ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

Recursive Pseudo-Exhaustive Two-Pattern Generator

PRIYANSHU PANDEY[1], VINOD KAPSE[2]
[1]M.TECH IV SEM, HOD[2]
priyanshupandey@gmail.com
Department of Electronics & Communication
Gyan Ganga Institute of Technology & Sciences.

Abstract

Pseudo-exhaustive pattern generators for built-in self-test (BIST) provide high fault coverage of detectable combinational faults with much fewer test vectors than exhaustive generation. In (n, k)-adjacent bit pseudo-exhaustive test sets, all 2k binary combinations appear to all adjacent k-bit groups of inputs. With recursive pseudoexhaustive generation, all (n, k)-adjacent bit pseudoexhaustive tests are generated for k, n and more than one modules can be pseudo-exhaustively tested in parallel. In order to detect sequential (e.g., stuck-open) faults that occur into current CMOS circuits, two-pattern tests are exercised. Also, delay testing, commonly used to assure correct circuit operation at clock speed requires two-pattern tests. In this paper a pseudoexhaustive two-pattern generator is presented, that recursively generates all two-pattern (n, k)-adjacent bit pseudoexhaustive tests for all k, n. To the best of our knowledge, this is the first time in the open literature that the subject of recursive pseudoexhaustive two-pattern testing is being dealt with. A software-based implementation with no hardware overhead is also presented.

Keywords: **BIT (built-in-test), generic pseudoexhaustive test & recursive pseudoexhaustive test, two pattern generation.**

I. Introduction

Built-in self test (BIST) techniques add circuitry that allows a chip to test itself. The added circuitry, when activated, takes control, drives the inputs, observes the outputs and reports whether the result is correct. BIST is often used on portions of the circuit that cannot be easily tested using external testing. Furthermore, with BIST at speed testing can be achieved and the quality of the delivered ICs is greatly increased. BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. BIST is also the solution to the testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. BIST employs on-chip test generation and response verification. Advantages of implementing BIST include: 1) low cost of test, 2) better fault coverage, 3) minimal test time.

BIST pattern generator classified into one-pattern generator and two-pattern generator. One-pattern generator detects the stuck-at faults in the combinational circuits. The two-pattern test is used to detect sequential faults such as stuck-open faults in CMOS circuits. It is to test the circuits at high speeds and also testing for the correct temporal behavior, it is known as delay testing.

A generic pseudoexhaustive two-pattern generator generates an (n, k) - pseudoexhaustive two-pattern test for any value of k, by enabling an input signal P[k], $1 \leq k \leq n$. The generic pseudoexhaustive two-pattern generator can be generalized into progressive two-pattern generator that generates all (n, k) pseudoexhaustive two-pattern test vectors for all values of k. This technique is called as recursive pseudo exhaustive two-pattern generation.

In recursive pseudoexhaustive testing, (n, k)-PETS are generated for all k= 1,2,3,…..,n. By the utilization of an array of XOR gates and binary counter we can recursively generate all (n, k) pseudoexhaustive test patterns for $k \leq n$ in minimal time.

II. Test Pattern Generation

In this paper two pattern generators like recursive pseudo exhaustive two pattern generator (RPET) are implemented to generate two pattern tests for modules having different cone sizes.

A. RPET

Recursive pseudoexhaustive two-pattern generator is also having the architecture similar to generic pseudoexhaustive two pattern generator. Additionally it have m = [log27] = 3-stage counter and 3- to- 7 decoder added to generic pseudoexhaustive two-pattern generator's architecture. When the two-pattern (n, n)-PET is complete, the recursive pseudoexhaustive test is also complete.

