International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-676-681 ISSN: 2249-6645

# A BIST Circuit for Fault Detection Using Recursive Pseudo-Exhaustive Two Pattern Generator

**K. Nivitha[1], Anita Titus[2]**
[1] ME-VLSI Design
[2] Dept of ECE
Easwari Engineering College, Chennai-89

***Abstract* – A Built-in self-test (BIST) technique based on pseudo-exhaustive testing is proposed in this paper. Two pattern test generator is used to provide high fault coverage. Testing for delay and sequential faults requires two-pattern tests. In two-pattern testing all possible combinations of the test vectors are applied to the circuit under test. In this paper a pseudo exhaustive two-pattern generator with fewer hardware that generates two-pattern (7,k)-adjacent bit pseudo exhaustive tests for any k<7 is used. Two circuits like Wallace tree multiplier and cryptographic circuit based on RSA algorithm are tested in parallel. The main advantage of this BIST is that the circuits have different cone sizes are tested at a time. This increases the speed of the BIST.**

**Index Terms – Built-in self-test (BIST), Pseudo-exhaustive two pattern testing, Test pattern generation.**

## I. INTRODUCTION

The task of testing a VLSI chip to guarantee its functionality is extremely complex and often very time consuming. A widely accepted approach to deal with the testing problem at the chip level is to incorporate built-in self-test (BIST) capability inside a chip. [1]

BIST is more suitable for periodic testing which is carried out periodically in the system itself. This can be either on-line or off-line. BIST is comprised of TPG, response analyzer and testing logic. In BIST scheme, design of TPG is very critical. Conventionally, Linear Feedback Shift Registers (LFSR) is commonly used TPG in BIST structure. Though it is suitable for BIST, it has many disadvantages. It is a random pattern generator, complex in design, it generates unwanted patterns which increases the switching activity of the CUT leads to increase in testing period. For these reasons exhaustive and pseudo exhaustive testing are preferred. In exhaustive testing number of test vector are more which is minimized in pseudo exhaustive testing [2].

BIST pattern generators are commonly discerned into one-pattern and two-pattern generators. One-pattern generators mainly detect combinational faults. It has been proved that many failure mechanisms in CMOS circuits cannot be modeled by the stuck-at faults. Furthermore, the digital circuits should operate at their highest possible speeds to increase their performance. For correct behavior of the circuits, delay faults are detected by two pattern test generators [6].

In BIST, the test pattern generation and the output response evaluation are done on chip itself so the hardware used for designing BIST should be minimized [1].

In [4] a method is proposed for generating universal pseudo-exhaustive test. Assuming n as input of TPG and m as input of the circuit it requires 7n registers, 8m counters, 3n multipliers and XOR gates. In [5] a method is proposed that uses Cellular Automata to generate recursive pseudo-exhaustive test patterns. The hardware requirement of the proposed scheme is 15 to 50 percent less XOR gates compared to the existing Recursive Pseudo-exhaustive Test Pattern Generation and test string length increases from 14 to 31. In [8] the hardware required to design recursive pseudo exhaustive generator is 21n plus 24m gates. Compared to all other papers, the recursive pseudo exhaustive two pattern generator (RPET) and generic pseudo exhaustive two pattern generator (GPET) in [3] requires minimum hardware utilization. RPET requires 18n plus 8m gates and GPET requires 16n gates. Both the TPGs are designed and GPET is used for the BIST designed in this paper. It generates all the tests for any cone size (k). By using this GPET more than a circuit is tested in parallel which increases the speed of the BIST. In this paper two circuits like 4×4 Wallace tree multiplier and 7-bit cryptographic circuit are tested in parallel. The outputs from circuits are compared with the stored values and faults are determined. The fault coverage of proposed BIST is more when compared to existing BIST.

## II.TEST PATTERN GENERATOR

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

### A. GPET

The generic pseudo-exhaustive two-pattern generator is presented in Fig. 1. It consists of a generic counter, 1's complement adder, a controller and a carry generator (C_gen). The 7-bit pattern PE [7:1] is given as an input to the controller, generic counter and C_gen and the output A [7:1] is taken from the Accumulator.

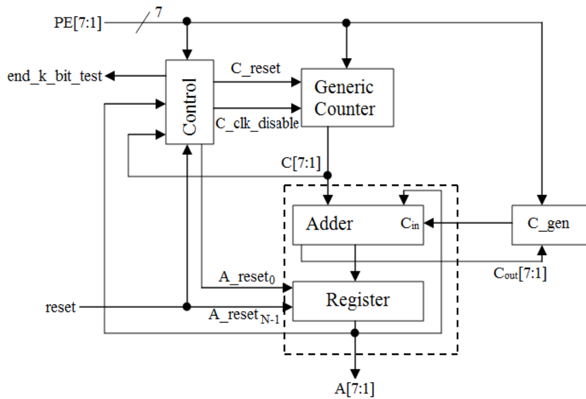


Fig.1: Generic pseudo exhaustive two pattern generator

The operation of the GPET varies based on the PE value. If the value PE[4] is enabled, a (7,3)-pseudo-exhaustive test set is generated and a 3-bit exhaustive test set is applied to the 2-bit groups A[3:1] ,A[6:4] , and a single bit test set is generated at A[7].

