VLSI Testing Technique for BIST:Using Priority Based Algorithm

Ganesh L K M*, LopamudraPattanayak**
(*AsstProfessor,,School of Electronics Engineering,KIIT University, Bhubaneswar,Odisha, INDIA)
(**M. Tech. VLSI Design and Embedded System ,,School of Electronics Engineering,KIIT University, Bhubaneswar,Odisha, INDIA)

ABSTRACT

The paper presents a low test time BIST based on Priority Algorithm (PA) is applied for the 32-bit Carry Look-Ahead Adder. This method assigns priority to the test patterns based on faulty coverage and independent faulty detecting test patterns. Experiment conducted on Cadences' RTL Compiler Tool and Cadences' Encounter Tool demonstrate that proposed scheme gives better performance with large reduction in test time and power dissipation during testing.

Keywords- *BIST, LFSR, Priority Test Patterns, Test Time*

I. INTRODUCTION

Built-In Self-test (BIST) is the ability of an integrated circuit (IC) to examine its own functional health, in order to detect and report faults that may jeopardize the reliability of the application wherein it is deployed. The test time of a chip depends on the types of tests conducted [1]. These may include parametric tests (leakage, contact, voltage levels, etc.) applied at a slow speed, and vector tests (also called "functional tests" in the ATE environment) applied at high speed. The time of parametric tests is proportional to the number of pins since these tests must be applied to all active pins of the chip. The vector test time depends on the number of vectors and the clock rate. The total test time for digital chips ranges between 3 to 8 seconds. The vectors may not cover all possible functions and data patterns but must have a high coverage of modeled faults. The main driver is cost, since every device must be tested. Test time (and therefore cost) must be absolutely minimized. During Test mode, power consumption will double than normal mode [2].
This Priority Test Pattern method saves significant test time and power consumption by shortening the pattern sequence.

II. BIST STRUCTURE

Block Diagram of BIST is shown in figure.1. BI is enable pin for BIST operation and BO is output of BIST operation, based on BO only can say whether given CUT is working properly are not.When BI = 0, Test Pattern Generator and Ideal Response Block are in OFF state. MUX will accept Input and applied to CUT and outputs are taken at Output pin. When BI = 1, Test Pattern Generator and Ideal Response Block are in ON state. MUX will accept Test Pattern Generator output and applied to CUT and outputs are taken at BO pin. When BO is 1,indicates IC is working properly otherwise malfunction is there in IC.

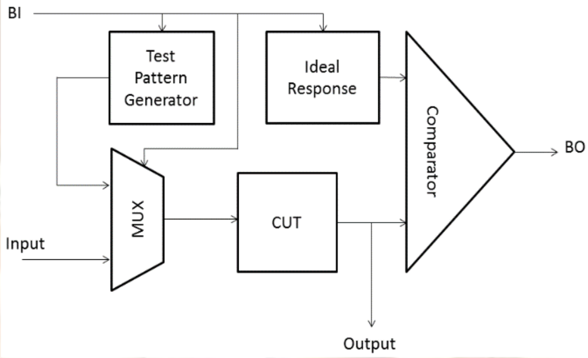

Fig 1. Block Diagram of BIST

III. PRIORITY ALGORITHM

If "n" is number of input present in digital circuit and "M = 2^n" input vector is present. For n input digital circuit, "F" is number of fault.

TABLE 1. LIST OF FAULT COVERED

Sl.No	Input Vector	No of Fault Covered
1	1st Input Vector	X_1
2	2nd Input Vector	X_2
..	……..	………
M	M Input Vector	X_M

Priorities are assigning as follows

- 1st Priority Y_1 = max (X1, X_2, X3,……..X_M)
- 2nd Priority Y_2 must select such that Y_2 must cover maximum remaining fault (F – Y_1). For example, 100 faults are present in digital circuits and first input pattern is covering maximum of 42 faults. Second input patterns must select such that it must cover maximum remaining fault (100 – 42 = 58).
- 3rd Priority Y_2 must select such that Y_2 must cover maximum remaining fault (F – Y_1 – Y_2) and soon.