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

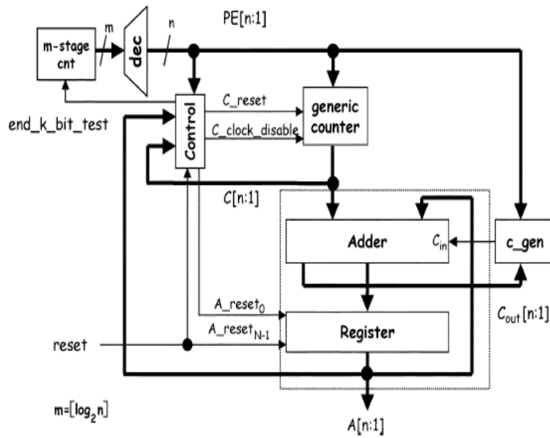

Fig. Recursive pseudo-exhaustive two-pattern generator

1)Generic Counter

The input of the 7-stage generic counter are counter reset (C_reset), 7 bit signal PE[7:1] and counter disable(C_clk_disable).If the signals PE[1] is enabled, then the generic counter operates as an 7-stage binary counter. When PE [4] is enabled, then the generic counter operates as two 3-bit subcounter and a 1-bit subcounter from LSB. Therefore, the generic counter generates all $2^{k-1} \times (2^{k-1} - 1)$ combinations to all groups of k-1 adjacent bits. When the signal C_reset is enabled, the generic counter counts from the initial value. The signal C_clk_disable is used to keep the counter idle. The operation of the generic counter is shown in the table.

TABLE I
GENERIC (7, K)- PSEUDOEXHAUSTIVE GENERATOR

PE [7:1]	(n, k)	Operates as…
0000001	(7, 7)	(xxxxxxx)
0000010	(7, 1)	(x) (x) (x) (x) (x) (x) (x)
0000100	(7, 2)	(x) (xx) (xx) (xx)
0001000	(7, 3)	(x) (xxx) (xxx)
0010000	(7, 4)	(xxx) (xxxx)
0100000	(7, 5)	(xx) (xxxxx)
1000000	(7, 6)	(x) (xxxxxx)

TABLE II
OPERATION OF A 7-STAGE GENERIC COUNTER

PE [7:1]	Operates as…	In each clock cycle increased by…
0000001	1x7 stage counter	0000001
0000010	7x1 stage counter	1111111
0000100	1x1 stage counter + 3x2 stage counter	1010101
0001000	1x1 stage counter + 2x3 stage counter	1001001
0010000	1x3 stage counter + 1x4 stage counter	0010001
0100000	1x2 stage counter + 1x5 stage counter	0100001
1000000	1x1 stage counter + 1x6 stage counter	1000001

2) Carry Generator

The C_gen is used to give the Cin input for the adder. The inputs of C_gen module are PE[7:1] and Cout[7:1]. If the signal PE[4] is enabled, the Cout[4] is given as a Cin. Based on the signal PE[7:1] the value of Cin changes.

3) Control

The control module is used to determine that a k–stage two-pattern test is generated at the k low-order bits of the generator. The input signals of the control module are reset, ACC[7:1], C[7:1], PE[7:3], and generates the signals C_clk_disable, C_reset, A_reset0 and end_k_bit_test.

The control logic controls the entire architecture of the pseudoexhaustive pattern generator. The purpose of control logic is to assure that a k-stage two-pattern test is generated at the k low-order stages of the

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

generator. The operation of the control logic is explained in the Fig.5. In a C-like notation, table III.

Phase	Step	
		{ int k = 2^k, A = K - 1
		do {
	1	C = 0 ;
1	2	do {
	3	C++; A = Acc (A, C, K);
	4	} while (C! = K-3);
	5	} while (A! = K-1);
2	6	C++;
	1	do {A = Acc (A, C, K); }
	2	while (A! = K-1);
	2	C = 0;
	1	do {
	2	C++;
3	3	A = 0;
	4	A = Acc (A, C, K);
	5	} while (A! = K-1);
	6	}
		Int Acc (int A, C, K) { return (A + C < K ? A + C : (A + C) % K + 1); }

Figure. Algorithm for control logic.

