



LATENCY AND DENSITY OPTIMIZED TPG FOR EFFICIENT BIST IMPLEMENTATION

¹NAGULAPALLI M N VENKATA DURGA MANVITHA, ²K.CHANDRA SEKHAR

¹M. tech, Dept. of ECE, Eluru College of Engineering and Technology, ELURU, AP

²Assistant Professor, Dept. of ECE, Eluru College of Engineering and Technology, ELURU, AP

ABSTRACT: Built-in self-test or BIST is a structural test method that allows the IC to periodically test its own operation. BIST uses a Pseudo-Random Pattern Generator (TPG) to generate the random test patterns that are applied to the test circuit. The nature of the generator thus directly influences the fault coverage achieved. Modern technology has focused on developing low-power systems for very-large scale integration (VLSI) high-speed designs. As a result, several design strategies have been implemented to mitigate trade-offs between performance, power, and area. Instead, several approaches have concentrated on low-power dissipation during BIST normal-mode operations rather than testmode operations. During the BIST test mode operation, the switching activity in the scan chains and test data compression using the appropriate TPG are crucial. Moreover, this testing should be achieved with high reliability and sensitivity. The objective of this existing project is to propose a new weighted TPG for a scan-based BIST architecture to generate efficient weighted patterns for enabling scan chains with reduced power consumption and area. Additionally, the pseudo-primary seed of TPG is maximized to obtain a considerable length in the weighted pseudorandom patterns. The maximum length weighted patterns are executed by assigning separate weights to the specific scan chains using a weight-enabled clock. This approach reduces the hardware overhead and achieves a low power consumption. The weighted patterns are also generated with fewer switching transitions and higher fault coverages. This process is observed with different circuits under test as their scan chains. The simulation results are tested and the results of the proposed TPG design are compared with existing potential TPG designs. This innovative method proposes a novel test pattern generator (TPG) for built-in self-test. Our method generates multiple single input change vectors in a pattern, i.e., each vector applied to a scan chain is an SIC vector. A reconfigurable Johnson counter and a scalable SIC counter are developed to generate a class of minimum transition sequences. The proposed TPG is flexible to both the test-per-clock and the test-per-scan schemes.

KEYWORDS: BIST, reachable states, on-chip, L.F.S.R, fixed hardware architecture, broad side test, decoder.

INTRODUCTION: Technology provides smaller, faster and lower energy devices which allow more powerful and compact circuitry, however, these benefits come with cost-the nano scale devices may be less reliable, thermal-and shot- noise estimations alone suggest that the fault rate of an individual nanoscale device may be orders of magnitude higher than today's devices. As a result, we can expect combinational logic be susceptible to faults. So in order to test any circuit or device we require separate testing technique which should be done automatically, for that purpose we are going to BIST. In recent years, the design for low power has become one of the greatest challenges in high-performance very large scale integration (VLSI) design. As a consequence, many techniques have been introduced to minimize the power consumption of new VLSI systems. However, most of these methods focus on the power consumption during normal mode operation, while test mode operation has not normally been a predominant concern. However, it has been found that the power consumed during test mode operation is often much higher than during normal mode operation [1]. This is because most of the consumed power results from the switching activity in the nodes of the circuit under test (CUT), which is much higher during test mode than during normal mode operation [1]-[3]. Several techniques that have been developed to reduce the peak and average power dissipated during scan-based tests can be found in [4] and [5]. A direct technique to reduce power consumption is by running the test at a slower frequency than that in normal mode. This technique of reducing power consumption, while easy to implement, significantly increases the test application time [6]. Furthermore, it fails in reducing peak-power consumption since it is independent of clock frequency. Another category of techniques used to reduce the power consumption in scan-based built-in self-tests (BISTs) is by using scanchain-ordering techniques [7]-[13]. These techniques aim to reduce the average-power consumption when scanning in test vectors and scanning out captured responses. Although these algorithms aim to reduce average-power consumption, they can reduce the peak power that may occur in the CUT during the scanning cycles, but not the capture power that may result during the test cycle (i.e., between launch and capture). The design of low-transition test-pattern generators (TPGs) is one of the most common and efficient techniques for low-power tests [14]-[20]. These algorithms modify the test vectors generated by the LFSR to get test vectors with a low number of transitions. The main drawback of these algorithms is that they aim only to reduce the average-power consumption while loading a new test vector, and they ignore the power consumption that results while scanning out the captured response or during the test cycle. Furthermore, some of these techniques may result in lower fault coverage and higher test application time. Other techniques to reduce average-power consumption during scan-based tests include scan segmentation into multiple scan chains [6], [21], test-scheduling techniques [22], [23], static compaction techniques [24], and multiple scan chains with many scan enable inputs to activate one scan

