FPGA Implementation of high Speed Test Pattern Generator for BIST

Prathyusha. B [1], J.L.V. Ramana Kumari [2], Dr. M. Asha Rani [3]

[1](M.TECH student (VLSI), Department of Electronics and Communication Engineering, VNR VJIET, Hyderabad-90)
[2] (ASSISTANT PROFESSOR, Department of Electronics and Communication Engineering, VNR VJIET, Hyderabad-90)
[3](PROFESSOR, Department of Electronics and Communication Engineering, JNTUH, Hyderabad-85)

ABSTRACT

Pseudorandom built-in self test (BIST) schemes are utilized to test integrated circuits. Testing a circuit is important to produce good yield. For this testing it takes more time for the test vector generation. So it is required to reduce the number of vectors to achieve complete fault coverage in BIST applications. In this paper an accumulator-based 3-weight test pattern generation scheme is presented; this scheme generates set of patterns with weights 0, 0.5, and 1 result in low testing time. Comparisons with non weighted LFSR based schemes indicate that this scheme compares favorably with respect to the testing time. The hardware requirements for this project are FPGA SPARTAN 3E and the softwares used are Modelsim and Xilinx.

Keywords **-** BIST, VLSI testing, weighted test pattern generation

I. INTRODUCTION

BIST is a design-for-testability technique that places the testing functions physically with the circuit under test (CUT). The basic BIST architecture requires the addition of three hardware blocks to a digital circuit: a test pattern generator, a response analyzer, and a test controller. The test pattern generator generates the test patterns for the CUT. A large number of pseudorandom generators include linear feedback shift registers (LFSR'S) [1], cellular automata, accumulators driven by a constant value [2]. A typical response analyzer is a comparator with stored responses compacts and analyzes the test responses to determine correctness of the CUT. A test control block is necessary to activate the test and analyze the responses. Such a test and diagnosis should be quick and have very high fault coverage. An n input circuit would then require 2^n combinations which can be very tiresome on the system with respect to the space and time. Also, more the number of transitions, the power consumed will be more.

Some faults are hard to detect in a circuit so a large number of random patters are to be generated before high fault coverage is achieved. A weighted pseudorandom technique have been proposed, where inputs are biased by changing the probability of "0" or "1" on a given input from 0.5 for pure pseudorandom tests to some other value. Usually weighted random pattern generation technique that rely on a single weight assignment fail to achieve complete fault coverage using number of test patterns even though the weights are suitable for most of the faults. To detect faults with these weight assignments some faults require long test sequences to be detected if they do not match their and propagation requirements. Approaches to derive weight assignments for a given deterministic tests are

efficient as they allow complete fault coverage with small number of test patterns [7].

The choice of the weights 0, 1 and 0.5 was done to minimize the hardware implementation cost [3]. A new efficient compaction scheme for the 3 weight patterns 0, 1, and 0.5 combines low implementation cost with low test time [6]. Current VLSI circuits such as data path architecture or digital signal processing chips commonly contain arithmetic modules such as accumulators or arithmetic logic units.

The use of accumulators [4] for built in testing by the compression of CUT responses has resulted in low hardware overhead and low impact on the circuit normal operating speed. In [5] if the input patterns are properly selected then the test vectors generated by the accumulator whose inputs are driven by a constant pattern have pseudo random characteristics. Modules containing hard to detect faults may require extra test hardware in two ways i.e. by inserting test points in the logic and by storing extra deterministic test patterns.

In this paper novel scheme for accumulator based 3-weight pattern generation is presented and it copes with the inherent drawbacks of the scheme [8]. This paper is organized as follows. In section 2, the idea of the accumulator based 3-weight generation is presented. In section 3, design methodology to generate the 3-weight patterns utilizing the accumulator is presented and process of testing for a 10 input CUT is presented.

