IJCAES
ISSN: 2231-4946
Volume III, Special Issue, August 2013

International Journal of Computer Applications in Engineering Sciences
Special Issue on National Conference on Information and Communication (NCIC'13)
www.caesjournals.org

Accumulator Based Pseudorandom BIST

S.Naveen Pitchumani[1] , Mrs.K.L.Hemalatha[2]
M.E VLSI DESIGN, Easwari Engineering College
Chennai, India.
[1]naveeen90@gmail.com, [2]hemasabs@gmail.com

***Abstract*— In a BIST design, the generation and application of the test vectors and the analysis of the resulting response are part of the circuit (or system) under test. Weighted pseudorandom built-in self-test (BIST) schemes have been used to reduce the number of test vectors for achieving complete fault coverage in BIST applications. 3-weight pattern generation uses only three weights, 0, 0.5 and 1. An accumulator-based 3-weight test pattern generation scheme is presented that tests the MAC unit with different multiplier such as Vedic Multiplier, Booth multiplier, Array multiplier and different adders such as Carry look ahead adder, Ripple carry adder; the proposed scheme generates set of patterns with weights 0, 0.5, and 1. Comparison is done based on the speed of operation of MAC with all these three types of multipliers and finally we carried out BIST with proposed Accumulator Based 3-Weight Pattern Generation by changing the adders used in the MAC.**

***Keywords*— VLSI testing, weighted test pattern generation, Built-in self-test (BIST), Carry Select adder (CSA), Carry look ahead adder (CLA), Ripple carry adder (RCA).**

I. INTRODUCTION

In recent years, developing BIST for all type types of design by reusing the design components has become one of the greatest challenges in very large scale integration (VLSI) design. Pseudorandom built-in self test (BIST) generators have been widely utilized to test integrated circuits and systems. The pseudorandom generators includes, linear feedback shift registers (LFSRs) [1], cellular automata [2], and accumulators driven by a constant value [3]. Large number of random patterns has to be generated before high fault coverage is achieved. Therefore, the proposed pseudorandom pattern uses inputs that 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. And also, the power consumed during test mode operation is always much higher than during normal mode operation.

Multiple weight assignments utilize weights 0, 1, and 0.5 to minimize the hardware implementation cost. This approach boils down to keeping some outputs of the generator steady (to either 0 or 1) and making the remaining outputs to change values (pseudo-) randomly (weight 0.5).Current VLSI circuits, e.g., data path architectures, or digital signal processors commonly contain arithmetic modules [accumulators or arithmetic logic units (ALUs)]. This has led to the idea of arithmetic BIST (ABIST) [4]. The basic idea of ABIST is to utilize accumulators for built-in testing and also it has been shown to result in low hardware overhead and low impact on the circuit normal operating speed [4]–[10]. In [5], accumulator-based test pattern generation scheme that compares favorably to previously proposed schemes.

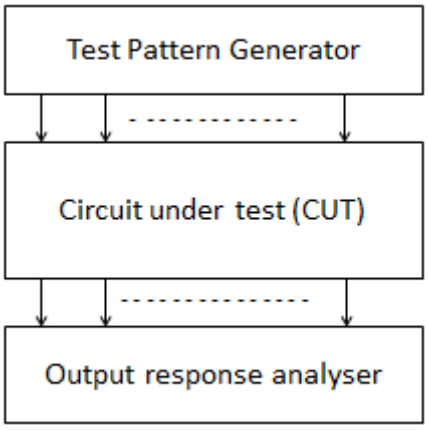

Fig1. Block Diagram for BIST

In [12], it was proved that if the input pattern is properly selected then the test vectors generated by an accumulator whose inputs are driven by a constant pattern can have acceptable pseudorandom characteristics. An accumulator-based weighted pattern generation scheme was proposed in [12], to overcome this problem. In order to reduce the test application time in accumulator-based test pattern generation one of the three weights, namely 0, 1, and 0.5 were used. However, the scheme proposed in [12] possesses three major drawbacks:

1) The adder used in the accumulator must be a ripple carry adder only.
2) Redesigning and modification of accumulator is required, which increases cost and requires redesign of the core of the datapath that is generally discouraged in current BIST schemes.
3) The normal operating speed of the adder is affected which increases delay.