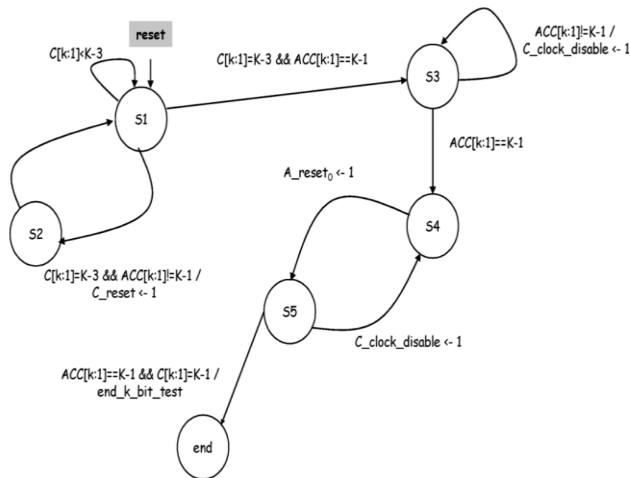

Figure. State diagram

4) 1's Complement Adder

The inputs of adder are cin, A[7:1], C[7:1] and its outputs are C_out[7:1], A[7:1]. It consists of 7 full adders, the carry output of the full adders are propagated to the next full adders as carry input. If the value of k is considered to be 3 then the accumulator operates as two 3-stage subaccumulator and one 1-stage accumulator. The carry output of each sub accumulator is given to the carry input of next sub accumulator. If there is any carry in the 3rd bit then it is added to the lowest order bit of the adder output. It performs one bit addition and output saves in internal register. For n-stage operation carry is generated by carry generator. The adder gets input from the counter. Cout is given to the carry gen. when the clock signal is enabled adder performs one bit addition. The internal register is capable of storing output of the adder. The adder is reset by reset A signal and the register is also reset by separate reset signal. The input clock signal is disabled the adder remains idle. The operation of adder is shown in table.

Table: operation of 1's complement adder

Previous A[7:1]	C[7:1]	Present A[7:1]	C_{out}[7:1]	k
1111111	1111111	1111111	111111**1**	1
0101010	0101010	1010101	01010**1**0	2
1011011	0110110	0010001	1111**1**10	3
0011001	1001100	1100110	001**1**000	4
0111001	0111101	1110111	01**1**1001	5

TABLE III
TWO-PATTERN TEST GENERATED BY TPG(3)

Phase #	Cycle #	C	A	Phase #	Cycle #	C	A	Phase #	Cycle #	C	A
	0		111	1	19	100	110	2	38	110	100
1	1	001	001	1	20	101	100	2	39	110	011
1	2	010	011	1	21	001	101	2	40	110	010
1	3	011	110	1	22	010	111	2	41	110	001
1	4	100	011	1	23	011	011	2	42	110	111
1	5	101	001	1	24	100	111	3	43	001	000
1	6	001	010	1	25	101	101	3	44	001	001
1	7	010	100	1	26	001	110	3	45	010	000
1	8	011	111	1	27	010	001	3	46	010	010
1	9	100	100	1	28	011	100	3	47	011	000
1	10	101	010	1	29	100	001	3	48	011	011
1	11	001	011	1	30	101	110	3	49	100	000
1	12	010	101	1	31	001	111	3	50	100	100
1	13	011	001	1	32	010	010	3	51	101	000
1	14	100	101	1	33	011	101	3	52	101	101
1	15	101	011	1	34	100	010	3	53	110	000
1	16	001	100	1	35	101	111	3	54	110	110
1	17	010	110	2	36	110	110	3	55	111	000
1	18	011	010	2	37	110	101	3	56	111	111

All Rights Reserved © 2012 IJARCET

TableIV
Output of RPET

m[3:1]	PE[7:1]	C[7:1]	A[7:1]
011	0001000	1001001 0010010 1011011 0100100 : 1111111	1111111 1001001 1011011 0110110 1011011 : 1111111
100	0010000	0010001 0100010 0110011 1000100 : 1111111	0010001 0110011 1100110 0101010 : 1111111
101	0100000	0100001 1000010 1100011 0000100 : 1111111	0100001 1100011 1000110 1001010 : 1111111
110	1000000	1000001 0000010 1000011 0000100 : 1111111	1000001 1000011 0000110 0001010 : 1111111
111	0000001	0000001 0000010 0000011 0000100 : 1111111	0000001 0000011 0000110 0001010 : 1111111

III. Circuit Under test

The output from the two pattern test generator is applied to the CUT. In this paper two circuits, Wallace tree multiplier and cryptographic circuit are tested in parallel to increase the speed of the BIST.

A) 4 bit Wallace tree multiplier

A Wallace tree multiplier is an efficient hardware implementation of a digital circuit that multiplies two binary values. The 4 bit Wallace tree multiplier is shown in fig.