chain at a time [25]. The latter technique also reduces the peak power in the CUT. OVERTESTING due to the application of two-patternscan-based tests was described in [1]-[3]. Over testing is related to the detection of delay faults under non-functional operation conditions. One of the reasons for these non-functional operation conditions is the following. When an arbitrary state is used as a scan-in state, a two-pattern test can take the circuit through state-transitions that cannot occur during functional operation. As a result, slow paths that cannot be sensitized during functional operation may cause the circuit to fail [1]. In addition, current demands that are higher than those possible during functional operation may cause voltage drops that will slow the circuit and cause it to fail [2], [3]. In both cases, the circuit will operate correctly during functional operation. Functional broadside tests [4] ensure that the scan-in state is a state that the circuit can enter during functional operation, or a reachable state. As broadside tests [5], they operate the circuit in functional mode for two clock cycles after an initial state is scanned in. This results in the application of a two-pattern test. Since the scan-in state is a reachable state, the two-pattern test takes the circuit through state-transitions that are guaranteed to be possible during functional operation. Delay faults that are detected by the test can also affect functional operation, and the current demands do not exceed those possible during functional operation. This alleviates the type of overtesting described in [1]-[3]. In addition, the power dissipation during fast functional clock cycles of functional broadside tests does not exceed that possible during functional operation.

Built in Self-Test (BIST): The trend to include more test logic on an ASIC has already been mentioned. Built-in self-test (BIST) is a set of structured-test techniques for combinational and sequential logic, memories, multipliers, and other embedded logic blocks. In each case the principle is to generate test vectors, apply them to the circuit under test (CUT) or device under test (DUT), and then check the response. BIST is a viable approach to test today's digital systems. With the ever increasing need for system integration, the trend today is to include in the same VLSI device a large number of functional blocks, and to package such devices, often, in Multi-Chip Modules (MCMs) that comprise complex systems. This leads to difficult testing problems in the manufacturing process and in the field. An attractive approach to solve these problems is to use a multi-level integrated Built-In Self-Test (BIST) strategy. This strategy assumes that BIST is used at each level of manufacturing test, and it is reused at all consecutive levels, i.e. device, MCM, board, system. Boundary-Scan standard to realize self-testing at different levels. This strategy can only be realized.

LFSR: The only linear function of single bits is xor, thus it is a shift register whose input bit is driven by the exclusive-or (xor) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen

feedback function can produce a sequence of bits which appears random and which has a very long cycle. Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common. Linear feedback shift registers (LFSRs) are commonly used as test pattern generators (TPGs) in low overhead built-in self-test (BIST) schemes. This is due to the fact that an LFSR can be built with little area overhead. Attainment of high fault coverage with sequences of practical lengths has traditionally been the main objective of BIST techniques. Even though this still remains the main objective, we believe, reducing heat dissipated during test application is becoming another important objective. The correlation between consecutive random patterns generated by an LFSR is low; this is a well-known property of LFSR-generated patterns. On the other hand, a significant correlation exists between consecutive vectors applied to the inputs of a circuit during its normal operation. Hence, switching activity in a circuit can be significantly higher during self-test than that during its normal operation. The hardware used in this paper for generating the primary input sequence consists of a linear-feedback shift-register (LFSR) as a random source [17], and of a small number of gates (at most six gates are needed for every one of the benchmark circuits considered). The gates are used for modifying the random sequence in order to avoid cases where the sequence takes the circuit into the same or similar reachable states repeatedly. This is referred to as repeated synchronization [18]. In addition, the on-chip test generation hardware consists of a single gate that is used for determining which tests based on will be applied to the circuit. The result is a simple and fixed hardware structure, which is tailored to a given circuit only through the following parameters.

