Design of Accumulator Based 3-Weight Pattern Generation Using Kogge Stone Adder

Sneha Khairnar, Snehal Khedekar, Vanita Kumbhar

Department of Eltronics & Telecommunication, JSPM”s BSIOTR

Abstract- Weighted sets consist of three weights ,namely 0,1,and 0.5 have been successfully use so far for test pattern generation, since they gives both low testing time and low consumed power. Since accumulators are generally found in current VLSI chips, this scheme can be effectively made to use drive down the hardware of BIST pattern generation, as well. From the execution results, it is proved that the testing power for the proposed method is reduced by a considerable percentage. In accumulators used the adder, if we RCA adder or CLA adder its consume the more power and required more gates . If we kogges stone adder its required less power and less gates**.** Comparisons with previously presented schemes indicate that the recommended scheme compares favorably regarding required hardware.

Index Terms- Built–In-Self Test, Weighted Test Pattern Generation, Kogge Stone Adder, and Test per clock

I. INTRODUCTION

Advances in VLSI technology have led to the fabrication of chips that contain a very large number of transistors. The chore of testing such a chip to verify correct functionality is extremely complex and often long delayed. Furthermore to the problem of testing the chips themselves, the incorporation of the chips into systems has caused test generations cost to grow exponentially. A widely accepted approach to deal with the testing problem at the chip level is to include in Built-In-Self-Test (BIST) capacity inside a chip. BIST is a design technique in which units of a circuit are used to test the circuit itself. BIST represents a combination of the concepts of built-in test (BIT) and self-test. Ideally, a BIST scheme should be easy to implement and must provide high fault coverage.

Pseudo-random BIST generators have been tremendously used to test integrated circuits and systems. For circuits whose faults are not detected easily, a large number of random patterns have to be generated before high fault coverage is attained. Therefore, weighted pseudo-random proficiency have been proposed where inputs are biased by changing the probability of a “0” or a “1” for given input from 0.5 (for pure pseudo-random tests) to some other value [4], [6].

Wang [2], [12] proposed a 3-weight random pattern creator based on scan chains using weights 0, 1, and 0.5, in a way similar to [4]. Recently, Zhang et al. [3] renovated the interest in the 3-weight pattern generation schemes, proposing an efficient compression strategy 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 rational interest since it combines low implementation cost with low test time. However, absolute value containing hard-to-detect faults still require extra test hardware either by inserting test points into the chore logic or by storing additional settled test patterns [15], [16]. In order to overcome this problem, an accumulator-based weighted pattern generation strategy was proposed in [10]. The scheme generates test patterns having one out of three weights, 0, 1, and 0.5 therefore it can be utilized to drastically reduce the test application time in accumulator-based test pattern generation. All the same, the scheme proposed in [5] possesses three major drawbacks: 1) it can be used on condition that the adder of the accumulator is a ripple carry adder; 2) it requires rejuvenate of accumulator; this reconstruction, apart from being costly, requires rejuvenate of the core of the data path, a practice that is generally pessimistic in current BIST schemes; and 3) it increases delay, since it affects the normal operating speed of the adder.

II.STUDIES & FINDINGS

We shall elaborate the idea of an accumulator-based 3-weight pattern generation by means of an example. Let us consider the test format for the c17 ISCAS benchmark given in Table I.

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

Table I: TEST SET FOR THE C17 BENCHMARK

Starting from this settled test set, in order to apply the 3-weight pattern generation scheme, one of the schemes put forward in [5], [8], and [9] can be used. According to these schemes, a typical weight assignment procedure would come about separating the test set into two subsets, S1 and S2 as follows: s1 = {T1 , T4} and s. The S2 = {T2 , T3 } eight 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” that the input is constantly provoked by the logic “1” value, and “0” estimate that the input is driven by the logic “0” value. In the first assignment, inputs A[2] and A[0] are constantly provoked by “1”, while inputs A[4], A[3], A[1] are pseudo-randomly created (i.e., have weights 0.5). Similarly, in the second weight assignment (subset S2), inputs A[2] and A[0] are constantly provoked by “0”, input A[1] is goaded by “1” and inputs A[4] and A[3] are pseudo-randomly created. The above argument calls for a configuration of the accumulator, where the following circumstances are met: 1) an accumulator output can be constantly provoked by “1” or “0” and 2) an accumulator cell with its output continuously goaded to “1” or “0” allows the carry input of the stage to transfer to its carry output unvarying. This latter condition is required in order to efficaciously generate pseudo-random patterns in the accumulator outputs whose weight assignment is “_”.

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

Table II: TRUTH TABLE OF THE FULL ADDER

Accumulator cell

The execution of the weighted-pattern generation scheme is based on the full adder truth table, presented in Table II. From Table II we can get a look that in lines #2, #3, #6, and #7 of the truth table Cout = Cin. Therefore, in order to bringing the carry input to the carry output, it is enough to set A[i] = NOT (B[i])..

