Accumulator Based 3-Weight Test Pattern Generation Scheme

Geetanjali M. Dhurmekar, Prof. Vaishali.S.Dhongde

Abstract—Pseudorandom built-in self test (BIST) generators have been globally used to test integrated circuit and systems. In a BIST design, the generation, application of the test vectors and analysis of the resulting response are part of the system or circuit under test. Weighted pseudorandom BIST schemes have been utilized in order to minimize the number of vectors to achieve complete fault coverage in BIST applications. In accumulator based 3-weight test pattern generation scheme, weighted sets comprising 3 weights namely 0, 1 and 0.5 have been successfully utilized, since they result in both low consumed power and low testing time. The main advantages of this scheme over existing schemes are: 1) Only three easily generated weights-0, 1, 0.5 are used. 2) As accumulators are commonly used in current VLSI chips, this scheme can efficiently drive down the hardware of BIST pattern generation. 3) This scheme does not require any redesign of adder (i.e. it can be implemented using any adder design). 4) This scheme does not affect the operating speed of the adder. The software requirements are Modelsim and Xilinx tool and hardware require for this system is FPGA SPARTAN 3.

Index Terms—*Built-in self test (BIST), test per clock, VLSI testing, weighted test pattern generation.*

1 INTRODUCTION

THIS pseudorandom built-in self test (BIST) generators have been globally used to test integrated circuits and systems. The team of pseudorandom generators includes, among others, linear feedback shift registers (LFSRs) [1], cellular automata [2], and accumulators driven by a constant value [3]. In circuits with hard-to-detect faults, a large number of random test patterns have to be generated before high fault coverage is achieved. Therefore, weighted pseudorandom techniques have been proposed where inputs are biased by changing the probability of a "0" or a "1" on a given input from 0.5 (for pure pseudorandom tests) to some other value [10], [14].

Weighted random test pattern generation methods relying on a single weight assignment usually fail to achieve complete fault coverage using a limited number of test patterns since, although the weights are computed to be suitable for most faults, some faults may require long test sequences to be detected with these weight assignments if they do not match their activation and propagation requirements.

Multiple weight assignments have been suggested for the case that different faults require different biases of the input combinations applied to the circuit, to ensure that a relatively small number of test patterns can detect all faults [4]. Approaches to derive weight assignments for given deterministic tests are attractive since they have the potential to allow complete fault coverage with a significantly smaller number of test patterns [10].

For the minimization of the hardware implementation cost, other schemes based on multiple weight assignments were introduced which utilizes the weights 0, 1, and 0.5. This approach goes down to keep some outputs of the generator steady (to either 0 or 1) and letting the remaining outputs change their values pseudo randomly (weight 0.5). This approach, apart from reducing the hardware overhead has advantageous effect on the consumed power, since some of the circuit under test (CUT) inputs (those having weight 0 or 1) remain steady during the specific test session [19]. Pomeranz and Reddy [5] proposed a 3-weight pattern generation scheme relying on weights 0, 1, and 0.5. The choice of weights 0, 1, and 0.5 was done in order to reduce the hardware implementation cost. Wang [8], [13] proposed a 3-weight random pattern generator based on scan chains using weights 0, 1, and 0.5, in a way similar to [5]. Recently, Zhang et al. [9] renovated the interest in the 3-weight pattern generation schemes, proposing an efficient compaction scheme for the 3-weight patterns 0, 1, and 0.5. From the above we can conclude that 3-weight pattern generation based on weights 0, 1, and 0.5 has practical interest since it combines low implementation cost with low test time.

Current VLSI circuits, e.g., data path architectures, or digital signal processing chips commonly contain arithmetic modules [accumulators or arithmetic logic units (ALUs)]. This arises the idea of arithmetic BIST (ABIST) [6]. The basic idea of ABIST is to utilize accumulators for built-in testing (compression of the CUT responses, or generation of test patterns) and this results in low hardware overhead and low impact on the circuit normal operating speed [15]–[18]. In [15], Manich et al. presented an accumulator based test pattern generation scheme that compares favorably to previously proposed schemes. In [7], it was proved that the test vectors generated by an accumulator whose inputs are driven by a constant test pattern can have acceptable pseudorandom characteristics, if the input pattern is properly selected. However, modules containing hard-to-detect faults still require extra test hardware either by inserting test points into the mission logic or by storing additional deterministic test patterns [16], [17]. In order to overcome this problem, an accumulator-based weighted pattern generation scheme was proposed in [11]. The scheme generates test patterns having one of three weights, namely 0, 1, and 0.5 therefore it can be used to drastically minimize the test application time in accumulator-based test pattern generation. However, the scheme proposed in [11] possesses three major drawbacks: 1) it can be utilized only in the case that the adder of the accumulator is a ripple carry adder; 2) it requires redesigning the accumulator; this modification, apart from

being costly, requires redesign of the core of the datapath, this generally discourages in current BIST schemes; and 3) it increases delay, since it affects the normal operating speed of the adder.

