ISSN 2348-375X

Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

ACCUMULATOR BASED WEIGHTED TEST PATTERN GENERATION FOR MSIC VECTORS

Manimegala.M[1*], Jayachandran T[2]

[1] PG Student, Department of Electronics and Communication Engineering, Nandha Engineering College, Perundurai, Erode-638 052, Tamilnadu, India
[2] Associate Professor, Department of Electronics and Communication Engineering, Nandha Engineering College, Perundurai, Erode-638 052, Tamilnadu, India

Received: 14-12-2013; Revised: 13-01-2014; Accepted: 12-02-2014

*Corresponding Author: **Ms Manimegala.M**
PG Student, Department of Electronics and Communication Engineering College, Perundurai, Erode-638 052, Tamilnadu, India Email:angelinmegala@gmail.com

ABSTRACT

BUILT IN SELF TEST (BIST) can effectively reduce the difficulty and complexity of VLSI testing. A reconfigurable Johnson counter and a scalable SIC counter are developed to generate a class of minimum transition sequences. The performances of the designed TPGs and the circuits under test with 45 nm are evaluated. Simulation results with ISCAS benchmarks demonstrate that MSIC can save test power and impose no more than 7.5% overhead for a scan design. In this paper an accumulator-based 3-weight test pattern generation scheme is presented; the proposed scheme generates set of patterns with weights 0, 0.5, and 1. 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. Comparisons with previously presented schemes indicate that the proposed scheme compares favorably with respect to the required hardware.

Keywords: Built-in self test (BIST), test per clock, VLSI testing, weighted test pattern generation.

INTRODUCTION

In conventional BIST architectures, the linear feedback shift register (LFSR) is commonly used in the test pattern generators (TPGs) and output response analyzers. A major drawback of these architectures is that the pseudorandom patterns generated by the LFSR lead to significantly high switching activities in the CUT[1], which can cause excessive power dissipation. They can also damage the circuit and reduce product yield and lifetime[2,3].

A. Prior Work

Several low-power approaches have also been proposed for scan-based BIST. The architecture in[4] modifies scan-path structures, and lets the CUT inputs remain unchanged during a shift operation. Using multiple scan chains with many scan enable (SE) inputs to activate one scan chain at a time, the TPG proposed in[5] can reduce average power consumption during scan-based tests and the peak power in the CUT. A pseudorandom BIST scheme was proposed to reduce switching activities in scan chains. Other approaches include LT-LFSR[6], a low-transition random TPG[7], and the weighted LFSR[8]. The TPG in[6] can reduce the transitions in the scan inputs by assigning the same value to most neighboring bits in the scan chain. Power reduction is achieved by increasing the correlation between consecutive test patterns[7]. The weighted LFSR decreases energy consumption and increases fault coverage by adding weights to tune the pseudorandom vectors for various probabilities[8]. This paper presents the theory and application of a class of minimum transition sequences. The proposed method generates SIC sequences, and converts them to low transition sequences for each scan chain. This can decrease the switching activity in scan cells during scan-in shifting.

B. Contribution and Paper Organization

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)[9]. 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 . In order to overcome this problem, an accumulator-based weighted pattern generation scheme was proposed in[10]. 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.

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[10]. More precisely: 1) 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 terms of the required hardware overhead[10,11].

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 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.

ACCUMULATOR-BASED PATTERN GENERATION

An accumulator-based test pattern generation scheme that compares favorably to previously proposed schemes. In 1998, 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[12]. 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[13,14]. In order to overcome this problem, an accumulator-based weighted pattern generation scheme was proposed[10]. 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.

TABLE I
TEST SET FOR THE C17 BENCHMARK

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

TABLE II
TRUTH TABLE OF THE FULL ADDER

#	Cin	A[i]	B[i]	S[i]	Cout	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	

We shall illustrate the idea of an accumulator-based 3-weight pattern generation by means of an example. Let us consider test set for the c17 ISCAS benchmark[15,16] given in Table I. 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 isW(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.The above 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 patterns in the accumulator outputs whose weight assignment is "_".

