An Efficient Concurrent BIST for Rom Module

Hamsa C[1] Murali Narasimham S[2]
[1]P.G Scholar [2]Associate Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]Bangalore Institute of Technology, Bangalore, India

Abstract— Input vector monitoring concurrent built-in self-test schemes perform testing during the normal operation of the circuit without imposing a need to set the circuit offline to perform the test. These schemes are evaluated based on the hardware overhead and the concurrent test latency 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 con current BIST scheme for ROM module, which is based on the idea of monitoring a set (called window) of vectors reaching the circuit inputs during normal operation. The proposed scheme is shown to perform significantly better than previously proposed schemes with respect to area and power.

Key words: Built-in self-test (BIST), testing, Design for testability

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 (CUT) 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 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.

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. During normal mode, the vector that drives the inputs of the CUT (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), 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 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 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. In this brief, a novel input vector monitoring concurrent BIST scheme is proposed, which compares favorably to previously proposed schemes [2]–[7] with respect to the hardware overhead/CTL tradeoff.

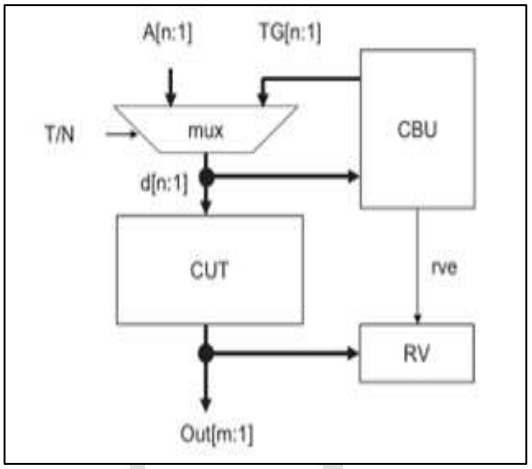

Fig. 1: Input Vector Monitoring Concurrent BIST.

II. EXISTING METHOD

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. 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.

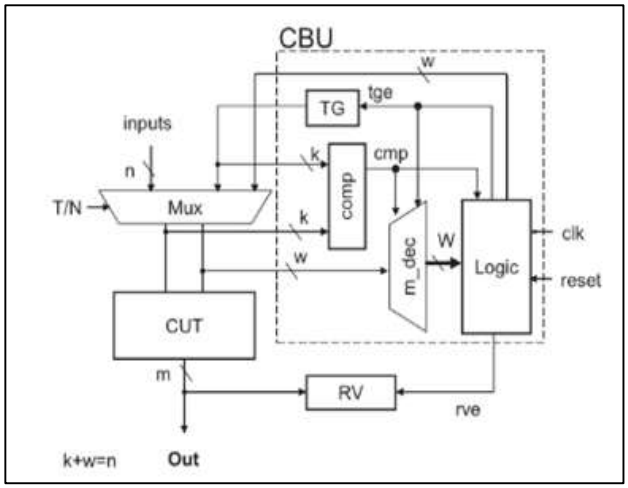

Fig. 2: Existing Architecture.

The proposed scheme improves previously proposed ones by exploiting the decoding structure of the ROM. In the sequel we shall present the design of the decoder used in the ROM and its exploitation for the proposed scheme. In Fig 3, a 3-to-8 decoder implemented using two level decoding is presented.

The input d[3] drives a 1-to-2 decoder; its outputs are denoted as D[1.2] and D[1.1]; the low-order inputs drive a 2to-4 decoder with outputs denoted as D[2.4] to D[2.1]. Consequently, the outputs of the 1-to-2 and 2-to-4 decoders are combined using eight AND gates and the outputs of the 3to-8 decoder D[1..8] are formed.

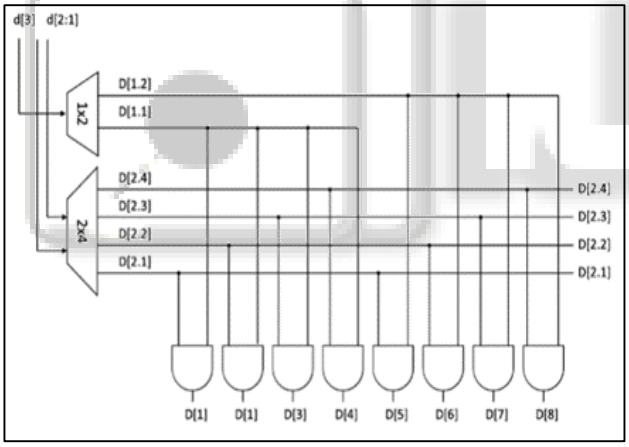

Fig. 3: 3-to-8 decoder with two-level decoding

III. PROPOSED SCHEME

The proposed block diagram is as shown below which provides better area and power when compared to the existing method. The scheme uses ROM as the circuit under test. A 5x32 decoder and a cell block are included in the CUT. The proposed block diagram is shown in Fig 4.

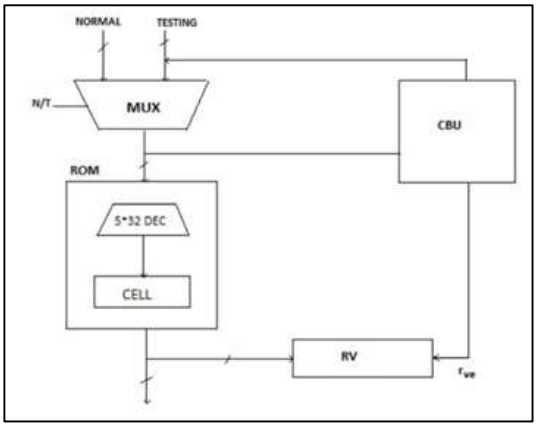

Fig. 4: Proposed Architecture.