In this paper, a novel scheme for accumulator-based 3-weight generation is presented. The proposed scheme copes with the inherent drawbacks of the scheme proposed in [11]. More precisely: 1) it does not impose any requirements about the design of the adder (i.e., it can be implemented using any adder design); 2) it does not require any modification or redesign of the adder circuit; and hence, 3) it does not affect the operating speed of the adder. Furthermore, the proposed scheme compares favorably to the scheme proposed in [11] and [15] in terms of the required hardware overhead.

This paper is organized as follows. In Section 2, the idea underlying the accumulator-based 3-weight test pattern generation is presented. In Section 3, the design methodology to generate the 3-weight test patterns using an accumulator is presented. In Section 4, the simulation results of the proposed scheme is compared presented. Finally, Section V, concludes this paper.

2 Accumulator Based 3-Weight Test Pattern Generation Scheme

Here is the idea of an accumulator-based 3-weight test pattern generation by means of an example. Let us consider the test set for the C17 ISCAS benchmark [12], [20] given in Table 1. Starting from this deterministic test set, in order to apply the 3-weight test pattern generation scheme, one of the schemes proposed in [5], [8], and [9] can be used. According to these schemes, a typical weight assignment procedure would involve separating the test set into two subsets, S1 and S2 as follows: S1 = { T1, T4 } and S2 = { T2, T3 }.The weight assignments for these subsets is W(S1) = {−, −, 1, −, 1} and W(S2) = {−, −, 0, 1, 0}, where a "−" denotes a weight assignment of 0.5, a "1" indicates that the input is constantly driven by the logic "1" value, and "0" indicates that the input is driven by the logic "0" value. In the first assignment, inputs A[2] and A[0] are constantly driven by "1", while inputs A[4], A[3], A[1] are pseudo randomly generated (i.e., have weights 0.5). Similarly, in the second weight assignment (subset S2), inputs A[2] and A[0] are constantly driven by "0", input A[1] is driven by "1" and inputs A[4] and A[3] are pseudo randomly generated.

These reasoning calls for a configuration of the accumulator, where the following conditions are met: 1) an accumulator output can be constantly driven by "1" or "0" and 2) an accumulator cell with its output constantly driven to "1" or "0" allows the carry input of the stage to transfer to its carry output unchanged. This latter condition is required in order to effectively generate pseudorandom test patterns in the accumulator outputs whose weight assignment is "−".

- *Geetanjali Motilal Dhurmekar is currently pursuing masters degree program in VLSI and Embedded system engineering in University of Pune, India, E-mail: dgeetanjali88@gmail.com*
- *Prof. Vaishali S. Dhongde is currently working as Head of Department in Vishwabharti college of engineering Ahmednagar,India. E-mail: nvvvvd@gmail.com*

TABLE 1
TEST SET FOR C17 BENCHMARK

Test Vector	Inputs A[4:0]
T1	00101
T2	01010
T3	10010
T4	11111

3 Design Methodology

The implementation of the weighted-pattern generation scheme is based on the full adder truth table, presented in Table2. From Table 2 we can see that in lines #2, #3, #6, and #7 of the truth table, Cout = Cin.

TABLE 2
TRUTH TABLE OF FULL ADDER

#	Cin	A[i]	B[i]	S[i]	C[i]	Comment
1	0	0	0	0	0	
2	0	0	1	1	0	Cout=Cin
3	0	1	0	1	0	Cout=Cin
4	0	1	1	0	1	
5	1	0	0	1	0	
6	1	0	1	0	1	Cout=Cin
7	1	1	0	0	1	Cout=Cin
8	1	1	1	1	1	

Therefore, in order to transfer the carry input to the carry output, it is enough to set A[i] = NOT (B[i]). The proposed scheme is based on this observation. The implementation of the proposed weighted test pattern generation scheme is based on the accumulator cell presented in Fig. 1, which consists of a Full Adder (FA) cell and a D-type flip-flop with asynchronous set and reset inputs whose output is also driven to one of the full adder inputs.

