IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 7, Issue 3 (Sep. - Oct. 2013), PP 86-91
www.iosrjournals.org

Design of accumulator Based 3-Weight Pattern Generation using LP-LSFR

G. Sindhu[1], B. Balaji[2]
1(M.Tech VLSI system design Deparment of ECE, Vaagdevi College Of Engineering,warangal,A.p, India.).
2(Assistant Professor, Department Of ECE, Vaagdevi College Of Engineering, Warangal,A.P, India.).

Abstract: *The objective of the BIST is to reduce power dissipation without affecting the fault coverage. Weighted pseudorandom built-in self - test (BIST) schemes have been utilized in order to drive down the number of vectors to achieve complete fault coverage in BIST applications. Weighted sets comprising three weights, namely 0, 1, and 0.5 have been successfully utilized so far for test pattern generation, since they result in both low testing time and low consumed power. In this approach, the single input change patterns generated by a counter and a gray code generator are Exclusive–ORed with the seed generated by the low power linear feedback shift register [LP-LFSR]. Since accumulators are commonly found in current VLSI chips, this scheme can be efficiently utilized to drive down the hardware of BIST pattern generation, as well. From the implementation results, it is verified that the testing power for the proposed method is reduced by a significant percentage.*

Keywords: *Built-in self- test (BIST), test per clock, VLSI testing, weighted test pattern generation, low power linear feedback shift register [LP-LFSR].*

I. Introduction

Pseudorandom built-in self- test (BIST) generators have been widely utilized to test integrated circuits and systems. The arsenal of pseudorandom generators includes, among others, linear feedback shift registers(LFSRs) [1], cellular automata [2], and accumulators driven bya constant value [3]. For circuits with hard-to-detect faults, a large number of random 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 ofa "0" or a "1" on a given input from 0.5 (for pure pseudorandom tests)to some other value [10], [15].

Weighted random pattern generation methods relying on a single weight assignment usually fail to achieve complete fault coverage using a reasonable 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 patterns can detect all faults [4]. Approaches to derive weight assignments forgiven deterministic tests are attractive since they have the potential toallow complete coverage with a significantly smaller number of test patterns [10].

In order to minimize the hardware implementation cost, other schemes based on multiple weight assignments utilized weights 0, 1,and 0.5. This approach boils down to keeping some outputs of the generator steady (to either 0 or 1) and letting the remaining outputs change values (pseudo-) randomly (weight 0.5). This approach, apart from reducing the hardware overhead has beneficial 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. 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 minimize the hardware implementation cost. Wang [8], [13] proposed a 3-weight random pattern generator based on scan chains utilizing 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 has fired 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 has been shown to result in low hardware overhead and low impact on the circuit normal operating speed [22]. In [22], 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 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. 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 utilized to drastically reduce 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 data-path, a practice that is generally discouraged 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 of the adder; and hence, 3) does not affect the operating speed of the adder. Furthermore, the proposed scheme compares favorably to the scheme proposed in [11] and [22] in terms of the required hardware overhead.

A better low power can be achieved by using single input change pattern generators. It is proposed that the combination of LFSR and scan shift register is used to generate random single input charge sequences. In [12], it is proposed that (2m-1) single input change test vectors can be inserted between two adjustment vectors generated by LFSR, m is length of LFSR. The average and peak power are reduced by using the above techniques. Still, the switching activities will be large when clock frequency is high.

This paper is organized as follows. In Section II, the idea underlying the accumulator-based 3-weight generation is presented. In Section III, the design methodology to generate the 3-weight patterns utilizing an accumulator is presented. In Section IV, the proposed scheme is compared to the previously proposed ones. Finally, Section V concludes this paper.

II. System Design Model

A. Accumulator-based 3-weight pattern generation

We shall illustrate the idea of an accumulator-based 3-weight patterngeneration by means of an example. Let us consider the test set for thec17 ISCAS benchmark [12], [31] given in Table I.Starting from this deterministic test set, in order to apply the3-weight pattern generation scheme, one of the schemes proposedin [5], [8], and [9] can be utilized. According to these schemes, atypical weight assignment procedure would involve separating thetest set into two subsets, S1 and S2 as follows:S1={T1,T4} andS2={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.

Table; 1 TEST SET FOR THE C17 BENCHMARK

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

The above reasoning calls for a configuration of the accumulator,where the following conditions are met: 1) an accumulator output canbe constantly driven by "1" or "0" and 2) an accumulator cell withits output constantly driven to "1" or "0" allows the carry input of thestage to transfer to its carry output unchanged. This latter condition isrequired in order to effectively generate pseudorandom patterns in theaccumulator outputs whose weight assignment is "_".