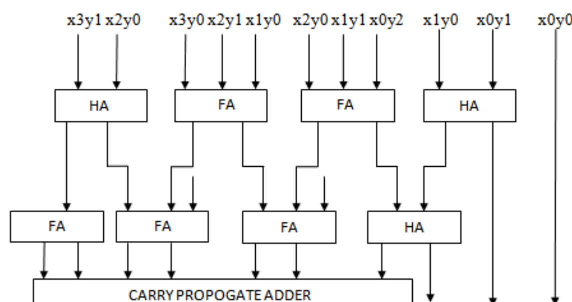

Fig 5.3.1: 7-bit Wallace tree multiplier

Wallace Tree: 2 carry-save levels, 5 FA, 3 HA

The Wallace tree has three steps:

- Multiply each bit of one of the arguments, by each bit of the other, yielding results. Depending on position of the multiplied bits, the wires carry different weights.
- Reduce the number of partial products to two by layers of full and half adders.
- Group the wires in two numbers, and add them with a conventional adder.

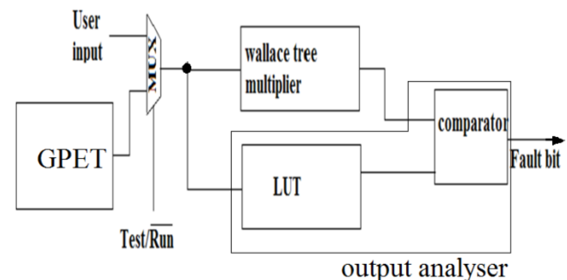

Fig: Proposed Block diagram of BIST

IV. OUTPUT ANALYSER

BIST techniques usually combine a built-in binary pattern generator with circuitry for compressing the corresponding response data produced by the circuit under test. The compressed form of the response data is compared with a known fault-free response.

V. RESULTS AND DISCUSSIONS

BIST plays a vital role in modern VLSI technology. The BIST should occupy less area for compact design of digital circuit. When compared to the results of [4], [5] the test pattern generator proposed in [3] requires fewer hardware to implement. Based on the technique used in [3] a test pattern is generated. The comparison of the hardware overhead is shown in table

All Rights Reserved © 2012 IJARCET

Table 5: comparison of hardware overhead in gates

Scheme	Gate Equivalents	No. of gates When n = 7
GPET	**12 x n**	**84**
RPET	**15 x n + 8 x m**	**129**
GPET [4]	**16 x n**	**112**
RPET[4]	**18 x n + 8 x m**	**150**
[5]	**7 x n + XOR gates + 3 x n +8 x m**	**202**
[6]	**7 x n + XOR gates + 3 x n +8 x m**	**229**