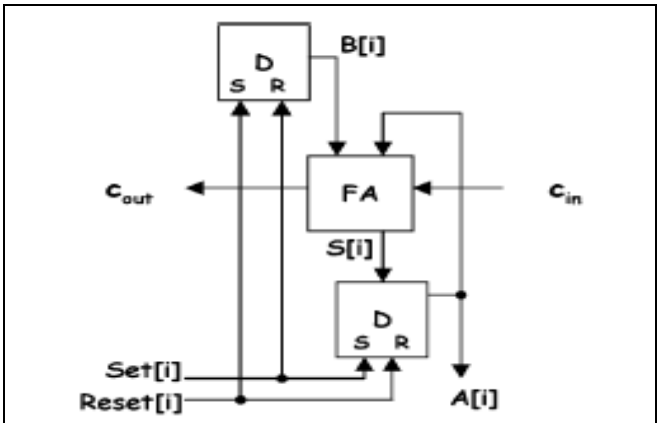

Fig. 1. Accumulator cell

In Fig. 1, we assume, without loss of generality, that the set and reset are active high signals. In this figure the respective cell of the driving register B[i] is also shown. For this accumulator cell, one out of three configurations can be used, as shown in Fig. 2.

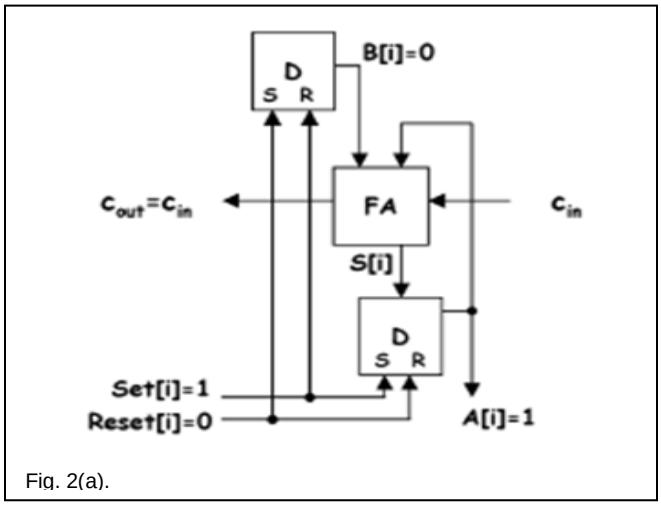

Fig. 2(a).

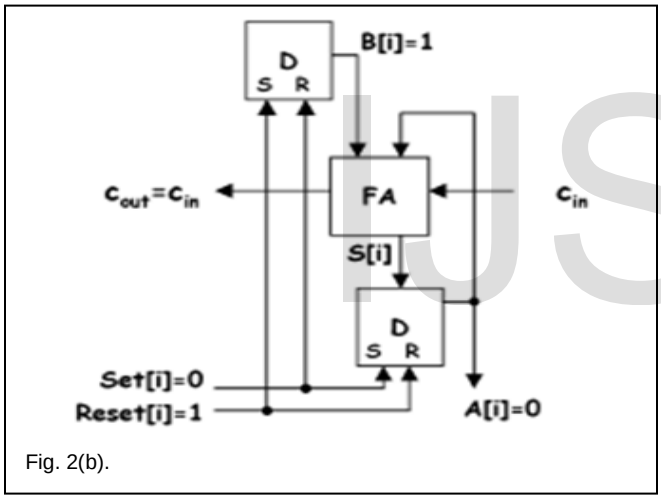

Fig. 2(b).

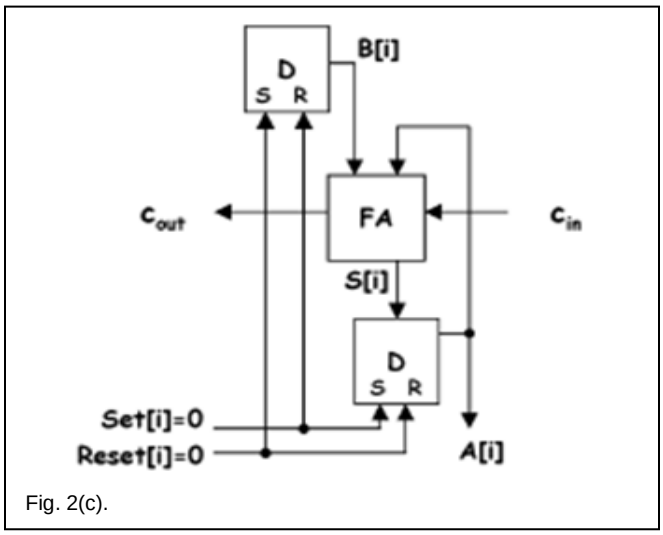

Fig. 2(c).