This paper is organized as follows. In Section II, the concept of the accumulator-based 3-weight generation is presented. In Section III, the design methodology of MAC unit utilizing different multiplier and adders are presented. In Section IV, the proposed scheme is compared with the previously proposed ones w.r.t area and also comparison done based on the speed of operation of MAC unit. Finally, Section V concludes this paper.

II. ACCUMULATOR-BASED 3-WEIGHT PATTERN GENERATION

A. Weighted Pattern Testing

Weighted pattern testing is performed by weighting the signal probability (probability that the signal is a _1') for each input to the CUT. Two issues in weighted pattern testing are what set of weights to use and how to generate the weighted signals. For computing weight sets, many techniques have been proposed. It has been shown that for most circuits, multiple weight sets are required to achieve sufficient fault coverage. For BIST, the weight sets must be stored on-chip and control logic is needed to switch between them which can result in a lot of overhead. In order to reduce the BIST overhead for weighted pattern testing, researchers have looked for efficient methods for on-chip generation of weighted patterns.

B. *3-weight pattern generation*

The implementation of the weighted-pattern generation scheme is based on the full adder truth table, presented in Table I. From Table I, we can see that in lines #2, #3, #6, and #7 of the truth table, $C_{out} = C_{in}$. So, 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.

Table I
Full Adder Truth Table

#	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	

The implementation of the proposed weighted pattern generation scheme is based on the accumulator cell presented. in Fig.2, 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. 2, we assume, without loss of generality, that the set and reset are active high signals. In the same Fig.2, the respective cell of the driving register B[i] is also shown. For this accumulator cell, one out of three configurations can be utilized.
Here we present these configuration that drives the CUT inputs

- For A[i] == 1 is required. Set[i] =1 & Reset[i] = 0 hence A[i] = 1 & B[i] = 0. Then output =1.
- For A[i]=0 is required Set[i] = 0,Reset[i] = 1,A[i] =0,
  B[i] =1.Then output = 0.
- For A[i] ='--' is required Set[i]=0,Reset[i]=0, 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.

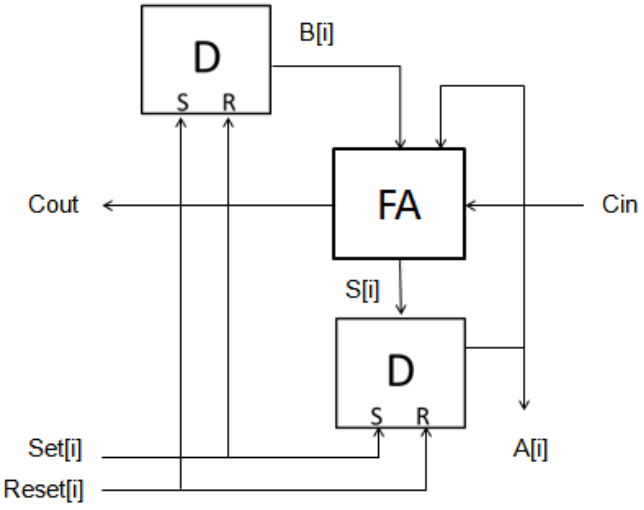

Fig 2.Accumulator cell

III. MAC UNIT

Multiplication is the most important fundamental arithmetic operation. Multiplication-based operations such as Multiply and Accumulate(MAC) is shown in Fig.3 and inner product are among some of the frequently used Computation- Intensive Arithmetic Functions(CIAF) currently implemented in many Digital Signal Processing (DSP)applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

A basic multiplier can be divided into three parts i) partial product generation ii) partial product addition and iii) final addition. In this paper we present three types of methodology for multiplication and two types for addition.