II. 3 WEIGHT PATTERN GENERATION

The idea of an accumulator based 3- weight pattern generation by means of an example. Consider the test set for the 10 input circuit shown in figure 4 is given in table 1. Accordingly, weight assignment procedure involves separating the test set into subsets and the weight assignment for these subsets is shown in table 3, where "–"denotes a weight assignment of 0.5. A "1" indicates that the input is constantly driven by logic 1 value and a "0" indicates that the input is constantly driven by logic 0 value. In the subset S1, A[0], A[2], A[5] and A[7] are constantly driven by logic"1" while A[3],A[6], A[8] and A[9] are driven by logic"0" and inputs A[1] and A[4] are pseudo randomly generates a value 0.5. For the subset S2, the inputs A[1],A[2],A[5] and A[7] are constantly driven by logic"1" while A[0] and A[9] are driven by logic"0" and inputs A[3], A[4], A[6] and A[7] are pseudo randomly generates a value 0.5. for subset S3, the inputs A[3] and A[4] are driven by logis"1" while A[0], A[1], A[2], A[5], A[6] and A[9] are driven by logic"0" and inputs A[7] and A[8] are pseudo randomly generates a value 0.5.

Test vectors	Inputs A[0:9]
T1	1010010100
T2	0110010010
T3	1110110100
T4	0001100110
T5	0111111110
T6	0001100000

Table 1. Test set for 10 input CUT

The above reasons configures the accumulator where the following conditions are met: 1)an accumulator output can be constantly driven by logic "1" or logic "0" .

#	Cin	A[i]	B[i]	S[i]	Cout	Comment
1	0	0	0	0	0	
2	**0**	**0**	**1**	**1**	**0**	**Cin=Cout**
3	**0**	**1**	**0**	**1**	**0**	**Cin=Cout**
4	0	1	1	0	1	
5	1	0	0	1	0	
6	**1**	**0**	**1**	**0**	**1**	**Cin=Cout**
7	**1**	**1**	**0**	**0**	**1**	**Cin=Cout**
8	1	1	1	1	1	

Table 2. Truth Table of Full Adder

2)an accumulator cell with its output constantly driven to "1" or "0" allows the carry input to carry to its carry output unchanged. This condition requires to effectively generating the pseudorandom patterns in the accumulator outputs whose weight assignment is "–".

III. DESIGN METHODOLOGY

The implementation of the weighted pattern generation scheme is based on the full adder truth table presented in table 2.From the table 2 we observe that the lines 2,3,6 and 7,***Cout = Cin***, in order to transfer the carry input to the carry output it is sufficient to set A[i]=NOT(B[i]).

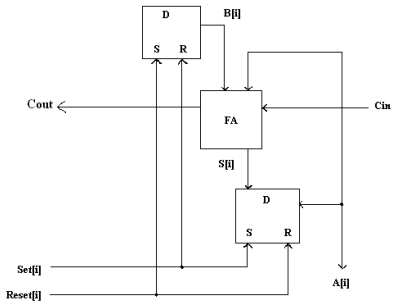

Figure 1. Accumulator Cell

The implementation of the weighted pattern generation scheme is based on the accumulator cell presented in fig 1 that consists of a full adder(FA) and D-type flip flop with asynchronous set and reset inputs whose output is also driven to one of the full adder inputs. A LFSR is used in the 3-weight pattern without loss of generality, that the set and reset are active high signals. The respective cell of the driving register B[i] is also seen. One out of three configurations can be utilized as shown in Fig 2.