### 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 1.

Table 1: operation of the generic counter

| PE[7:1] | Operates as… | In each clock increased by… |
|---|---|---|
| 0000001 | 1×7-stage counter | 0000001 |
| 0000010 | 7×1-stage counters | 1111111 |
| 0000100 | 1×1-stage counter + 3×2-stage counters | 1010101 |
| 0001000 | 1×1-stage counter + 2×3-stage counters | 1001001 |
| 0010000 | 1×3-stage counter + 1×4-stage counter | 0010001 |
| 0100000 | 1×2-stage counter + 1×5-stage counter | 0100001 |
| 1000000 | 1×1-stage counter + 1×6-stage counter | 1000001 |

### 2) Carry generator

The C_gen is used to give the $C_{in}$ input for the adder. The inputs of C_gen module are PE[7:1] and $C_{out}$[7:1]. If the signal PE[4] is enabled, the $C_{out}$[4] is given as a $C_{in}$. Based on the signal PE[7:1] the value of $C_{in}$ 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, $A_reset_0$ and end_k_bit_test.

### 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. The operation of adder is shown in table 2.

Table 2: 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 | 1111111 | 1 |
| 0101010 | 0101010 | 1010101 | 0101010 | 2 |
| 1011011 | 0110110 | 0010001 | 1111110 | 3 |
| 0011001 | 1001100 | 1100110 | 0011000 | 4 |
| 0111001 | 0111101 | 1110111 | 0111001 | 5 |

### 5) TPG Algorithm

This algorithm consists of three steps which generates all 7-bit two pattern tests within $2^k \times (2^k-1)$ clock cycles. For simplification $2^k$ is considered as K.

- The counter counts from 1 to K-3 and then it is reset; this is repeated until the outputs of the counter are equal to K-3 and the outputs of the accumulator are equal to K-1.
- The counter is incremented to K-2 and the accumulator repeatedly accumulates K-2 until its output is equal to K-1.
- All transitions to and from zero are generated, by resetting the accumulator and incrementing the counter every second clock cycle.

The output of the generic pseudo exhaustive two pattern generator (GPET) for PE[4] is shown in table 3.

Table 3: Output of the GPET

| C | A | C | A | C | A |
|---|---|---|---|---|---|
| STEP 1 | | 1101 | 0100 | 0110 | 0010 |
| | 1111 | 101 | 100 | 110 | 010 |
| 10010 | 111 | 1001 | 1101 | 0110 | 1001 |
| 01 | 1001 | 001 | 101 | 110 | 001 |
| 00100 | 001 | 0010 | 1111 | 0110 | 1111 |
| 10 | 1011 | 010 | 111 | 110 | 111 |
| 10110 | 011 | 1011 | 1011 | STEP 3 | |
| 11 | 0110 | 011 | 011 | 1001 | 0000 |
| 01001 | 110 | 0100 | 1111 | 001 | 000 |
| 00 | 1011 | 100 | 111 | 1001 | 1001 |
| 11011 | 011 | 1101 | 1101 | 001 | 001 |
| 01 | 1001 | 101 | 101 | 0010 | 0000 |
| 10010 | 001 | 1001 | 0110 | 010 | 000 |
| 01 | 0010 | 001 | 110 | 0010 | 0010 |
| 00100 | 010 | 0010 | 1001 | 010 | 010 |
| 10 | 0100 | 010 | 001 | 1011 | 0000 |
| 10110 | 100 | 1011 | 0100 | 011 | 000 |
| 11 | 1111 | 011 | 100 | 1011 | 1011 |
| 01001 | 111 | 0100 | 1001 | 011 | 011 |
| 00 | 0100 | 100 | 001 | 0100 | 0000 |
| 11011 | 100 | 1101 | 0110 | 100 | 000 |
| 01 | 0010 | 101 | 110 | 0100 | 0100 |
| 10010 | 010 | 1001 | 1111 | 100 | 100 |
| 01 | 1011 | 001 | 111 | 1101 | 0000 |
| 00100 | 011 | 0010 | 0010 | 101 | 000 |
| 10 | 1101 | 010 | 010 | 1101 | 1101 |
| 10110 | 101 | 1011 | 1101 | 101 | 101 |
| 11 | 1001 | 011 | 101 | 0110 | 0000 |
| 01001 | 001 | 0100 | 0010 | 110 | 000 |
| 00 | 1101 | 100 | 010 | 0110 | 0110 |
| 11011 | 101 | 1101 | 1111 | 110 | 110 |
| 01 | 1011 | 101 | 111 | 1111 | 0000 |
| 10010 | 011 | STEP 2 | | 111 | 000 |
| 01 | 0100 | 0110 | 0110 | 1111 | 1111 |
| 00100 | 100 | 110 | 110 | 111 | 111 |
| 10 | 0110 | 0110 | 1101 | | |
| 10110 | 110 | 110 | 101 | | |
| 11 | 0010 | 0110 | 0100 | | |
| 01001 | 010 | 110 | 100 | | |
| 00 | 0110 | 0110 | 1011 | | |
| | 110 | 110 | 011 | | |

## B) RPET

The recursive pseudo exhaustive two pattern generator is shown in fig 2. Additionally It consists of an m = [$\log_2 7$] = 3-stage counter driving the inputs of a 3-to-7 decoder then GPET.

