ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

A COMPARATIVE ANALYSIS AND AN IMPLEMENTATION ON THE PATTERN GENERATION OF A 3- WEIGHTED AND ADAPTIVE METHODS OF PSEUDORANDOM TECHNIQUE

S.Raghupathi[1], N.Manikanda sritharan[2],

Faculty Member (ECE / ENGG), IBRI College of Technology, IBRI, Sultanate of Oman[1]

PG Scholar, Shivani Institute of Technology, Trichy -9, Tamilnadu, India[2]

Abstract: Fault detection in VLSI circuit plays vital role in implementation of a design. Widely used, pseudo random Built- in-Self-Test (BIST) is taken into account for this research paper. The store of pseudorandom generators includes, among others, linear feedback shift registers (LFSRs), cellular automata, and accumulators driven by a constant value. Categories of pseudorandom BIST, the 3- weighted pattern Test Generation (TG) and Adaptive pattern Test Generation (TG) were been analyzed with various factors such as design,security,efficiency,performance and accuracy to define the appropriate method for the required design. Both the methods are applied to Scan Based process accumulation, for the above said comparison for the proposed scheme. The ultimate aim is to find the Reduction in Hardware overhead and the consumed time. On thorough observation the design is implemented through Spartan-3 FPGA.

Keywords: Pseudorandom, 3 weighted, Adaptive Test Generation (TG), Process Accumulation.

I. INTRODUCTION

Pseudo random 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), cellular automata, and accumulators driven by a constant value. 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 of a "0" or a "1" on a given input from 0.5 (for pure pseudorandom tests) to some other value. 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 fault. Approaches to derive weight assignments for given deterministic tests are attractive since they have the potential to allow complete coverage with a significantly smaller number of test patterns.

1. Minimizing

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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

2. Proposing

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. This has fired the idea of arithmetic BIST (ABIST) 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. 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 proposed. 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 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. 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 terms of the required hardware overhead.

II.ACCUMULATOR BASED 3-WEIGHT PATTERN GENERATION

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

Starting from this deterministic test set, in order to apply the 3-weight pattern generation scheme, one of the schemes proposed in and can be utilized. According to these schemes, a typical weight assignment procedure would involve separating the test set into two subsets, S1 and S2 as follows:

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.

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

Table 1.Test set for C17 Benchmark

#	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	

Table 2. Truth Table for full adder

III. 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, . Therefore, in order to transfer the carry input to the carry output, it is enough to set. The proposed scheme is based on this observation. 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.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.

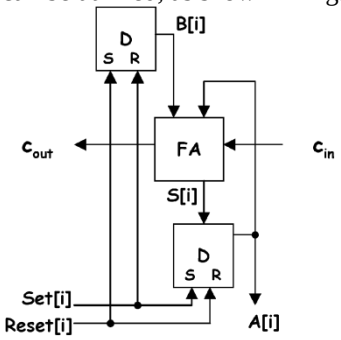

Fig. 1.Accumulator cell.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

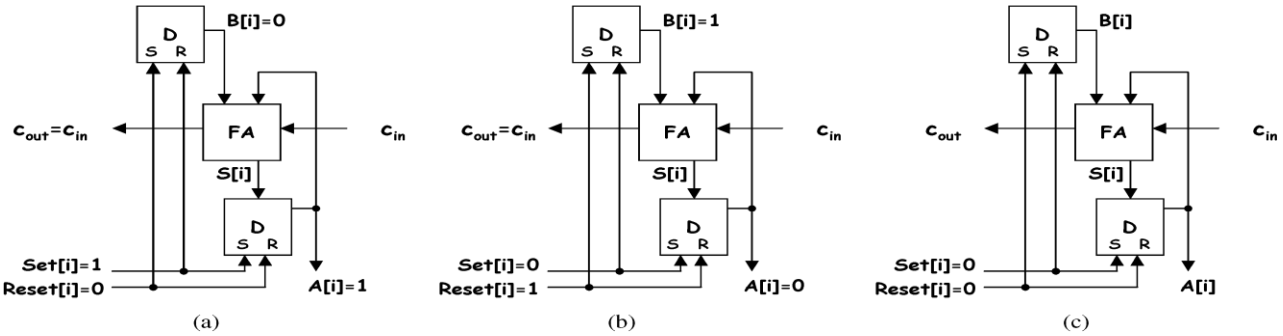

Fig 2. Block diagram of proposed input system

In Fig. 2(b), we present the configuration that drives the CUT inputs when is required. and and hence and . Then, the output is equal to 0 and is transferred to .