Table: TRUTH TABLE OF THE FULL ADDER

#	Cin	A[i]	B[i]	S[i]	C_{out}	Comment
1	0	0	0	0	0	
2	**0**	0	1	1	**0**	$C_{out} = C_{in}$
3	**0**	1	0	1	**0**	$C_{out} = C_{in}$
4	0	1	1	0	1	
5	1	0	0	1	0	
6	**1**	0	1	0	**1**	$C_{out} = C_{in}$
7	**1**	1	0	0	**1**	$C_{out} = C_{in}$
8	1	1	1	1	1	

B. Design methodology

The implementation of the weighted-pattern generation scheme isbased on the full adder truth table, presented in Table II. From Table IIwe can see that in lines #2, #3, #6, and #7 of the truth table, $C_{out} = C_{in}$.Therefore, in order to transfer the carry input to the carry output, it isenough to set A[i] = NOT (B[i]). The proposed scheme is based onthis observation.

The implementation of the proposed weighted pattern generationscheme is based on the accumulator cell presented in Fig. 1, which consistsof a Full Adder (FA) cell and a D-type flip-flop with asynchronousset and reset inputs whose output is also driven to one of the full adderinputs. In Fig. 1, we assume, without loss of generality, that the set andreset are active high signals. In the same figure the respective cell ofthe driving register B[i] is also shown. For this accumulator cell, oneout of three configurations can be utilized, as shown in Fig. 2.

In Fig. 2(a) we present the configuration that drives the CUT inputswhen 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 C_{in} istransferred to C_{out}. In Fig. 2(b), we present the configuration that drives the CUT inputswhen 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 C_{in} istransferred to C_{out}.In Fig. 2(c), we present the configuration that drives the CUT inputswhen A[i] = "_" is required. Set[i] = 0 and Reset[i] = 0.

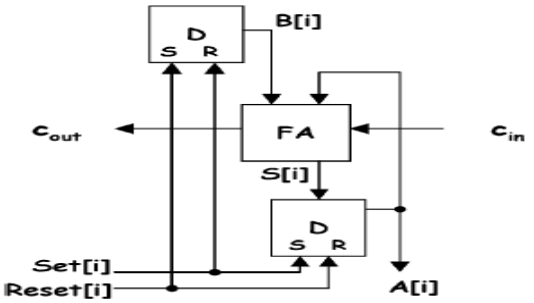


Fig. 1.Accumulator cell.

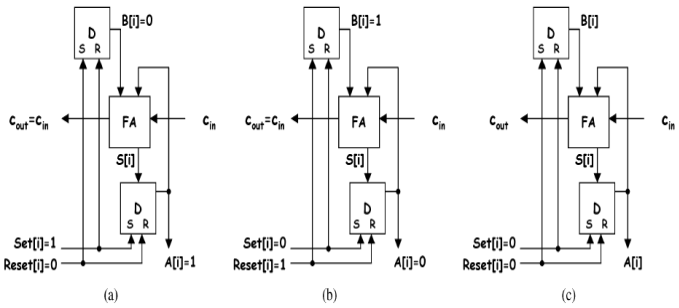


Fig. 2. Configurations of the accumulator cell of Fig. 1.

The Dinput of the flip-flop of register B is driven by either 1 or 0, dependingon the value that will be added to the accumulator inputs in order togenerate satisfactorily random patterns to the inputs of the CUT.In Fig. 3, the general configuration of the proposed scheme is presented.The Logic module provides the Set[n-1:0] and Reset[n-1:0] signalsthat 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 ofRegister A, also drive the R inputs of the flip-flops of Register B andvice versa.

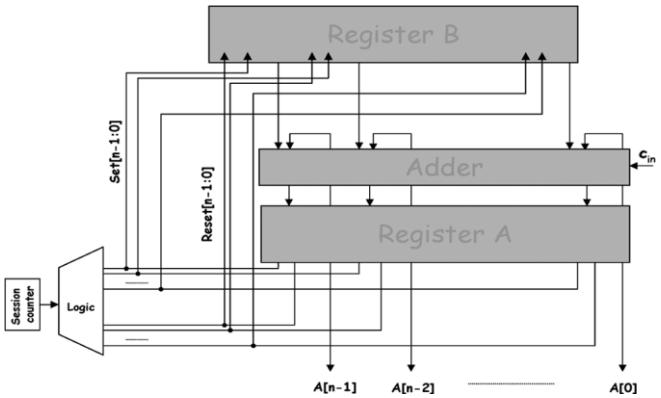
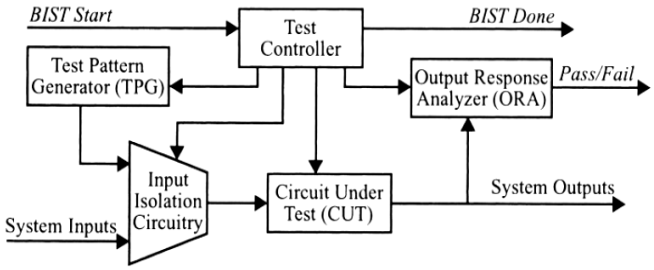


Fig. 3.Proposed scheme.