- 1) The number of LFSR bits.
- 2) The length of the primary input sequence.
- 3) The specific gates used for modifying the LFSR sequence into the sequence.
- 4) The specific gate used for selecting the functional broad- side tests that will be applied to the circuit based on.
- 5) Seeds for the LFSR in order to generate several primary inputs.

EXISTING METHOD: Compared with the existing methods, the proposed weighted TPG is designed with some advantages, including fewer switching transitions achieved using the specific weighted patterns and reduced power attained using fewer hardware components in the design. This reduces the hardware overhead and improves the fault coverages in the BIST. The TPG method shown in below Figure is the proposed TPG, which includes the Galois operation and additional hardware for weighted pattern generation. The Galois operation in the proposed TPG is shown by the black dashed line and assumes constant pseudo-primary seeds (A, X) for simplification. However, the constant seed bits can be enlarged using the same subset of initial primary seeds. The seed subsets are used to achieve the maximum length in weighted patterns with less switching activity. The additional hardware indicated by the blue line uses a

smaller number of components for generating the weighted pseudorandom TPG output. In addition, the additional hardware design uses a weight-enabled clock, which enables specific weights through successive clock cycles. The particular weights are given to the respective scan chains through the weighted Mux. The weight generator clock selection effectively reduces the fault coverage in terms of the random-pattern resistant fault and the redundant faults in the BIST architecture. A 3-bit pseudorandom TPG is proposed according to the Galois scheme over a field of $GF(2^m)$. The test patterns are generated concurrently using the shift registers and Galois operation. The synchronous clock for the TPG leads the bit sequences to be lost while it is incorporated for m -bits. Hence, the m -bit TPG is designed using asynchronous clocks in shift registers. The input vector bit (X) is multiplied continuously by the pseudo-primary seed bit (A) and added to the test vectors (Z). In addition, the state of the registers accommodates the multilevel parallelism in the TPGs. Consequently, the next $(i+1)$ -th state after the i -th state is described in terms of the feedback loop structure.

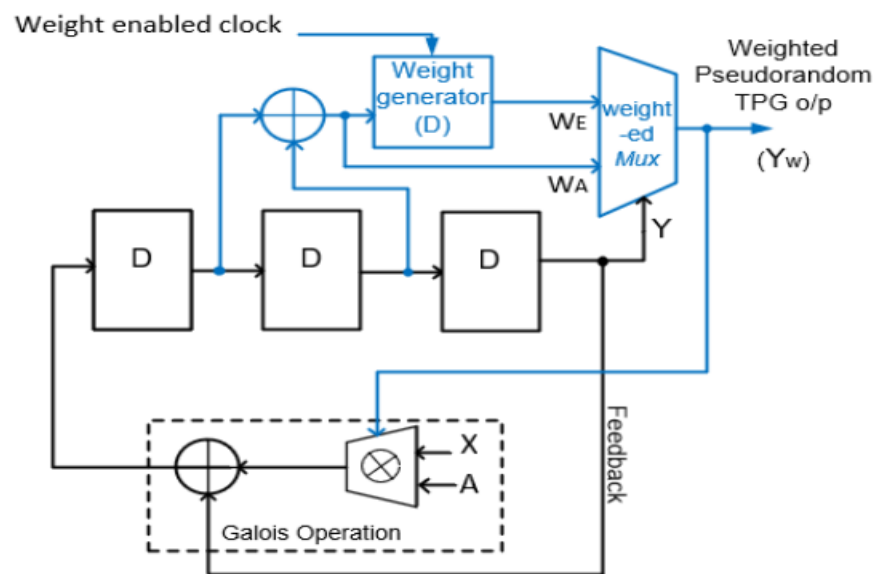


FIGURE1. Proposed 3-bit weighted pseudorandom TPG

Furthermore, the constant pseudo-primary seeds are enlarged in the Galois operation of the proposed TPG design. Furthermore, the constant pseudo-primary seeds are enlarged in the Galois operation of the proposed TPG design.