Here we develop

- Array multiplier.
- Booth multiplier.
- Vedic multiplier.

And

- Carry look ahead adder.
- Ripple carry adder.

Finally we compare the speed of MAC with all these three types of multipliers and adder in Table III and finally we carried out BIST with proposed accumulator Based 3-Weight Pattern Generation by changing the adders used in the MAC.

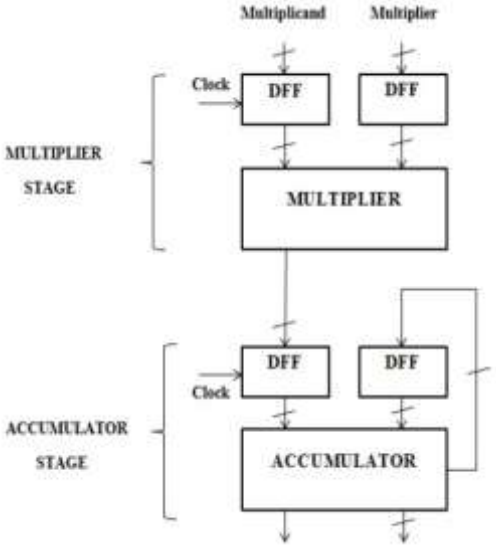

Fig 3.MAC unit

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture is shown in Fig 4.The number of test patterns applied by [12] and the proposed scheme is the same, since the test application algorithms that have been invented and applied by previous researchers, e.g., [13], [14], [15] 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. Table II, presents the proposed scheme is compared with the previously proposed ones w.r.t area.

Both schemes require a session counter in order to alter among the different weight sessions; the session counter consists of log2k bits, where k is the number of test sessions (i.e., weight assignments) of the weighted test set. The scheme proposed in [12] 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 [14]. 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.

Table II, Compares the proposed Accumulator with previously designed Accumulator without redesigning.

TABLE II
Comparison of Hardware Overhead

Parameter	**With RCA (No. of Logic elements)**	**With CSA (No. of Logic elements)**	**% Decrease**
Hardware Overhead	555	409	26.3

The simulation results for Accumulator based 3 weight pattern generation and the MAC unit using Vedic multiplier & CLA is respectively shown below.