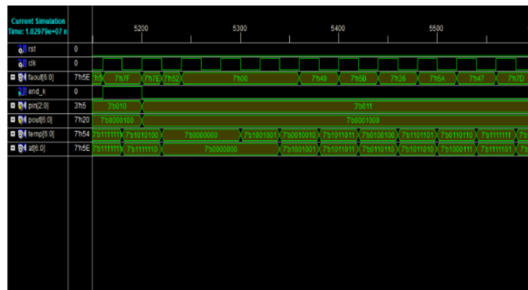

Fig: Waveform of RPET

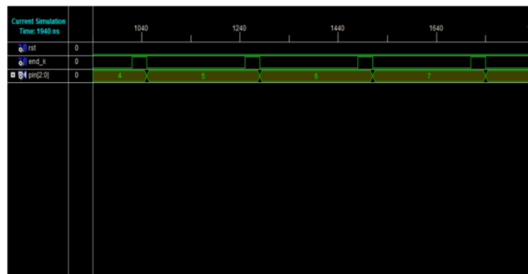

Fig: Waveform of n stage counter

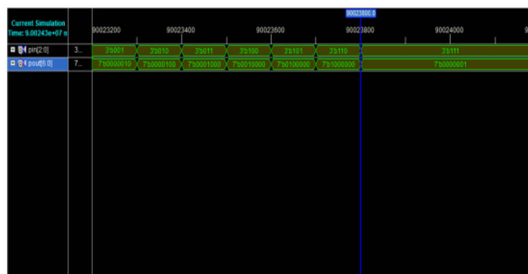

Fig : Waveform of decoder

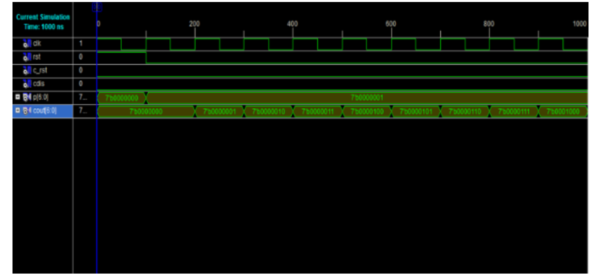

Fig: Waveform of the generic counter

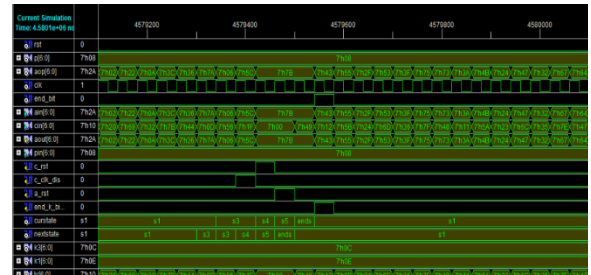

Fig: Waveform of the counter

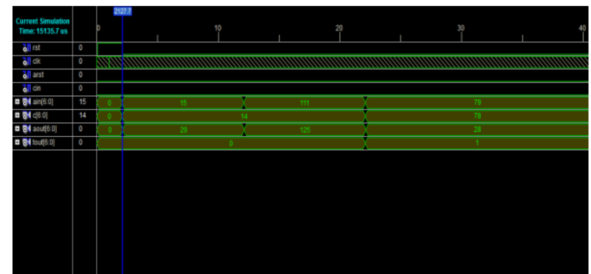

Fig: Waveform of the Adder

VI. CONCLUSION

A GPET and RPET based BIST is designed in this paper. Two circuits having different cone size are tested at a time using this BIST circuit. The proposed BIST is synthesized using Xilinx tool.

VII. REFERENCES

[1] Parag K. Lala, An introduction to logic circuit testing, Morgan&Claypool publishers.

[2] M. Abramovici, M. Breuer, and A. Freidman, Digital Systems Testing and Testable Design. New York: Computer Science Press, 1990.

[3] Ioannis Voyiatzis, Dimitris Gizopoulos and Antonis Paschalis, "Recursive Pseudo-Exhaustive Two Pattern Generation," IEEE

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

trans. Very Large Scale Integration (VLSI) sys, vol. 18, no. 1, jan 2010.

[4] J. Rajski and J. Tyszer, “Recursive pseudoexhaustive test pattern generation,” IEEE Trans. Comput., vol. 42, no. 12, pp. 1517–1521, Dec. 1993.

[5] P. Dasgupta, S. Chattopadhyay, P. P. Chaudhuri, and I. Sengupta, “Cellular automata-based recursive pseudo-exhaustive test pattern generation,” IEEE Trans. Comput., vol. 50, no. 2, pp. 177–185, Feb. 2001.

[6] R. Wadsack, “Fault modeling and logic simulation of CMOS and nMOS integrated circuits,” Bell Syst. Techn. J., vol. 57, pp. 1449-1474, May–Jun. 1978.

[7] C. Chen and S. Gupta, “BIST test pattern generators for two-pattern testing-theory and design algorithms,” IEEE Trans Comput., vol. 45, no. 3, pp. 257–269, Mar. 1996.

[8] Chih-Ang Chen, “Efficient BIST TPG Design and Test Set Compaction via Input Reduction,” IEEE trans. on CADICS, vol. 17, NO. 8, AUG 1998.

[9] Dong Xiang, “A Reconfigurable Scan Architecture with Weighted Scan-Enable Signals for Deterministic BIST,” IEEE Trans. On CADICS, Vol. 27, No. 6, Jun 2008.

[10] K. Yang, K.T. Cheng, and L. C. Wang, “TranGen: A SAT-based ATPG for path-oriented transition faults,” in Proc. ASP-DAC, 2004, pp. 92–97.

[11] C. Chen and S. Gupta, “BIST test pattern generators for two-pattern testing-theory and design algorithms,” IEEE Trans. Comput., vol. 45, no. 3, pp. 257–269, Mar. 1996.

All Rights Reserved © 2012 IJARCET