In Fig. 2(c), we present the configuration that drives the CUT inputs when " " is required. and . 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. 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.

IV. COMPARISONS

In this section, we shall perform comparisons in three directions. In Section IV-1, we shall compare the proposed scheme with the accumulator-based 3-weight generation scheme that has been proposed. In Section IV-2, shall compare the proposed scheme with the 3-weight scan schemes that have been proposed. In Section IV-3, in order to demonstrate the applicability of the proposed scheme we shall compare the proposed scheme with the accumulator-based test pattern generation scheme proposed.

1. Comparisons With test pattern

The number of test patterns applied by and the proposed scheme is the same, since the test application algorithms that have been invented and applied by previous researchers, can be equally well applied with both implementations. Therefore, the comparison will be performed with respect to: 1) the hardware overhead and 2) the impact on the timing characteristics of the adder of the accumulator. Both schemes require a session counter in order to alter among the different weight sessions; the session counter consists of bits.

where is the number of test sessions (i.e., weight assignments) of the weighted test set. The scheme proposed in requires the redesign of the adder; more precisely, two NAND gates are inserted in each cell of the ripple-carry adder. For the proposed scheme, no modification is imposed on the adder of the accumulator. Therefore, there is no impact on the data path timing characteristics. the adder of the accumulator. Therefore, there is no impact on the data path timing characteristics.

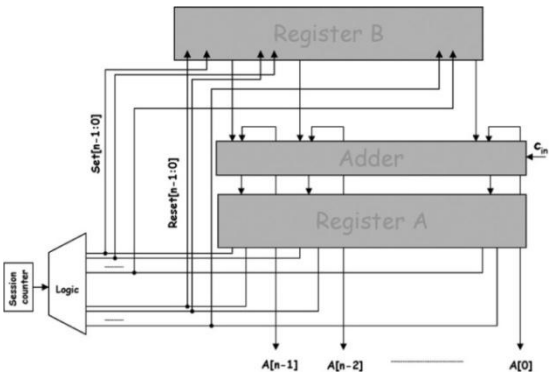

Fig 3.Logic module block

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

circuit	hardware overhead			delay from c_{in} to c_{out}				
					(ripple)		(prefix)	
name	[11]	pr op.	de cr.	[11]	pr op.	de cr.	pr op	de cr.
c880	41%	8%	81%	240	180		24	90%
c1355	28%	7%	74%	164	123		22	87%
c1908	13%	3%	77%	132	99		21	84%
c2670	34%	8%	75%	932	699	20%	32	97%
c3540	11%	5%	57%	200	150		23	89%
c5315	17%	2%	90%	712	534		30	96%
c7552	17%	4%	75%	828	621		31	96%

Table 3.comparsion with test pattern

In Table 3 we present comparison results for some of the ISCAS'85 benchmarks. In the first column of Table 3, we present the benchmark name; in the second and third columns we present the hardware overhead of the accumulator-based scheme proposed and in this work, respectively; in the fourth column we present the delay of the adder in terms of number of gates that the carry signal has to traverse, from the input of the adder (lower stage full adder cell) to the output (higher stage full adder cell), as well as the respective decrease obtained by the proposed scheme.