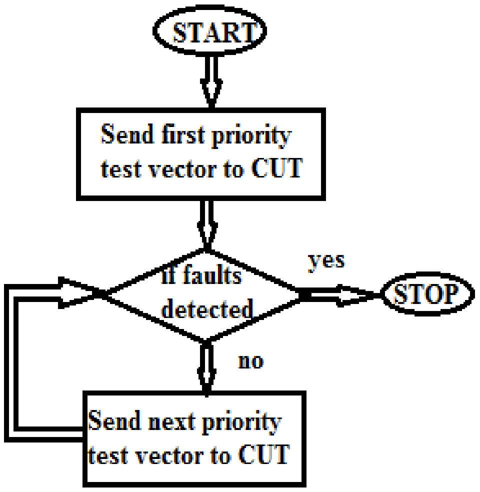

Fig 2. Flow chart of Priority Algorithm

For example CLA is taken as CUT

- CLA is used in most ALU designs.
- It is faster compared to ripple carry logic adders or full adders especially when adding a large number of bits.
- The Carry Look Ahead Adder is able to generate carries before the sum is produced using the propagate and generate logic to make addition much faster.

When BI is 1 and it will represented in a verilog code with test timing insertion and after that it will be simulated in Xilinx 9.2 version and the RTL diagram of 32 bit CLA is show in figure

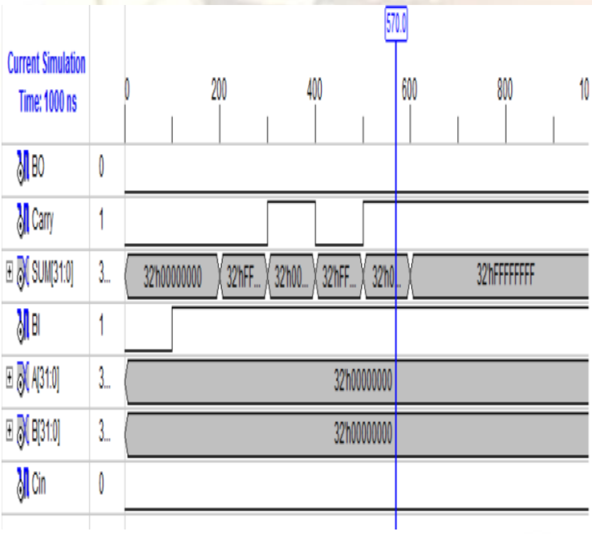

Fig 3. Test bench of 32 bit CLA

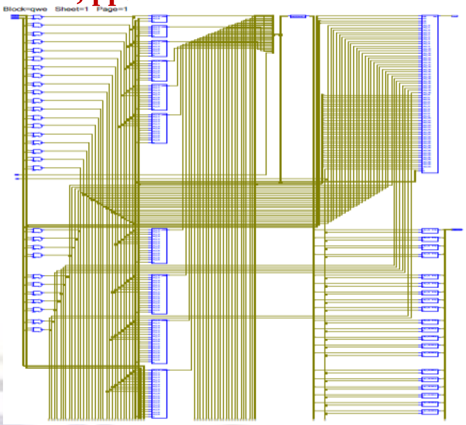

Fig 4.RTL diagram of 32 bit CLA

Four test patterns are sending to CLA (CUT) which is independent to each other for fault detecting.
Total no of faults is 258(64*2+33*2+32*2)

1. No of fault coverage =66/258=25%
(Input A=8'h00000000, Input B=8'hFFFFFFFF and Input Cin= 1'b0) stuck_at_1 at port A and port Cin can be detected
2. No of fault coverage =66/258=25%
(Input A=8'hFFFFFFFF, Input B=8'h00000000 and Input Cin= 1'b0) stuck_at_1 at port B and port Cin can be detected
3. No of fault coverage =66/258=25%
(Input A=8'h00000000, Input B= 8'hFFFFFFFF and Input Cin = 1'b1)
stuck_at_0 at port B and port Cin can be detected
4. No of fault coverage =66/258=25%
(Input A=8'hFFFFFFFF, Input B= 8'h00000000 and Input Cin=1'b1)
stuck_at_0 at port A and port Cin can be detected.