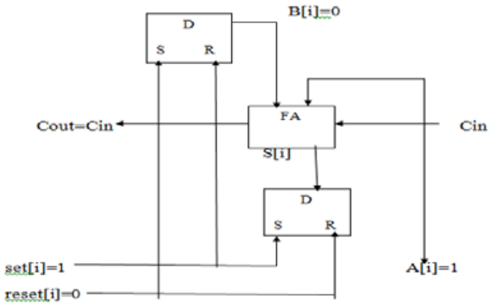

Fig.(a)

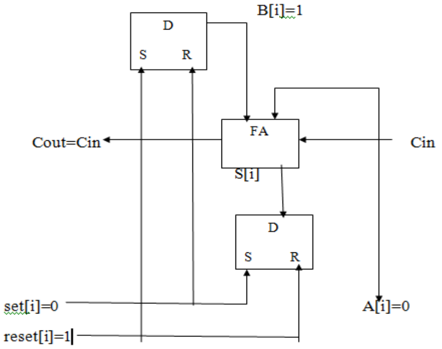

Fig.(b)

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

The D input of the flip-flop of register B is goaded by either 1 or 0, depending on the value that will be added to the accumulator inputs in order to develop satisfactorily random patterns to the inputs of the CUT.

BIST

Testing a circuit every time before they beginning, is called Built-In-Self-Test (BIST). Once BIST finds a fault, the readjustment in connections to convert the faulty part with a fault free one is a design problem and would be not be discussed here. The trend to include more test logic on an ASIC has already been mentioned. Built-in self-test (BIST) is a format of structured-test techniques for combinational and sequential logic, memories, multipliers, and other embedded logic blocks. In each case the principle is to generate test vectors, apply them to the circuit under test (CUT) or device under test (DUT), and then check the response.

KOGGE-STONE ADDER

Adder is one among the basic arithmetic operates. Currently implementing a high speed VLSI style could be an important point and as adders are utilized in various fields of applications, coming up with a high speed adder is one among the necessary surface. In this paper we designed and imposed a high speed Kogge Stone parallel prefix adder of 8, 16 to be meted out and contrast with Carry Look Ahead adder (CLA) and Carry Skip Adder (CSA) and also pointed out the potency of Kogge Stone adder with relevance delay.

The Kogge Stone has low logic depth, high node count, and minimal fan out. While a high node count implicit a larger area, the low logic depth and minimal fan out allow faster performance. There are mainly three estimation stages in Kogge Stone Adder. They are:

1. Preprocessing
2. Carry generation network
3. Post processing

Preprocessing Stage Preprocessing is the first stage where the generate and transmit signals of all the input pairs of signals A and B are created separately for each bit. The logical equations of the transmit and generate signals are given as in following equations: Pi =Ai x or Bi (1) Gi =Ai an dBi (2) Carry Generation Phase Carry generation is the second phase of the KSA. At this stage the carries of all the bits are generated separately for each bit. They are divided into smaller pieces and this overall procedure is carried out side by side for all the bits. Carry generate and Carry propagate bits are used as halfway signals and their logical equations are given as follows: CPi:j= Pi:k+ 1 and Pk:j ...(3) CGi:j= Gi:k+ 1 or (Pi:k+ 1 and Gk:j) ...(4) Post Processing Stage This is the last stage of the KSA which is typical for all types of adders, i.e. calculation of summation of the bit given by the logical Equations (5) and (6):

Ci–1 = (Pi and Cin) or Gi (5)

Si = Pi x orCi–1 . (6)

The Kogge Stone adder is quickest layout, because it sequences logarithmically. Every time we add a mergering step, it doubles the number of bits that can be added. The one and only Problem in KSA is that the number of wires gets incremented as the number of bits increments. The diagram of the 8bit KSA is as shown in Figure 2. As shown in Figures 1 and 2 the wiring increments with respect to the number of bits thus incrementing on chip area. A substantial part of delay is because of the route delay as the KSA has additional wiring Thus in KSA design the area parameter is to be adjust to get the high speed..

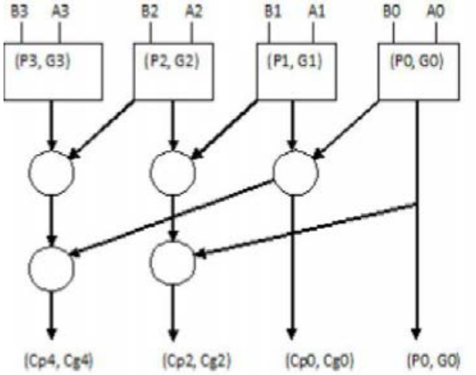

Fig (a): Schematic of the 4 Bit Kogge Stone Adder

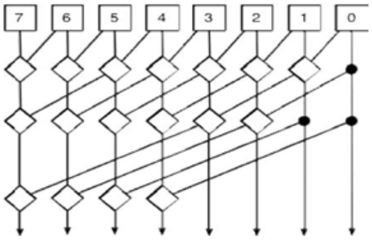

Fig (b): Schematic of the 8 Bit Kogge- Stone Adder.