DESIGN METHODOLOGY

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

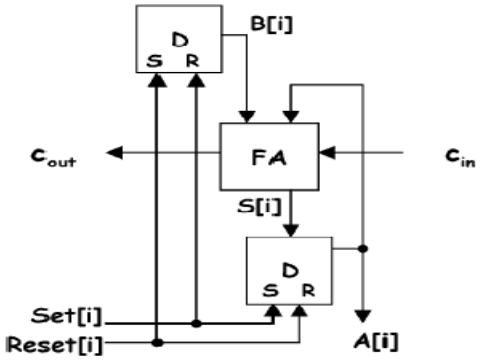

Figure 1: Accumulator cell

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

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

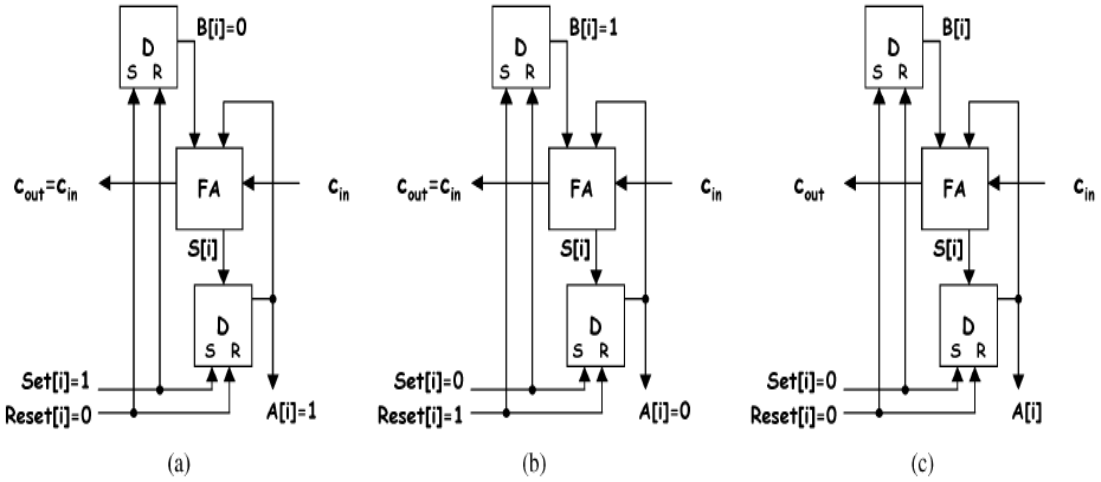

Figure 2: Three configurations of Accumulator cell

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 henceA[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 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] signals that drive the S and R inputs of the Register A and Register B inputs.

PROPOSED SCHEME

The proposed scheme uses a Johnson counter consists of log 2 K bits, where K is the number of test sessions (i.e., weight assignments) of the weighted test set. The scheme proposed in[11] requires the redesign of the adder; more precisely, two NAND gates are inserted in each cell of the ripple-carry adder. In order to provide the inputs to the set and reset inputs of the flip flops, decoding logic is implemented, similar to that in[17].

For the calculation of the delay in the adder operation we have considered both ripple carry and prefix adder implementations. For the comparisons of the ripple carry adder implementations, the adder cell utilized in [10] is considered; in the cell presented in [10], initially the delay from the Cin to Cout of the adder cell is two NAND gates and one XOR gate; in the modified cell proposed in [10], the delay is increased to three NAND and one XOR gate; we have considered that the delay of a NAND gate is one gate equivalent, while the delay of an XOR gate is two gate equivalents.

The proposed methodology to reduce the total test time using an accumulator-based scheme is presented. The scheme operates in test sessions based on triplets of the form (S, I, L), where S is the starting value of the accumulator, I is the increment, and L is he number of cycles the increment is applied before going to the next session. The seeds are stored in a ROM; for the hardware calculation we have considered that a ROM bit is equivalent to 1/4 gates.

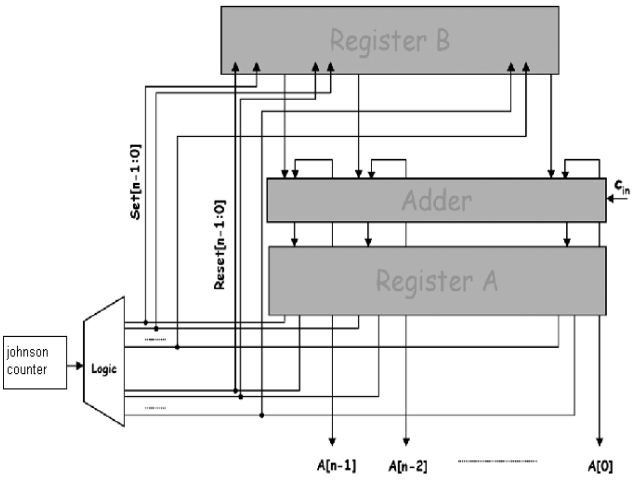

Figure 3: Proposed scheme

COMPARISON

The comparison data for some of the ISCAS'85 and ISCAS' 89 benchmarks are presented in Table V, where the same fault coverage, i.e., 100% is targeted. Next, we present the number of test patterns and hardware overhead for the proposed scheme. From Table V it is trivial to see that the proposed scheme presents an important decrease in the hardware overhead, while the number of tests is comparable, while in some cases it also outperforms The average increase in the number of tests is 19%, while the average decrease in hardware overhead is 98%.