In Fig. 2(a) we present the configuration that drives the CUT inputs when A[i] = 1 is required. Set[i] = 1 and Reset[i] = 0 and hence A[i] = 1 and B[i] = 0. Then the output is equal to 1, and Cin is transferred to Cout.

In Fig. 2(b), we present the configuration that drives the CUT inputs when A[i] = 0 is required. Set[i] = 0 and Reset[i] = 1 and hence A[i] = 0 and B[i] = 1. Then, the output is equal to 0 and Cin is transferred to Cout.

In Fig. 2(c), we present the configuration that drives the CUT inputs when A[i] = '–' is required. Set[i] = 0 and Reset[i] = 0. The D input of the flip-flop of register B is driven by either 1 or 0, depending on the value that will be added to the accumulator inputs in order to generate satisfactorily random test patterns to the inputs of the CUT.

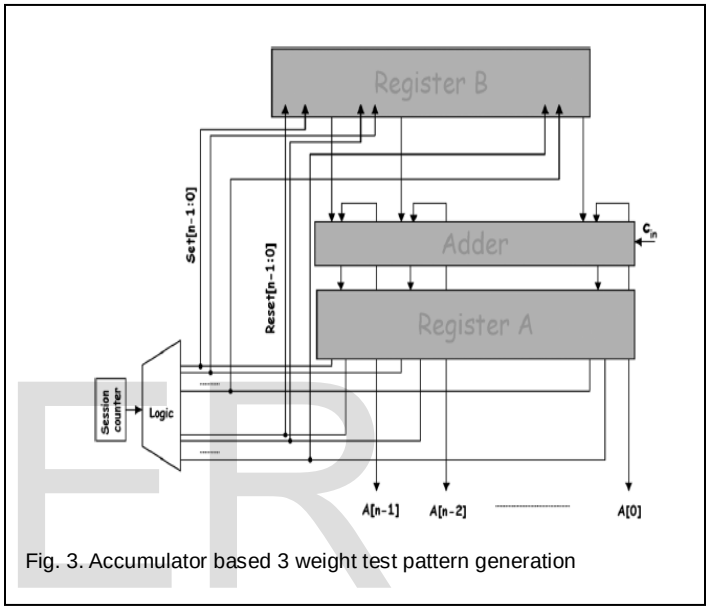

Fig. 3. Accumulator based 3 weight test pattern generation

In Fig. 3, we present the general configuration of accumulator based 3-weight test pattern generation scheme. The Logic module provides the Set[n-1:0] and Reset[n-1:0] signals that drive the S and R inputs of the Register A and Register B inputs. Note that the signals that drive the S inputs of the flip-flops of Register A, also drive the R inputs of the flip-flops of Register B and vice versa.

4 Simulation Results

Modelsim is a simulation tool for hardware design which provides behavioural simulation of a number of languages, i.e. Verilog, VHDL, and System C. Verilog HDL is an industry standard language used to create analog, digital and mixed-signal circuits. HDL's are languages which are used to describe the functionality of a piece of hardware as opposed to the execution of sequential instructions like that in a regular software application.

Verilog code is generated for this accumulator circuit using Modelsim 6.4c.Then RTL schematic is produced using Xilinx 13.2. These test patterns are implemented in FPGA SPARTAN 3 (xc3s400 pq208) to realize and verify digital designs.