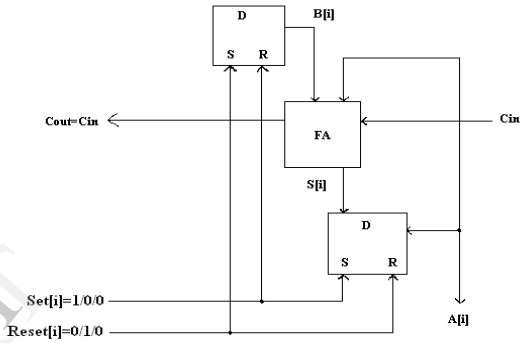

Figure 2. Configurations of accumulator cell

In the Fig 2, 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 equal to 1 and ***cin*** is transferred to ***cout***. The configuration that drives the CUT inputs when A[i] =i is required. Set[i]=0 and reset[i]=1and hence A[i]=0 and B[i]=1.Then the output is equal to 0 and ***Cin*** is transferred to ***cout***. The configuration that drives the CUT inputs when A[i] = "-" is required. Set[i]=0 and reset[i]=0.The D input of FF 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 random patterns to the inputs of the CUT.

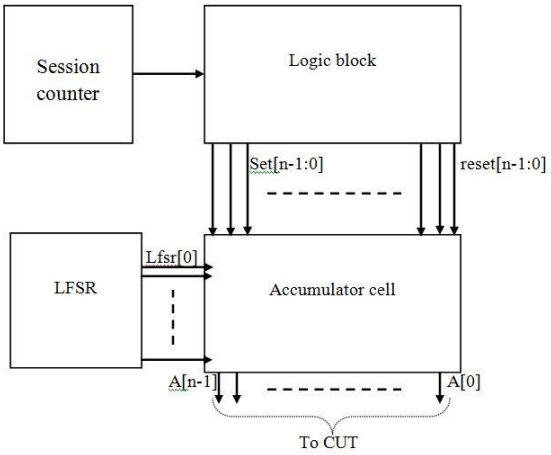

Figure 3. Block diagram of TPG

In fig 3 the general configuration of the scheme is presented. The logic module provides that 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. The block diagram for BIST is shown in figure 5

	{T1,T3}	{T2,T5}	{T4,T6}
N1	1	0	0
N2	-	1	0
N3	1	1	0
N9	0	-	1
N10	-	-	1
N13	1	1	0
N14	0	-	0
N17	1	-	-
N18	0	1	-
N21	0	0	0

Table3. Subsets with weight assignment

In this circuit, it is having different gates with 10 inputs and 3 outputs. As there are 22 nets therefore 42 faults can be generated from these nets. The faults are struck at 0 faults and struck at 1 fault. Faults are imposed for testing purpose at 5 different inputs.

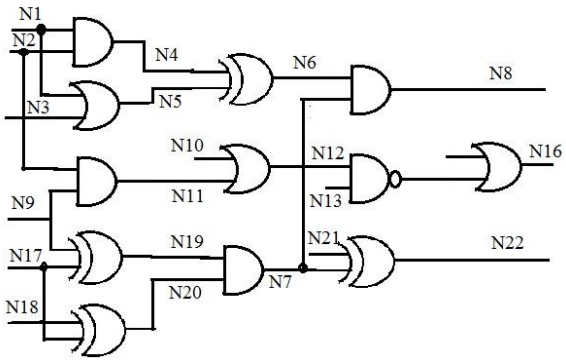

Figure 4. 10 input combination CUT

The controllability and observability of the faults should be observed. At each pattern different faults are covered. At different time slots different patterns are generated. By using accumulator based LFSR all patterns are efficiently generated

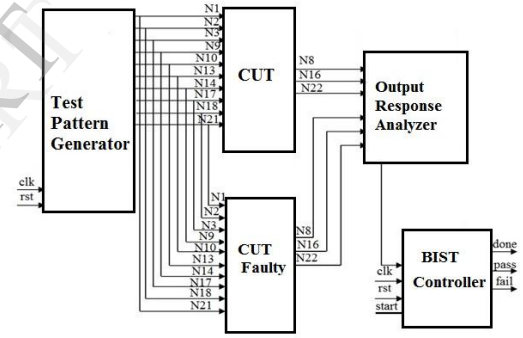

Figure 5. Block diagram of BIST

RESULTS

The verilog code is simulated in modelsim simulator and synthesized in Xilinx 10.2, and the BIST output is checked in Spartan 3E FPGA kit.