C. BIST Approach



BIST is a design for testability (DFT) technique in which testing is carried out using built –in hardware features. Since testing is built into the hardware, it is faster and efficient. The BIST architecture shown in fig.1 needs three additional hardware blocks such as a pattern generator, a response analyzer and a test controller to a digital circuit. For pattern generators, we can use either a ROM with stored patterns, or a counter or a linear feedback shift register (LFSR).A response analyzer is a compactor with stored responses or an LFSR used as a signature analyzer. A controller provides a control signal to activate all the blocks.
BIST has some major drawbacks where architecture is based on the linear feedback shift register[LFSR].The circuit introduces more switching activities in the circuit under test (CUT)during test than that during normal operation. It causes excessive power dissipation and results in delay penalty into the design.

D. ALGORITHM FOR LP-LFSR

The algorithm for LP-LFSR is given below:
• Consider a N-bit external (or) internal linear feedback shift register [n>2].
• For example n-bit, external LFSR is taken, which consists of n-flip flops in series. A common clocksignal is applied as control signal for all flip flop.
• For exchanging the output of adjacent flip flops, multiplexers are used. The output of the last stage flip flop is taken as a select line.
• If the last stage flip flop output is one, any one of the flip flop output is swapped with its adjacent flip flop output value.
• If the last stage flip flop output is Zero, no swapping will be carried out.
• The output from other flip flops will be taken as such.
• If the LFSR is moved through a complete cycle of 2n states then the transitions expected are 2n-1.When the output of the adjacent flip flops are swapped, the expected transitions are 2n-2.Thus the transitions produced are reduced by 50% compared with original LFSR. The transition reduction is concentrated mainly on any one of the multiplexer output.
Gray converter modifies the counter output such that two successive values of its output are differing in only one bit. Gray converters can be implemented as shown below.

g[n-1]= k [n-1]
g[n-2]= k [n-1] XOR k[n-2]
.
.
g[2]= k [2] XOR k[3]

g[1]= k[1] XOR k[2]
g[0]= k[0] XOR k[1]

In [12] it is stated that that the conventional LFSR's outputs cannot be taken as the seed directly, because some seeds may share the same vectors. Thus the LP-LFSR should ensure that any two of the signal input changing sequences do not share the same vectors or share as few vectors as possible. Test patterns generated from the proposed structure are implemented as following equations.

x[0] = f[0] XOR g[0]
x[1] = f[1] XOR g[1]
x[2] = f[2] XOR g[2]
x[3] = f[3] XOR g[3]
x[4] = f[4] XOR g[4]
x[5] = f[5] XOR g[5]
.
.
.
X [n-1] = f[n-1] XOR g[n-1]

Thus the XOR result of the sequences is single input changing sequence. In turn reduces the switching activity and so power dissipation is very less compared with conventional LFSR. Fig. 4 is an example of counter and its respective gray value. It is shown that all values of g[2:0] are single input changing patterns.

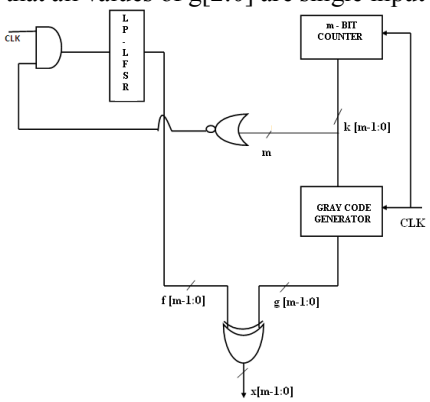
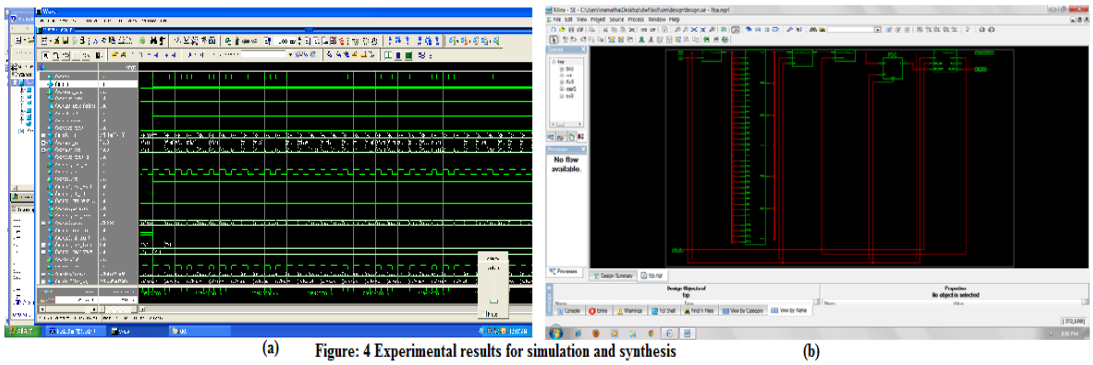