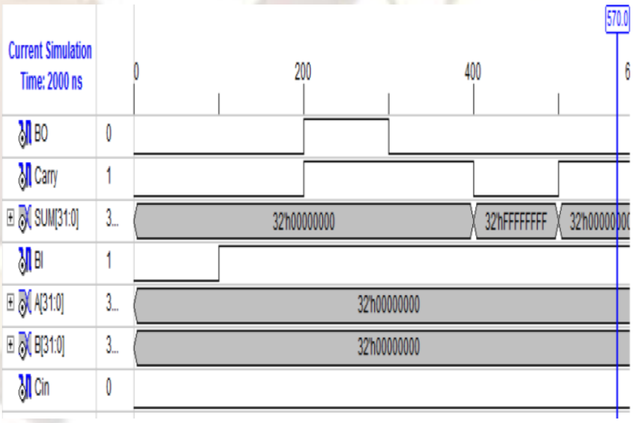

Fig 5. Test bench of 32bit CLA using BIST

IV. SIMULATION RESULTS

Verilog codes were used to simulate the test generation process performed using Cadences' RTL complier Tool and Layout analysis is performed using Cadences' Encounter Tool.

1. Leakage power, dynamic power and total power calculations in nW for 32-bit CLA, LFSR-BIST and PA-BIST are shown in figure 6,7 and 8 respectively.

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
thirtytwobit	241	257.453	1191632.147	1191889.600

Fig 6. Power Calculation of 32-bit CLA

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
using_lfsr	521	524.314	2421457.750	2421982.064

Fig 7. Power Calculation of LSFR-BIST

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
thirtytwobittesting	325	415.948	1824406.624	1824822.572

Fig 8. Power Calculation of PA-BIST

2. No of Gates used to design of 32-bit CLA, LSFR-BIST and PA-BIST are calculated and shown in figure 9, 10 and 11 respectively.

Gate	Instances	Area	Library
AOI21X1	16	212.890	tsmc18
DFFHQX1	3	159.667	tsmc18
DFFHQXL	30	1596.672	tsmc18
DFFTRX1	12	678.586	tsmc18
DFFTRXL	20	1130.976	tsmc18
DFFX1	1	56.549	tsmc18
INVX1	32	212.890	tsmc18
INVXL	47	312.682	tsmc18
OAI21XL	16	212.890	tsmc18
SDFFHQX1	42	2794.176	tsmc18
SDFFHQXL	1	66.528	tsmc18
SDFFXL	21	1397.088	tsmc18
total	241	8831.592	

Fig 9. Gates Calculation of 32-bit CLA

Gate	Instances	Area	Library
ADDHXL	1	36.590	tsmc18
AND2X2	2	26.611	tsmc18
AOI221X1	2	46.570	tsmc18
AOI22X1	57	948.024	tsmc18
AOI22XL	6	99.792	tsmc18
CLKINVX3	1	9.979	tsmc18
DFFHQX1	96	5109.350	tsmc18
DFFHQXL	1	53.222	tsmc18
DFFTRX1	22	1244.074	tsmc18
DFFTRXL	9	508.939	tsmc18
DFFX1	67	3788.770	tsmc18
INVX1	3	19.958	tsmc18
INVXL	25	166.320	tsmc18
NAND2BX1	7	93.139	tsmc18
NAND2X1	11	109.771	tsmc18
NAND2XL	90	898.128	tsmc18
NOR2X1	1	9.979	tsmc18
OAI21XL	25	332.640	tsmc18
OAI222XL	2	53.222	tsmc18
OAI2BB1XL	29	482.328	tsmc18
SDFFHQX1	1	66.528	tsmc18
SDFFXL	63	4191.264	tsmc18
total	521	18295.200	

Fig 10. Gates Calculation of LFSR-BIST

Gate	Instances	Area	Library
AOI21X1	1	13.306	tsmc18
DFFHQX1	33	1756.339	tsmc18
DFFHQXL	33	1756.339	tsmc18
DFFTRX1	32	1809.562	tsmc18
DFFTRXL	34	1922.659	tsmc18
DFFX1	32	1809.562	tsmc18
INVX1	1	6.653	tsmc18
INVXL	31	206.237	tsmc18
NAND2BX1	32	425.779	tsmc18
NAND2BXL	1	13.306	tsmc18
OAI21XL	31	412.474	tsmc18
SDFFHQX1	1	66.528	tsmc18
SDFFXL	63	4191.264	tsmc18
total	325	14390.006	

Fig11.Gates Calculation of PA-BIST

3. Area of 32-bit CLA, LSFR-BIST and PA-BIST are calculated and shown in figure12,13 and 14 respectively.

Type	Instances	Area	Area %
sequential	130	7880.242	89.2
inverter	79	525.571	6.0
logic	32	425.779	4.8
total	241	8831.592	100.0