The output gives the ***pass*** signal if no fault is detected and ***fail*** signal if a fault is detected. When all the test patterns are applied it gives a ***done*** signal. The simulated output is shown in figure 6.

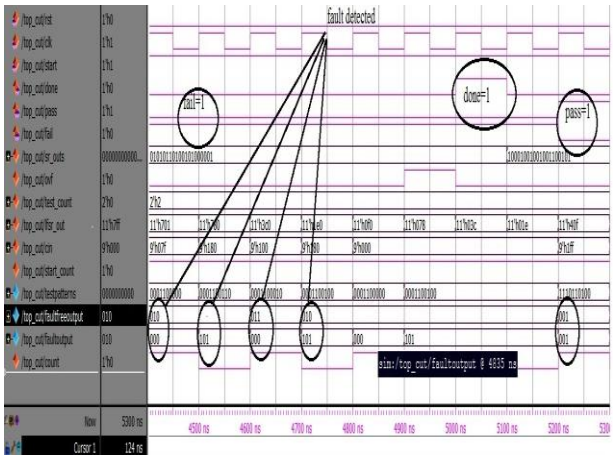

Figure 6. Simulated output of 3 weight BIST

This result is compared with non weighted LFSR based BIST. The simulated output is shown in figure 7

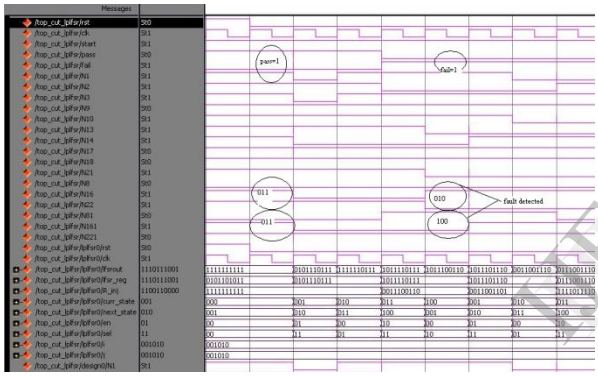

Figure 7. Simulated output of non weighted LFSR based BIST

The hardware design summary when synthesized in Xilinx is shown in figure 8

Project File:	tpg.ise	**Current State:**	Programming File Generated
Module Name:	bist15	**• Errors:**	No Errors
Target Device:	xc3s500e-5fg320	**• Warnings:**	23 Warnings
Product Version:	ISE 10.1 - Foundation Simulator	**• Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**• Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	0 (Timing Report)

tpg Partition Summary
No partition information was found.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	360	9,312	3%	
Number of 4 input LUTs	422	9,312	4%	
Logic Distribution				
Number of occupied Slices	409	4,656	8%	
Number of Slices containing only related logic	409	409	100%	
Number of Slices containing unrelated logic	0	409	0%	
Total Number of 4 input LUTs	475	9,312	5%	
Number used as logic				
Number used as a route-thru	53			
Number used as Shift registers	100			
Number of bonded IOBs	12	232	5%	
Number of RAMB16s	1	20	5%	
Number of BUFGMUXs	2	24	8%	
Number of BSCANs	1	1	100%	
Number of RPM macros	17			

Open Timing Constraints Report

Figure 8. Design summary of 3 weight BIST

The design is tested by dumping into Spartan 3E kit and output is analyzed using chipscope pro analyzer. This is used to monitor the output that is continuously changing where we cannot directly see on the kit. The output of chipscope analyzer after dumping on to the kit is shown in figure 9. The RTL schematic is shown in figure 10.