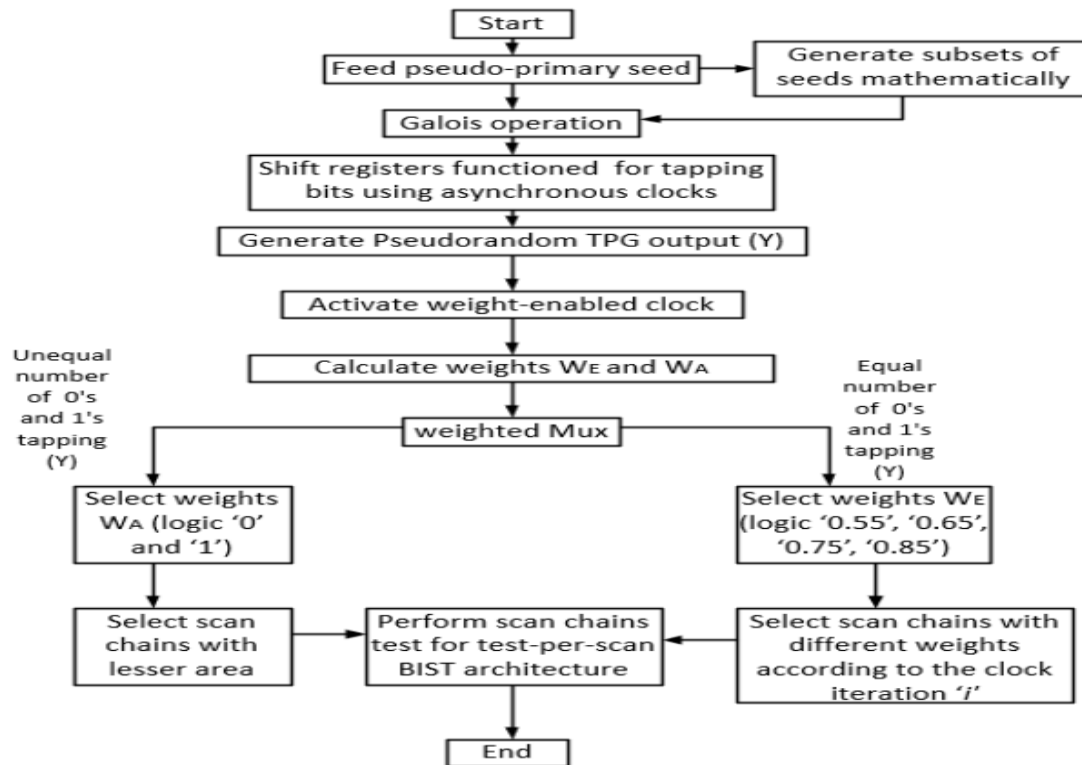


FIGURE2. Flowchart of the proposed weighted pseudorandom TPG operation

A flowchart summary of the proposed weighted TPG operation is shown in Figure 5. The weighted Mux acts as a phase shifter to shift the actual and estimated weighted patterns to the scan chains. The weighted Mux also selects the convolution bits W_E and W_A with self-control, as the weighted patterns W_E and W_A can be shifted to the output Y_w according to the pseudorandom test patterns (Y). The scan chains are identified by the weighted patterns W_E and W_A using the following essential properties. 1) First, the weighted patterns W_A generated with the probabilities of having '0' or '1' assigned to the certain scan chains occupy a smaller area. The output of the weighted Mux depends on the important feature of the pseudorandomness of the seed inputs. Consider a case in which Y_0 will be swapped with Y_1, Y_2 until Y_n , according to the value of the Galois operation (Z) in the proposed 3-bit TPG. Here, Y_2 is Y , which is connected to the selection input of the Mux. This determines the weighted pattern. Hence, overall switching transitions in the scan chain primary inputs can be reduced by 25%. 2) Second, the weighted patterns W_E are selected by the weighted Mux if the pseudorandom output (Y) generates an equal number of '0' or '1' values after swapped with Y_1, Y_2 until Y_n , according to the value of the Galois operation (Z) in the proposed 3-bit TPG. Here, Y_2 is Y , which is connected to the selection input of the Mux. This determines the weighted pattern. Hence, overall switching transitions in the scan chain primary inputs can be reduced

by 25%. 2) Second, the weighted patterns WE are selected by the weighted Mux if the pseudorandom output (Y) generates an equal number of '0' or '1' values after

OPERATION OF THE PROPOSED WEIGHTED TPG FOR THE SCAN CHAINS SELECTION

Iteration	Multiplexed Inputs		Shift reg. o/p			TPG o/p	Actual Weight	Estimated weight				Scan chains selection
	a_0	x_0	$D1$	$D2$	$D3$	Y	$W_A(i)$	$W_E(i+j)$				Y_w
								$i+1$	$i+2$	$i+3$	$i+4$	
0	0	0	0	0	0	0	0	0.55	0.65	0.75	0.85	S_0, S_1, S_2, S_3, S_4
1	0	1	0	0	1	1	0	0.75	0.85	0.65	0.55	S_0, S_3, S_4, S_2, S_1
2	1	0	1	1	0	0	0	0.65	0.55	0.75	0.85	S_0, S_2, S_1, S_3, S_4
3	1	1	0	1	1	0	1	0.55	0.75	0.85	0.65	S_5, S_1, S_3, S_4, S_2
4	0	0	1	0	1	0	1	0.75	0.55	0.65	0.85	S_5, S_3, S_1, S_2, S_4
5	0	1	0	1	0	1	1	0.85	0.65	0.55	0.75	S_5, S_4, S_2, S_1, S_3