Fig 12. Area Calculation of 32-bit CLA

Type	Instances	Area	Area %
sequential	130	7880.242	89.2
inverter	79	525.571	6.0
logic	32	425.779	4.8
total	241	8831.592	100.0

Fig 13. Area Calculation of LFSR-BIST

Type	Instances	Area	Area %
sequential	228	13312.253	92.5
inverter	32	212.890	1.5
logic	65	864.864	6.0
total	325	14390.006	100.0

Fig 14. Area Calculation of PA-BIST

4. Layouts of LFSR-BIST and PA-BIST without negative slacks are shown in figure 15 and 16 respectively.

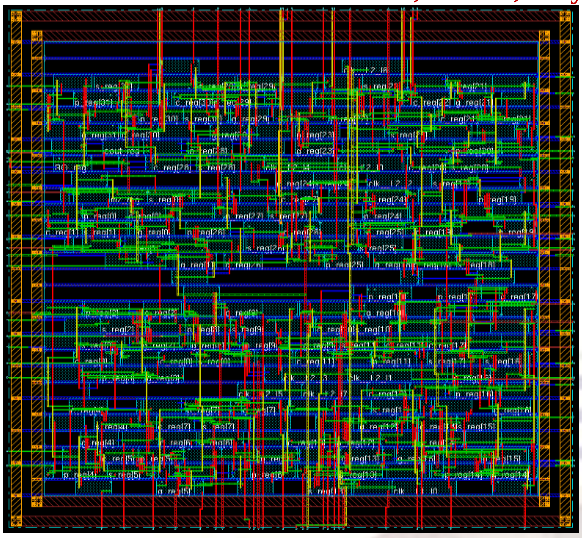

Fig 15. Layout of LFSR-BIST

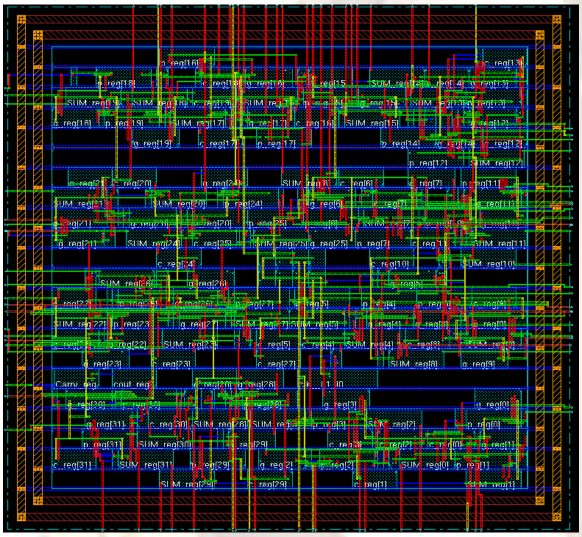

Fig16. Layout of PA-BIST

LFSR-BIST and PA-BIST with all parameters are compared in Table-II

Table 2. Comparison of LSFR-BIST and PA-BIST

Parameters	CLA	Using LFSR	Percentage	Using PA-BIST	Percentage
AREA (in µm)	8831	18295.2	51%	14390.006	38%
Power (in nW)	1191632	2421982	50%	1824406	34%
No of Gates	241	521	53%	325	34%

V. CONCLUSION

As showed in table 2, PA-BIST can highly reduce the test time, during BIST application, that is, all the total power, area and the no of gates are highly reduced. Experimental results based on Cadences' RTL Complier tool and Encounter tool for BIST applications show that about 51% to 38% reduction in area, 50% to 34% reduction in the power and 53% to 34% reduction in no of gates used and BIST achieved without losing the stuck-at fault coverage.

A comparison of reduction of power consumption between LFSR-BIST and PA-BIST was reported, to demonstrate that PA-BIST is much more efficient in the reduction of peak power consumption when the pattern is applied to 32-bit Carry Look Ahead Adder.

REFERENCE

1. Michael L. Bushnell,Vishwani D. Agrawal "*Essentials of ElectronicTesting fordigital, memory and mixed-signal vlsi circuits*"Kluwer Academic Publishers, pp.11-12.
2. Y Zorian, "*A Distributed BIST control scheme for Complex VLSIdevices.*". IEEE PZSITeslSymp. 1993, pp.4-9.
3. Michael L. Bushnell,Vishwani D. Agrawal "*Essentials of ElectronicTesting fordigital, memory and mixed-signal vlsi circuits*"Kluwer Academic Publishers, pp.496-498.