In Table 3, the hardware overheads are calculated in gate equivalents, where an -input NAND or NOR accounts for 0.5 gates and an inverter accounts for 0.5 gates. For the calculation of the delay in the adder operation (columns under heading "#gates from to ") For the calculation of the delay of prefix adders, the formula obtained by is utilized, where the delay is of the order , where is the number of the adder stages. From Table 3, we can see that the proposed scheme results in 57%–90% decrease in hardware overhead, while at the same time achieving a decrease in operational delay overhead that ranges from 84% to 97% for the considered benchmarks.

2. Comparisons with Scan-Based Schemes

Since the test application algorithms that have been invented and applied by can be equally well applied with the proposed scheme, test application time is similar to that reported there.

In the 3-weight pattern generation scheme proposed by the scan chain is driven by the output of a linear feedback shift register (LFSR). Logic is inserted between the scan chain and the CUT inputs to fix the outputs to the required weight (0, 0.5, or 1). In order to implement the scheme, a scan-structure is assumed. Furthermore, an LFSR required to feed the pseudorandom inputs to the scan inputs is implemented (the number of LFSR stages is , where is the number of scan cells), as well as a scan counter, common to all scan schemes. A number of 3-gate modules is required for every required weighted input (in [Table 4], the hardware overhead is calculated for the ISCAS'85 benchmarks).

benchmark			[22]		proposed	
circuit	h/w	#inp	tests	h/w	#tests	h/w
c880	383	60	1112	36	768	27
c1355	546	41	1409	26	1024	38
c1908	880	33	3198	23	1536	23
c2670	1193	157	1962	1386	4096	101
c3540	1669	50	2167	31	1536	73
c5315	2307	178	1453	189	2048	39
c7552	3512	206	2918	1090	4608	139
s5378	1004	214	2078	791	5120	47
s9234	2027	247	14803	4763	11264	181
s13207	2573	700	14476	7497	12288	61
s15850	3448	611	14902	18438	21504	159
s38584	11448	1464	8449	26235	16384	82
Average values			**5744**	**5042**	**6848**	**81**

Table 4.Comparison with proposed schemes

3. Comparisons With proposed systems

In Table 4, in the first three columns we present the benchmark characteristics (name, hardware overhead in gates, and number of inputs). In the two columns to follow, we present the number of tests required for the scheme and the respective hardware overhead in gate equivalents. Next, we present the number of test patterns and hardware overhead for the proposed scheme.. It is interesting to note that the hardware overhead (with respect to the hardware overhead of the benchmarks) is practical, in contrast to which sometimes exceeds the benchmark hardware (c2670, s5378, s9234, s13207, s15850, s38584). It is also interesting to note the average values presented in the last line of the Table. The average increase in the number of tests is 19%, while the average decrease in hardware overhead is 98 %.

V. 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. Comparisons with a previously proposed accumulator-based 3-weight pattern generation technique indicate that the hardware overhead of the proposed scheme is lower (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 show that the proposed schemes results in lower hardware overhead. Finally, comparisons with the accumulator-based scheme proposed in reveal that the proposed scheme results in significant decrease (98%) in hardware overhead

REFERENCES

[1] P.Bardell, W.McAnney, and J.Savir, Built-In Test for VLSI: pseudorandom Techniques. New York: willey 1987.
[2] A.Stroele, "A Self test approach using accumulator as test pattern generation, in proc, Intsymp circuits syst., 1995, pp 2120-2123.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

[3] H.J Wunderlich, "multiple distributions for biased random test patterns in proc, IEEE Int. Test conf., 1988.
[4] I.Pomeranz and S.M. Reddy 3 weight pseudo random test pattern generation based on a deterministic test for combinational and sequential circuits IEEE Trans, computi-Aided Des.Integr.Circuits syst., vol 12.
[5] 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.

BIOGRAPHY

S. Raghupathi received his **M.Tech** in VLSI form **SASTRA university** in the year 2006. With 9 years of teaching experience the authors research interest isin asynchronous VLSI design

N.Manikandasritharan received his **B.Tech** in Information technology from Lathamathavan engineering college Madurai in 2011. Currently doing **M.E**(VLSI) in shivani institute of technology trichy.