III.COMPARISONS

If we used the 8 bit adder accumulator require delay is 3.328 ns but if we use the CLA require 3.441ns .If we used the 16 bit adder accumulator require delay is 5.462 ns but if we use the CLA require 5.655ns [19].

a) Proposed System:

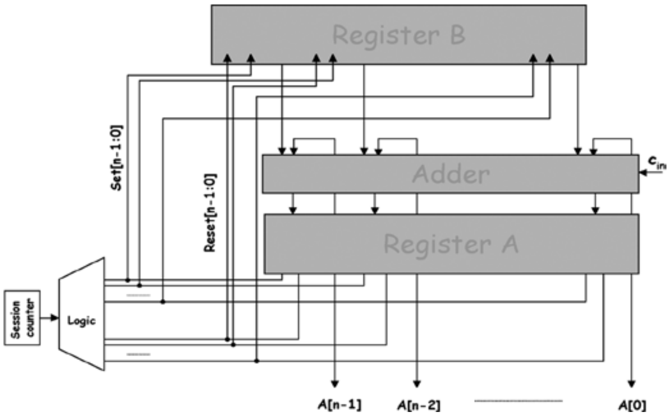

Fig.(a): Proposed scheme.

In Fig.(a), the general configuration of the implemented scheme is presented. The Logic module provides the Set [n-1:0] and Reset[n-1:0] signals that operate the S and R inputs of the Register A and Register B inputs. Note that the signals that operate 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. In above fig adder is replaced by the Kogge stone adder.

b) Benchmark circuit:

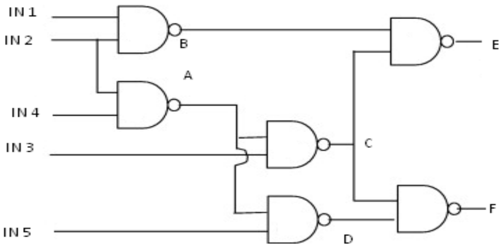

Fig. (b)(1): C17 Benchmark circuit

IV. Simulation Results

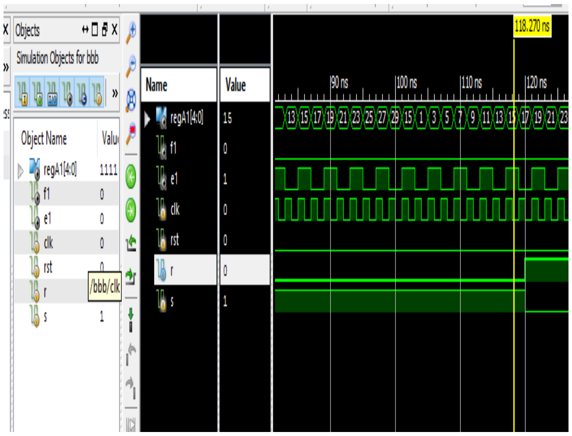

Figure (b)(2): C17 Benchmark circuit with inputs from accumulator cell output waveform

V. Reviewer Comments

>> This research paper represents research with adequate factual analysis & findings.

>> Reviewer board after full consideration accepted for publication

VI. Conclusion

An Accumulator-based 3-weight technique can reduce the hardware implementation cost and fault coverage is more. The power is get reduced and increased fault coverage. The fault coverage is more and it is proved by testing a c17 benchmark circuit. The time for testing a circuit is more and very simple method, because of its automatic test pattern generation. The test patterns are generated automatically for applied test vectors.

REFERENCES

[1] 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.
[2] S. Wang, “Low hardware overhead scan based 3-weight weighted random BIST,” in *Proc. IEEE Int. Test Conf.*, 2001, pp. 868–877.
[3] 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.
[4] J. Savir, “Distributed generation of weighted random patterns,” *IEEE Trans. Comput.*, vol. 48, no. 12, pp. 1364–1368, Dec. 1999.
[4] 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.
[5] 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.
[6]S.Manich,L.Garcia,L.Balado,J.Rius,R.Rodrøguez,andJ.Figueras, “Improvingtheefficiencyofarithmeticbistbycombiningtargetedand general purpose patterns,” presented at the Des. Circuits Integr. Syst. (DCIS), Bordeaux, France, 2004.
[7] S.Manich,L.Garcia,L.Balado,E.Lupon,J.Rius,R.Rodriguez,andJ. Figueras, “On the selection of efficient arithmetic additive test pattern generators,” in Proc. Eur. Test Workshop, 2003, pp. 9–14.

AUTHORS

First Author – Sneha Khairnar, BE (appearing) Electronics &Telecommunication, JSPM’s BSIOTR, sneha.khairnar4@gmail.com
Second Author – Snehal Khedekar, BE (appearing) Electronics &Telecommunication, JSPM’s BSIOTR, snehalkhedekar24@gmail.com
Third Author –Vanita Kumbhar, BE (appearing) Electronics &Telecommunication, JSPM’s BSIOTR,vanita.kumbhar171193@gmail.com

Correspondence Author – Prof.Rupali Rakibe,rakiberupali11@gmail.com