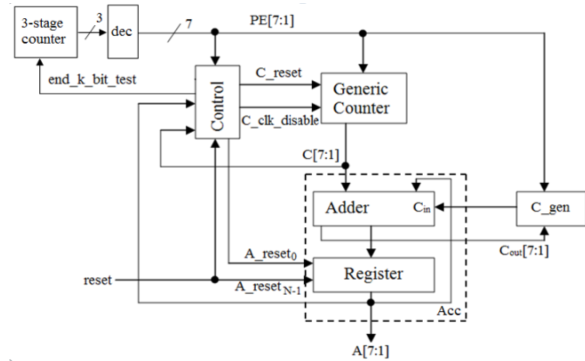


Fig2:Recursive pseudo exhaustive two pattern generator

It recursively enable PE[k] for all values of k, $2 \le k \le 7$. The output of the recursive pseudo exhaustive two pattern generator (RPET) is shown in table 4.

Table 4: Output of the RPET

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

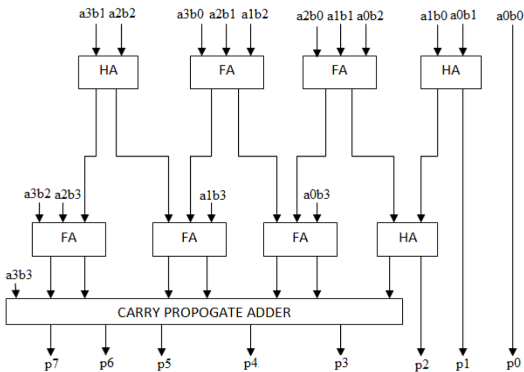


Fig 3: 4-bit Wallace tree multiplier

The Wallace tree has three steps:

- Multiply each bit of one of the arguments, by each bit of the other, yielding 16 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.

### B) 7-bit cryptographic circuit

It is used for secure transmission of private information over insecure channels. There are two types of cryptographic methods 1) Symmetric─ same key for encryption and decryption 2) Asymmetric─ Mathematically related key pairs for encryption and decryption i.e., Public and private keys.

Asymmetric circuits are more secure than symmetric circuits. In this paper RSA algorithm based cryptographic circuit is implemented. The RSA algorithm involves three steps: key generation, encryption and decryption. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. In run mode these circuits perform normal operation and in testing mode it performs fault detection. The block diagram of the BIST is shown in fig 4.

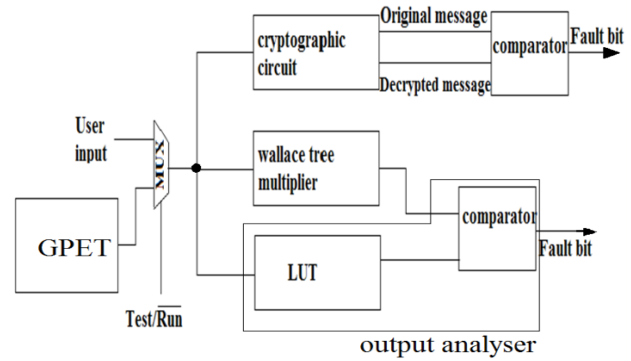


Fig: 4 Proposed Block diagram of BIST

## 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 5.

Table 5: comparison of hardware overhead in gates

| scheme | Gate equivalents | No. of gates when n=7 |
|---|---|---|
| GPET | 16×n | 113 |
| RPET | 18×n + 8×m | 148 |
| [4] | 7×n + XOR gates + 3×n +8×m | 202 |
| [5] | 7×n + XOR gates + 3×n +8×m | 229 |

The RSA based cryptographic circuit is tested with counter based one pattern BIST and proposed BIST. The fault coverage of the circuits is determined. Fault detected using Normal BIST is 25. Fault detected using Proposed BIST is 59 in the same circuit. Total number of faults in this circuit is 60. So the fault coverage of the proposed BIST is 98%. When compared to [8], [9], [10] proposed BIST detects maximum number of faults.

Fig: 5 Waveform of GPET

Fig: 6 Waveform of RPET

Fig 7: Waveform of one pattern BIST

Fig 8: Waveform of proposed BIST

Fig 9: Waveform of the BIST testing two circuits

## 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 simulated using the tool model sim and synthesized using Quartus 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 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.