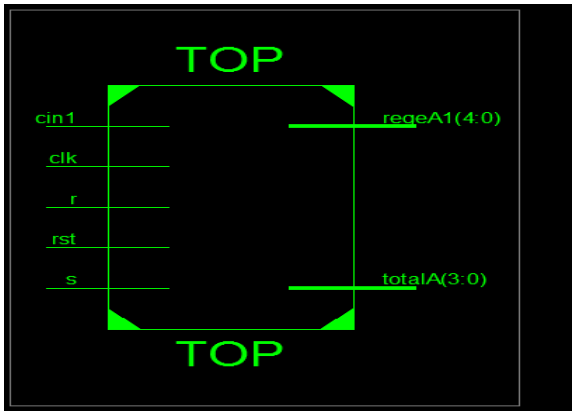

Fig. 4. Block diagram

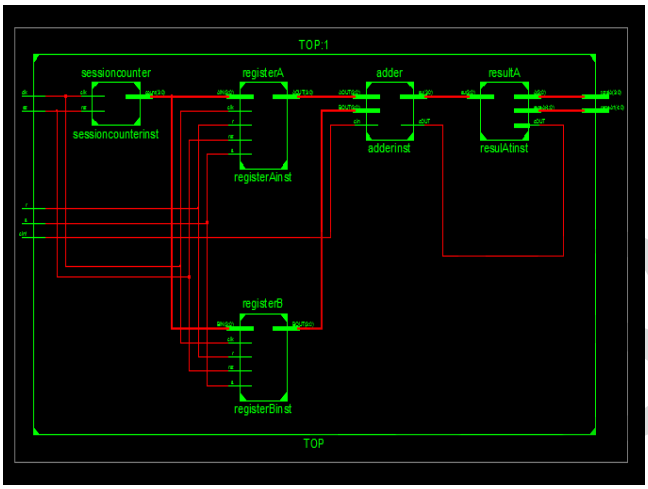

Fig. 5. RTL Schematic

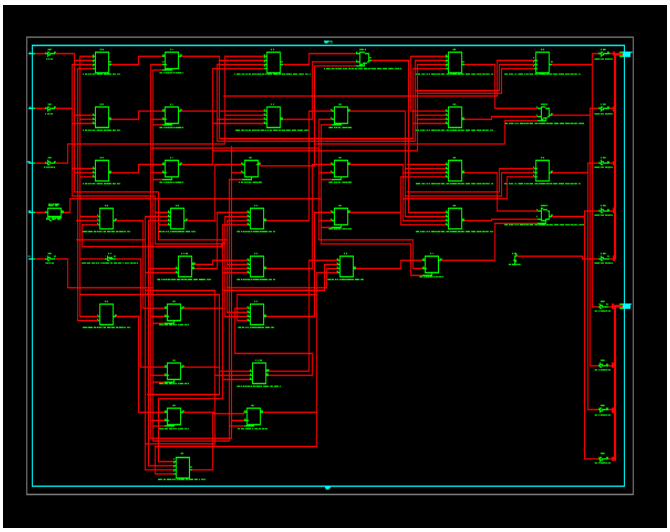

Fig. 6. Techynology Schematic

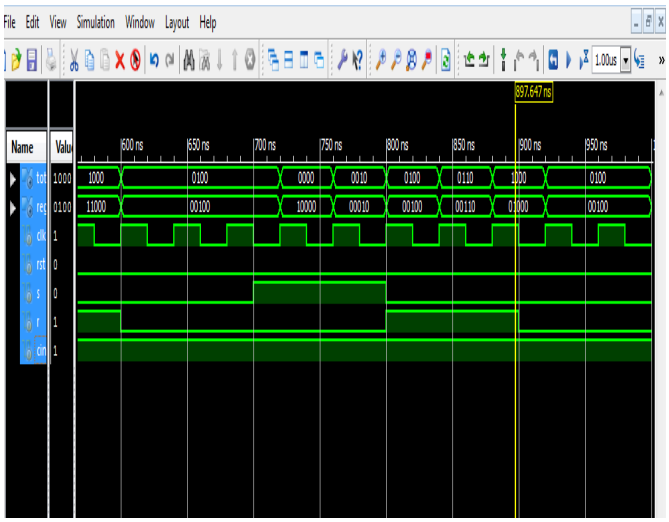

Fig. 7. Simulation Results

5 Conclusion

We have presented an accumulator-based 3 weight (0, 0.5, and 1) test-per-clock generation scheme, which can be utilized to efficiently generate weighted test patterns without any modification in the structure of the adder design. Comparisons with a previously proposed accumulator based 3 weight pattern generation technique results in low hardware overhead while no redesign of the accumulator is required, this results into reduction of test application time. Due to weighted pseudorandom BIST scheme the fault coverage is significantly high.