The block diagram of the concurrent BIST unit (CBU) is as shown in Fig 5. The block includes a comparator, logic block, a 3x8 decoder, TG block.

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.

B. Hit Of Vector (I.E., Vector Belongs In The Active Window And Reaches The CUT Inputs For The First Time):

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.

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.

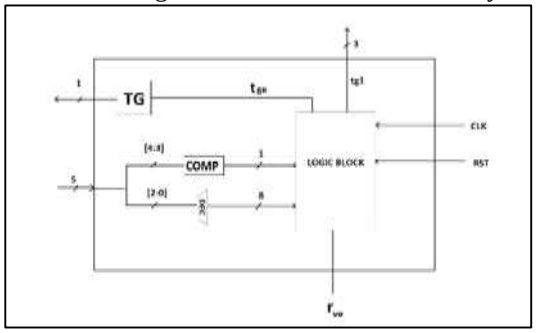

Fig. 5: Block diagram of CBU.

A 5 bit input is applied to the CUT. A 5x32 decoder used in the CUT block is as shown in the Fig 6. The output of the decoder is given to the cell. The cell stores the value in the available memory locations.

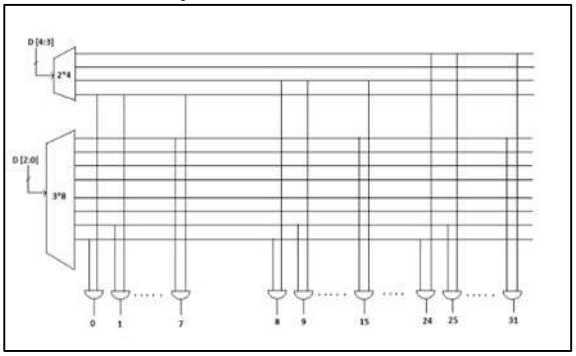

Fig. 6: Decoder design used in CUT Block.

In modification work as shown in Fig 7, we change the two input AND gates of 3x8 decoder and 5x32 decoder to mux. The replacement with mux can result in the reduction of the total transistor count. So we can able to get the efficient (min.10% than existing method) result than the existing method.

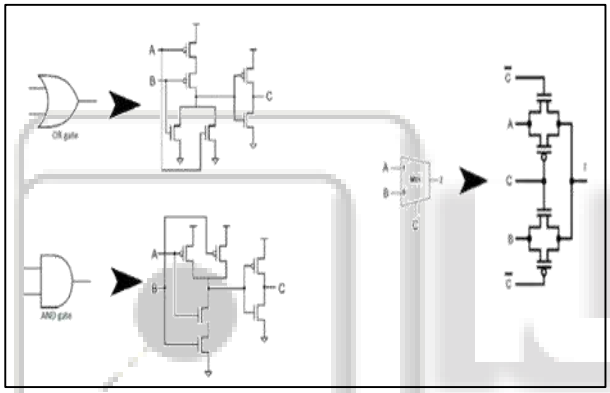

Fig. 7: Proposed Modification

IV. Simulation Results

BIST is mainly used to reduce the complexity, and thereby decrease the cost and reduce reliance upon external 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. The simulation waveform is shown in Fig. 8.

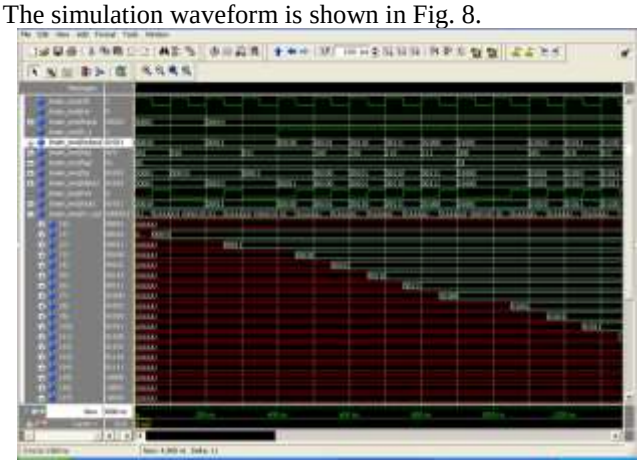

Fig. 8: Simulation Waveform

The power consumption of the existing and the proposed method was obtained using Xilinx Modelsim. The existing method consumed a power of 44mW. Due to modification done in the proposed method the power consumed is 39mW. Hence a reduction in power when compared to the existing method was obtained.

V. Conclusion

Concurrent on line Input vector monitoring concurrent BIST schemes exploit vectors appearing to the inputs of the CUT during normal operation to perform concurrent on line testing. In this work we have presented an input vector monitoring concurrent BIST scheme that, utilizing the concept of basic test sets, presents lower hardware overhead than previously proposed schemes. The proposed scheme is more efficient than other BIST schemes in terms of hardware overhead.

References

[1] E. J. McCluskey, "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, "Concurrent comparative built-in testing of digital circuits," Dept. Electr. Comput. Eng., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. ECE-8711, 1986.
[3] 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
[4] 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.
[5] I. 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] I. 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.