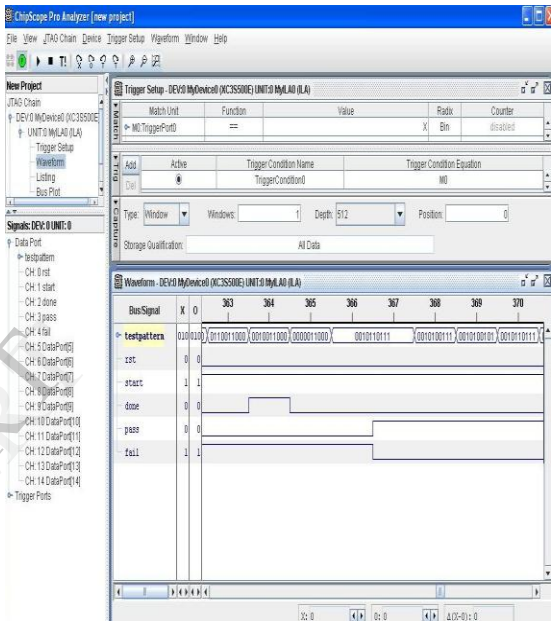

Figure 9. Chip scope analyzer output

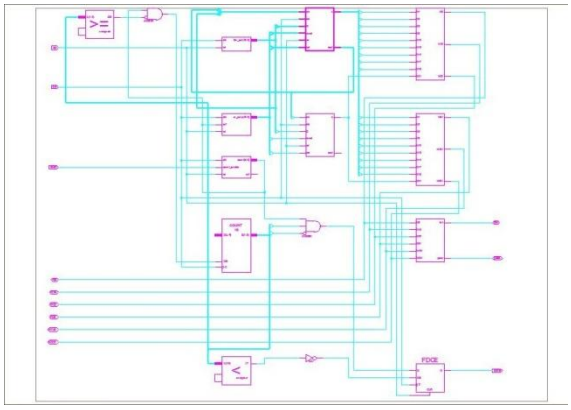

Figure 10: RTL schematic of 3weight BIST

The power is reduced when compared to non weighted LFSR based BIST and is shown in table 3

Constraint	3 weight BIST	LFSR based BIST
Power	0.081W	0.114W
Simulation time	5100ns (for this CUT)	More than 5100ns
Test vectors covered	6	1 at 39000ns
Power	81mW	124mW

Table 3. Comparison with LFSR based TPG

CONCLUSION

The test pattern generator with this 3-weight (0, 0.5, and 1) test-per-clock generation scheme efficiently generated weighted patterns without altering the structure of the adder, while at the same time no redesign of the accumulator is imposed, thus resulting in reduction in test application time. Comparison with the non weighted LFSR based BIST ,this weighted pattern generation with accumulator covers all the patterns and the test time is also less. The power consumption is less compared to other techniques. This technique has high fault coverage since all test vectors are applied in less time.

REFERENCES

1. P. Bardell, W. McAnney, and J. Savir, **Built-In Test for VLSI: Pseudorandom Techniques**. *New York: Wiley, 1987.*
2. A. Stroele, **A self test approach using accumulators as test pattern generators,** *in Proc. Int. Symp. Circuits Syst., 1995, pp. 2120–2123.*
3. I. Pomeranz and S. M. Reddy, 3 weight **pseudorandom test generation based on deterministic test set for combinational and sequential circuits,** *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 12, no. 7, Jul. 1993.*
4. K. Radecka, J. Rajski, and J. Tyszer, **Arithmetic built-in self-test for DSP cores**,*IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,Nov. 1997*
5. J. Rajski and J. Tyszer, **Arithmetic Built-In Self Test For Embedded Systems**. *Upper Saddle River, NJ: Prentice Hall PTR, 1998.*
6. S. Zhang, S. C. Seth, and B. B. Bhattacharya, **Efficient test compaction for pseudo-random testing,** *in Proc. 14th Asian Test Symp., 2005.*
7. J.Savir, **Distributed generation of weighted random patterns**, *IEEE Trans. Computer, vol. 48, no. 12, pp. 1364–1368, Dec. 1999*
8. I. Voyiatzis, D. Gizopoulos, and A. Paschalis, **Accumulator-based weighted pattern generation**, *presented at the IEEE Int. Line Test Symp., Saint Raphael, France, Jul. 2005.*