Acknowledgment

We thank to Dr.A.K. Kureshi, Prof.V.S.Dhondge & Prof. V.G.Puranik of E&TC Dept., Vishwabharti's Academy College of Engineering., Ahmednagar, for their contribution of this work.

References

[1] P. Bardell, W. McAnney, and J. Savir, Built-In Test For VLSI: Pseudorandom Techniques. New York: Wiley, 1987

[2] P. Hortensius, R. McLeod, W. Pries, M. Miller, and H.Card, "Cellular automata-based pseudorandom Generators for built-in self test," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 8, no. 8, pp. 842–859, Aug. 1989.

[3] A. Stroele, "A self test approach using accumulators as test pattern generators," in Proc. Int. Symp. Circuits Syst.,1995,pp. 2120–2123.

[4] H. J. Wunderlich, "Multiple distributions for biased random test patterns," in Proc. IEEE Int. Test Conf., 1988, pp. 236–244.

[5] I. Pomeranz and S. M. Reddy, "3 weights pseudo-random test generation based on a deterministic test set for combinational and sequential circuits," IEEE Trans. Compute.-Aided Des. Integer. Circuits Syst., vol. 12, no. 7, pp. 1050–1058, Jul. 1993.

[6] K. Radecka, J. Rajski, and J. Tyszer, "Arithmetic built-in self-test for DSP cores," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 16, no. 11, pp. 1358–1369, Nov.1997.

[7] J. Rajski and J. Tyszer, Arithmetic Built-In Self Test For Embedded Systems. Upper Saddle River, NJ: Prentice Hall PTR, 1998.

[8] S. Wang, "Low hardware overhead scan based 3-weight Weighted random BIST," in Proc. IEEE Int. Test Conf., 2001, pp. 868–877.

[9] S. Zhang, S. C. Seth, and B. B. Bhattacharya, "Efficient test Compaction for pseudo-random testing," in Proc. 14th Asian Test Symp., 2005, pp. 337–342.
[10] J. Saver, "Distributed generation of weighted random Patterns," IEEE Trans. Compute., vol. 48, no. 12, pp. 1364–1368, Dec. 1999.
[11] I. Voyiatzis, D. Gizopoulos, and A. Paschalis, "Accumulator-based weighted pattern generation," Presented at the IEEE Int. Line TestSymp., Saint Raphael, French Riviera, France, Jul. 2005.
[12] F. Burgles and H. Fujiwara, "A neutral net list of 10 Combinational benchmarks circuits and a target translator in FORTRAN," presented at the Int. Symp. Circuits Syst., Kyoto, Japan, 1985.
[13] S. Wang, "Low hardware overhead scan based 3-weight Weighted random BIST architectures," U.S. Patent 6 886 124, Apr. 26, 2005.
[14] S. Manich, L. Garcia-Deiros, and J. Figueras, "Minimizing test time in arithmetic test-pattern generators with constrained memory resources,"IEEE Trans. Comput.Aided Des. Integr. Circuits Syst., vol. 26, no. 11,pp. 2046–2058, Nov. 2007.
[15] S. Manich, L. Garcia-Deiros, and J. Figueras, "Minimizing test time in arithmetic test-pattern generators with constrained memory resources,"IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 26, no. 11,pp. 2046–2058, Nov. 2007.
[16] S. Manich, L. Garcia, L. Balado, J. Rius, R. Rodriguez, and J. Figueras, "Improving the efficiency of arithmetic bist by combining targeted and generalpurpose patterns," presented at the Des. Circuits Integr. Syst. (DCIS), Bordeaux, France, 2004.
[17] S. Manich, L. Garcia, L. Balado, E. Lupon, J. Rius, R. Rodriguez, and J. Figueras, "On the selection of efficient arithmetic additive test pattern generators," in Proc. Eur. Test Workshop, 2003, pp. 9–14.
[18] I. Voyiatzis, "An accumulator—based compaction scheme with reduced aliasing for on-line BIST of rams," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 9, pp. 1248–1251, Sep. 2008.
[19] M. B. Santos, I. C. Teixeira, J. P. Teixeira, S. Manich, R. Rodriguez, and J. Figueras, "RTL level preparation of high- quality/low-energy/low-power BIST," in Proc. Int. Test Conf., 2002, pp. 814–823.
[20] C. Albrecht, "IWLS 2005 Benchmarks," Lake Arrowhead, CA, 2005 [Online]. Available http://www.iwls.org/ iwls2005/benchmarks.html.