6	1	0	1	1	1	1	0	0.65	0.75	0.85	0.55	S_0, S_2, S_3, S_4, S_1
7	1	1	1	0	0	1	1	0.55	0.85	0.75	0.65	S_5, S_1, S_4, S_3, S_2

The weights generated by the TPG in different clock cycles are assumed to be $w_0, w_1, w_2, \dots, w_n \in \{0.55, 0.65, 0.75, 0.85\}$, which are the probability distributions. The respective weights are assigned to the scan chains as $S_0, S_1, S_2, \dots, S_n$. Here, n denotes the total number of scan chains to be tested. The full-length test patterns are weighted based on the probability distribution analysis. For scan chain selection, the weighted patterns are calculated in the estimated times of $i+1, i+2, i+3$, and $i+4$. The probability of the weights is considered to insert more scan-shift cycles rather than capture cycles. Hence, the proposed TPG compares the estimated and actual weights at successive clock cycles to detect their faults. The conventional TPG updates the next stages linearly without a weighted function, which is not consistent. The comparison of weights for selecting the scan chains allows specifying the output from the shift register at each clock cycle. The operation of the proposed 3-bit TPG with the weighted functions is shown in Table II. The seed bit polynomial $Y[i]$ is defined as $1 + x^3$ for even weights and $1 + x + x^3$ for odd weights. The successive weights are generated as an even parity of '0' and an odd parity of '1' concurrently by the Galois operation using equation (1). Additionally, the last term $W[\sum_{i=0}^{n-1} x^n m-1]$, indicated in equation (4), is assumed to be $W[x0] = W_A$ in the TPG. Initially, the weight W_A accumulates in the weighted Mux, and later, at the $(i + 1)$ -th iteration, the weight W_E is achieved. Although W_A generates both even and odd parities, the additional W_E bits are continuously defined for accurate weights in the TPG output. This is represented as $W_E[i + 1]$ in equation (5) and can be accomplished in the weight generator using the weight enable signal and the tapped convolution values of the cascaded register function. According to the weights W_A and W_E using additional hardware, eight iterations can occur during the duration of the $(i+j)$ -th clock cycle. The $(i+j)$ -th clock cycle weighted patterns W_E are listed as $i, i+1, i+2, i+3$ and $i+4$. The overall repetitive weighted clock cycle selects the scan chains using weights from '0' to '1' as 0, 0.55, 0.65, 0.75, 0.85, and 1. The