Figure 3. Low Power Test Pattern Generator

III. Simulation Results

To validate the effectiveness of the proposed method, we select weight generator using conventional linear feedback shift register [LFSR] for comparison with proposed system. Table 1 shows the power consumption comparison between TPG using conventional LFSR and the proposed LP-LFSR after applying the generated patterns to the 4x4 and 8x8 Braun array multipliers respectively. The generated test patterns from above two techniques are used to test the synchronous pipelined 4x4 and 8x8 Braun array multipliers. The average test power consumption for the proposed LP-LFSR is presented. The test patterns generated from this LP-LFSR is tested with 4x4 and 8x8 Braun array synchronous pipelined multipliers.



(a) **Figure: 4 Experimental results for simulation and synthesis** (b)

IV. 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 patterns without altering the structure of the adder. The seed generated from (LP-LFSR) is Ex-ORed with the single input changing sequences generated from gray code generator, which effectively reduces the switching activities among the test patterns. Thus the proposed method significantly reduces the power consumption during testing mode with minimum number of switching activities using LP-LFSR in place of conventional LFSR in the circuit used for test pattern generator. From the implementation results, it is verified that the proposed method gives better power reduction compared to the exiting method.

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 testapproach 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 weight pseudo-random test generation based on a deterministic test set for combinational and sequential circuits," *IEEE Trans. Comput.-Aided Des. Integr. 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. Savir, "Distributed generation of weighted random patterns," *IEEE Trans. Comput.*, 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 Test Symp., Saint Raphael, French Riviera, France, Jul. 2005.

[12] R.H.He,X.W.Li and Y.Z.Gong," A scheme for low power BIST test pattern generator," micro electronics& computer,no.2,pp.36-39 Feb.2003.

[13] S. Wang, "Low hardware overhead scan based 3-weight weighted random BIST architectures," U.S. Patent 6 886 124, Apr. 26, 2005.

[14] C. Hamacher, Z. Vranesic, and S. Zaky, *Computer Organization*. New York: McGraw Hill, 2002.

[15] F. Brglez, C. Gloster, and G. Kedem, "Hardware-based weighted random pattern generation for boundary scan," in *Proc. IEEE Int. Test Conf. (ITC)*, 1989, pp. 264–274.

[16] H.-J. Wunderlich, "Self test using unequiprobable random patterns," in *Proc. 17th Int. Symp.Fault-Tolerant Comput. (FTCS)*, 1987, pp. 258–263.

[17] O. Novák, Z. Pløva, J. Nosek, A. Hlawiczka, T. Garbolino, and K. Gucwa, "Test-per-clock logic BIST with semi-deterministic test patterns and zero-aliasing compactor," *J. Electron. Testing: Theor. Appl.*, vol. 20, no. 1, pp. 109–122, Feb. 2004.

[18] Y. Son, J. Chong, and G. Russell, "E-BIST: Enhanced test-per-clock BIST architecture," *IEE Proc.—Comput. Digit.Techn.*, vol. 149, pp. 9–15, Jan 2002.

[19] K. Yamaguchi, M. Inoue, and H. Fujiwara, "Hierarchical BIST: Testper- clock BIST with low overhead," *Electron. Commun. Japan (Part II: Electron.)*, vol. 90, no. 6, pp. 47–58, Jun. 2007.

[20] E. Kalligeros, X. Kavousianos, D. Bakalis, and D. Nikolos, "An efficient seeds selection method FOR lfsr-based test-per-clock BIST," in *Proc. Int. Symp. Quality Electron.Des.*, 2002, p. 261.

[21] A. D. Singh, M. Seuring, M. Gossel, and E. S. Sogomonyan, "Multimode scan: Test per clock BIST for IP cores," *ACM Trans. Design Autom. Electr. Syst.*, vol. 8, no. 4, pp. 491–505, Oct. 2003.

[22] 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.