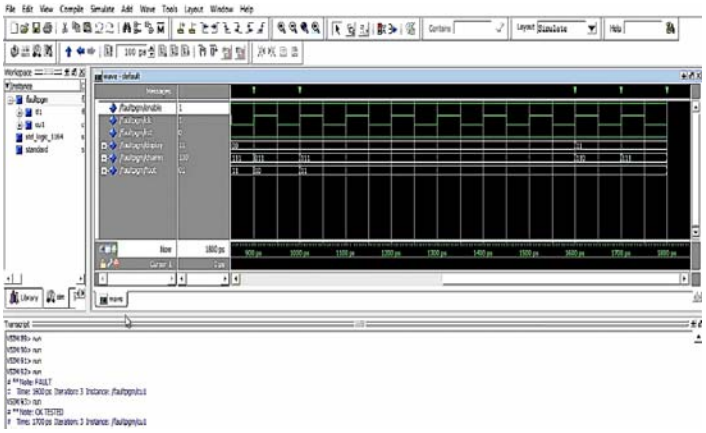

Fig. 4 Simulation result for testing Full adder as CUT

Here Considered Full Adder as CUT and The simulation result shows that all nodes of CUT are tested that the fault at 1600ps was found. Similarly the testing of all nodes can be done simultaneously.

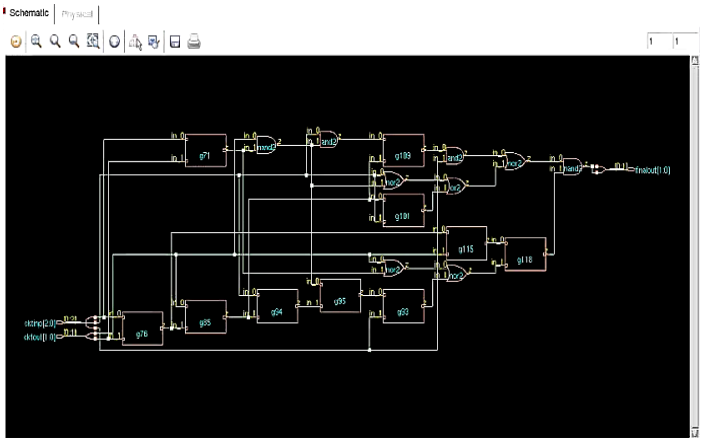

Fig. 5 RTL schematic for CBIST architecture

The proposed C-BIST architecture functioning of testing process has done with Model sim 6.4 and the RTL schematic for C-BIST has been presented by Cadence Tool. The Hardware area overhead and Technology mapping of respective BIST has implemented and the device area utilization was calculated and tabulated below.

V. COMPARISON OF C-BIST WITH OTHER BIST ARCHITECTURES

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. Multiple Hardware Signature Analysis Technique (MHSAT) [5], Order Independent Signature

ISSN 2348 – 7968

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. 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.

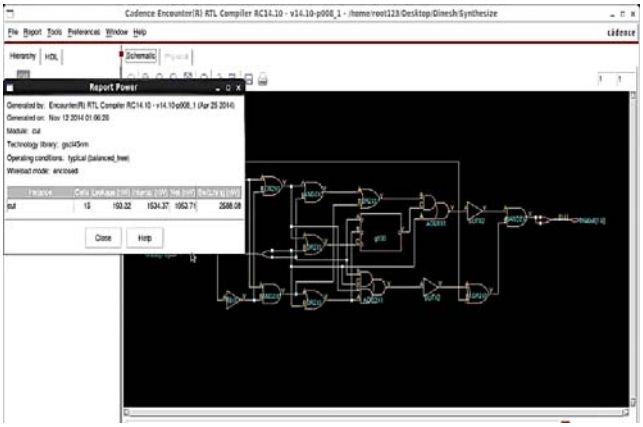

Fig. 6 Report for Power Consumption of CUT

For example, if a CTL of 3s 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 < 0.8s, the proposed scheme achieves the same CTL with R-CBIST with significantly less hardware overhead.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	11	1,408	1%
Number of 4 input LUTs	5	1,408	1%
Number of occupied Slices	7	704	1%
Number of Slices containing only related logic	7	7	100%
Number of Slices containing unrelated logic	0	7	0%
Total Number of 4 input LUTs	5	1,408	1%
Number of bonded IOBs	5	108	4%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	3.00		

Fig. 7 Report for device area utilization

VI. CONCLUSION

BIST schemes composed of 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 clear 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.

ACKNOWLEDGMENT

Mrs.M.P.Nirmala M.E she is working now as Assistant Professor in Adhiparasakthi Engineering College. Melmaruvathur.

B.Dinesh Kumar pursuing M.E. (VLSI Design) in Adhiparasakthi Engineering College, Melmaruvathur.

REFERENCES

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

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

[3] 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 testing 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 comparative built-in testing of digital circuits," Dept. Electr. Comput.Eng., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. ECE-8711, 1986.

[7] 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.

[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 accumulators 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-eliminationbased 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.