CONCLUSION

We have presented an accumulator-based test-per-clock generation scheme, which can be utilized to efficiently generate weighted patterns without altering the structure of the adder. Comparisons with a previously proposed accumulator-based test pattern generation technique indicate that the hardware overhead of the proposed scheme is lower (approx.75%), while at the same time no redesign of the accumulator is imposed, thus resulting in reduction of 20%–95% in test application time. Comparisons with scan based schemes, showed that the proposed schemes results in lower hardware overhead.

REFERENCES

1. Zorian Y, A distributed BIST control scheme for complex VLSI devices," in 11th Annu. IEEE VLSI Test Symp. Dig. Papers, Apr. 1993; 4–9.
2. Girard P, Survey of low-power testing of VLSI circuits, IEEE Design Test Comput., 2002; 19 (3): 80–90.
3. Abu-Issa A and Quigley S, Bit-swapping LFSR and scan-chain ordering: A novel technique for peak- and average-power reduction in scan-based BIST," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 2009; 28 (5): 755–759.
4. Hertwing A and Wunderlich H, Low power serial built-in self-test, in Proc. Eur. Test Workshop, 1998; 49–53.
5. Wang S and Wei W, A technique to reduce peak current and average power dissipation in scan designs by limited capture, in Proc. Asia South Pacific Design Autom. Conf., 2007; 810–816.
6. Wang S and Gupta S, LT-RTPG: A new test-per-scan BIST TPG for low switching activity, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 2006; 25 (8):1565–1574.
7. Nourani M, Tehranipoor M and Ahmed N, Low transition test pattern generation for BIST-based applications, IEEE Trans. Comput., 2008; 57 (3): 303–315.
8. Zhang S, Roy K and Bhawmik S, POWER TEST: A tool for energy conscious weighted random pattern testing, in Proc. 12th Int. Conf. VLSI Design, Jan. 1999; 416–422.
9. Radecka K, J. Rajski, and J. Tyszer, Arithmetic built-in self-test for DSP cores," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 1997; 16 (11): 1358–1369.
10. Voyiatzis I, Gizopoulos D and Paschalis A, Accumulator-based weighted pattern generation," presented at the IEEE Int. Line Test Symp., Saint Raphael, French Riviera, France, 2005.
11. Manich S, Garcia-Deiros L and Figueras J, Minimizing test time in arithmetic test-pattern generators with constrained memory resources, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2007; 26 (11): 2046–2058.
12. Rajski J and Tyszer J, Arithmetic Built-In Self Test For Embedded Systems. Upper Saddle River, NJ: Prentice Hall PTR, 1998.
13. Manich S, Garcia L, Balado L, Rius J, Rodrøguez R and Figueras J, Improving the efficiency of arithmetic bist by combining targeted and general purpose patterns, presented at the Des. Circuits Integr. Syst. (DCIS), Bordeaux, France, 2004.
14. Manich S, Garcia L, Balado L, Lupon E, Rius J, Rodriguez R, and Figueras J, On the selection of efficient arithmetic additive test pattern generators, in Proc. Eur. Test Workshop, 2003; 9–14.
15. Brglez F and Fujiwara H, A neutral netlist of 10 combinational benchmarks circuits and a target translator in FORTRAN, presented at the Int. Symp. Circuits Syst., Kyoto, Japan, 1985.
16. Albrecht C, IWLS 2005 Benchmarks, Lake Arrowhead, CA, 2005 [Online]. Available: http://www.iwls.org/ iwls2005/benchmarks.html.
17. Wang S, Low hardware overhead scan based 3-weight weighted random BIST, in Proc. IEEE Int. Test Conf., 2001; 868–877.
18. Voyiatzis I, An ALU based BIST scheme for word-organized rams, IEEE Trans. Comput., 2008; 57 (8): 1012–1022.
19. Voyiatzis I, An accumulator—based compaction scheme with reduced aliasing for on-line BIST of rams," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2008; 16 (9): 1248–1251.
20. Zimmerman R, Binary Adder Architectures for Cell-Based VLSI and their Synthesis, Ph.D. dissertation, Swiss Federal Inst. Technol. (ETH), Zurich, Switzerland, 1998.
21. Huang, J. Y. Jou, and S. Y. Kuo, Gauss-elimination-based generation of multiple seed-polynomial pairs for LFSR, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 16, no. 9, pp. 1015–1024, Sep. 1997.

Source of support: Nil, Conflict of interest: None Declared