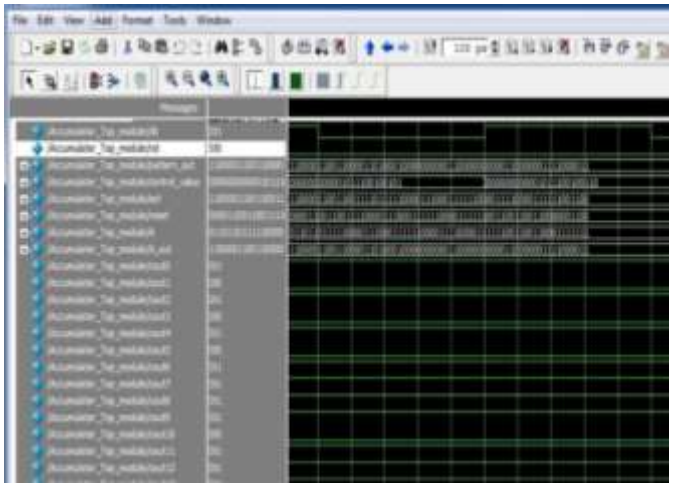

Fig 4.Accumulator-Simulated output

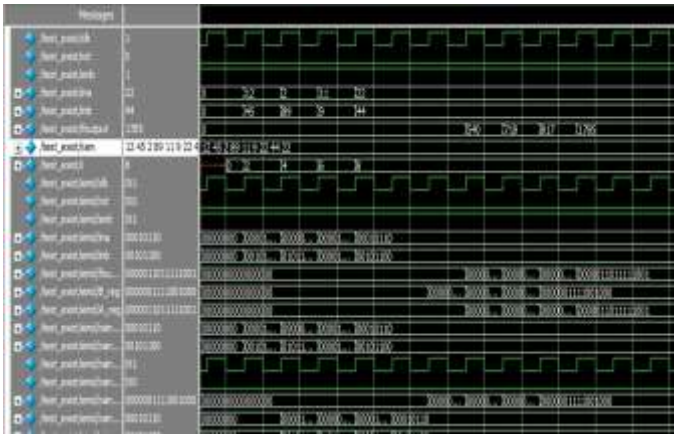

Fig 5.Simulation of MAC unit using Vedic Multiplier and Carry look ahead adder

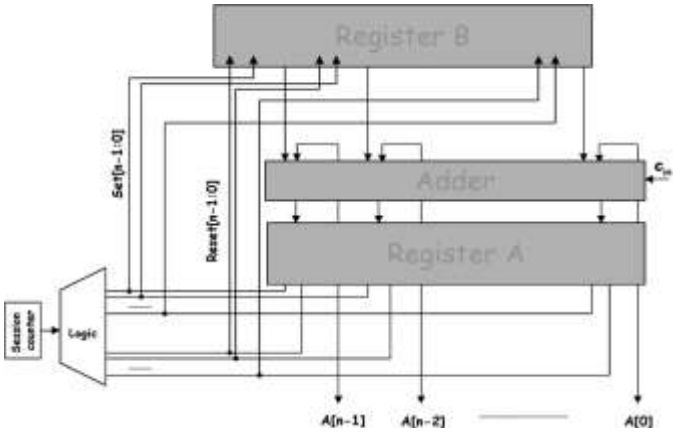

Fig 6.Proposed Accumulator architecture

Table III, Compares the operation speed of MAC Unit using different configurations in terms of frequency.

TABLE III
Speed Analysis of MAC Unit using different configurations

PARAMETER w.r.t SPEED	**CLA**	**RCA**
VEDIC MULTIPLIER	*151.26Mhz*	*100.16Mhz*
BOOTH MULTIPLIER	*143.29Mhz*	*86.69Mhz*
ARRAY MULTIPLIER	*52.75Mhz*	*52.65Mhz*

V. CONCLUSION

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. Accumulator-based 3-weight pattern generation technique indicate that the hardware overhead of the proposed scheme is lower (≈26%), while at the same time no redesign of the accumulator is imposed, thus resulting in reduction in test application time. MAC (multiplier accumulator control unit) is been designed and the accumulator is been reused as BIST, which is available in MAC unit to test that design. The proposed Accumulator based 3-weighted pattern generation is used with both RCA and CLA of adders in order to generate the pattern.

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] 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.
[5] 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.
[6] S. Manich, L. Garcia, and J. Figueras, "Arithmetic test pattern generation:A bit level formulation of the optimization problem," presented at the Des. Circuits Integr. Syst. (DCIS), Lisbon, Portugal, 2005.
[7] S. Manich, L. Garcia, L. Balado, J. Rius, R. Rodrøguez, and J. Figueras,"Improving the efficiency of arithmetic bist by combining targeted and general purpose patterns," presented at the Des. Circuits Integr. Syst.(DCIS), Bordeaux, France, 2004.

[8] 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.
[9] I. Voyiatzis, "An ALU based BIST scheme for word-organized rams," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1012–1022, Aug. 2008.
[10] 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.
[11] J. Rajski and J. Tyszer, Arithmetic Built-In Self-Test For Embedded ystems.Upper Saddle River, NJ: Prentice Hall PTR, 1998.
[12] 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.
[13] 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.
[14] S. Wang, Low hardware overhead scan based 3-weight weighted random BIST,‖ in Proc. IEEE Int. Test Conf., 2001, pp. 868–877.
[15] 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.