weighted clock is operated as an asynchronous clock signal. The different operating frequencies of the weighted clock are determined using the decimal values of the asynchronous D flip-flops as a weight generator. However, the scan chains are partitioned into six numbers based on the weighted pattern selection; this process selects more than one number of scan chains at any clock cycle due to its designated weight function.

DRAWBACKS:

Area overhead

Performance penalties

Additional design time & effort

Additional risk to project

Lack of orthogonal testing

PROPOSED METHOD:

MSIC TEST PATTERN GENERATOR:

Multiple Single Input Change (MSIC) is a test pattern generator (TPG) method that can change a Single Input Change (SIC) vector to exclusive low transition vectors for multiple scan chains [7],[9]. The first step in this process is to decompress the SIC vector to its multiple code words and the generated code words will bit-XOR with a same seed vector in turn. Hence, a test pattern with similar test vectors will be applied to all scan chains. The MSIC TPG consists of an SIC generator, a seed generator, an XOR gate network, and a clock and control block. Test Pattern Generation Method Consider m primary inputs (PIs) and M scan chains in a full scan design, and each scan chain has l scan cells. Figure shows the symbolic representation for one generated pattern. The generated vector of an m -bit LFSR with the primitive polynomial can be expressed as $S(t) = S_0(t)S_1(t)S_2(t), \dots, S_{m-1}(t)$ (hereinafter referred to as the seed), and the vector generated by an l -bit Johnson counter can be expressed as $J(t) = J_0(t)J_1(t)J_2(t), \dots, J_{l-1}(t)$. In the first clock cycle, $J = J_0 J_1 J_2, \dots, J_{l-1}$ will bit-XOR with $S = S_0 S_1 S_2, \dots, S_{m-1}$, and the results $X_1 X_{l+1} X_{2l+1}, \dots, X_{(M-1)l+1}$ will be shifted into M scan chains, respectively. In the second clock cycle, $J = J_0 J_1 J_2, \dots, J_{l-1}$ will be circularly shifted as $J = J_{l-1} J_0 J_1, \dots, J_{l-2}$, which will also bit-XOR with the seed $S = S_0 S_1 S_2, \dots, S_{m-1}$. The resulting $X_2 X_{l+2} X_{2l+2}, \dots, X_{(M-1)l+2}$ will be shifted into M scan chains, respectively. After l clocks, each scan chain will be fully loaded with a unique Johnson codeword, and seed $S_0 S_1 S_2, \dots, S_{m-1}$ will be applied to m PIs. Since the circular Johnson counter can generate l unique Johnson codewords through circular shifting a Johnson vector, the XOR gates and circular Johnson counter actually constitute a linear sequential decompressor. There are two kinds of SIC generators based on the different scenarios of scan length, to generate Johnson vectors and Johnson codewords, that is; the reconfigurable Johnson counter and the scalable SIC counter. For a short scan length, a reconfigurable Johnson counter is

Copyright @ 2022ijearst. All rights reserved.

INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY

Volume.02, IssueNo.04, May-2022, Pages: 241-255

developed as shown in Figure 3 to generate an SIC sequence in time domain. An SIC counter named the “scalable SIC counter” is developed when the maximal scan chain length l is much larger than the scan chain number M . The SIC counter consist of subtractor, adder, chain of D flipflops and chain of muxes.

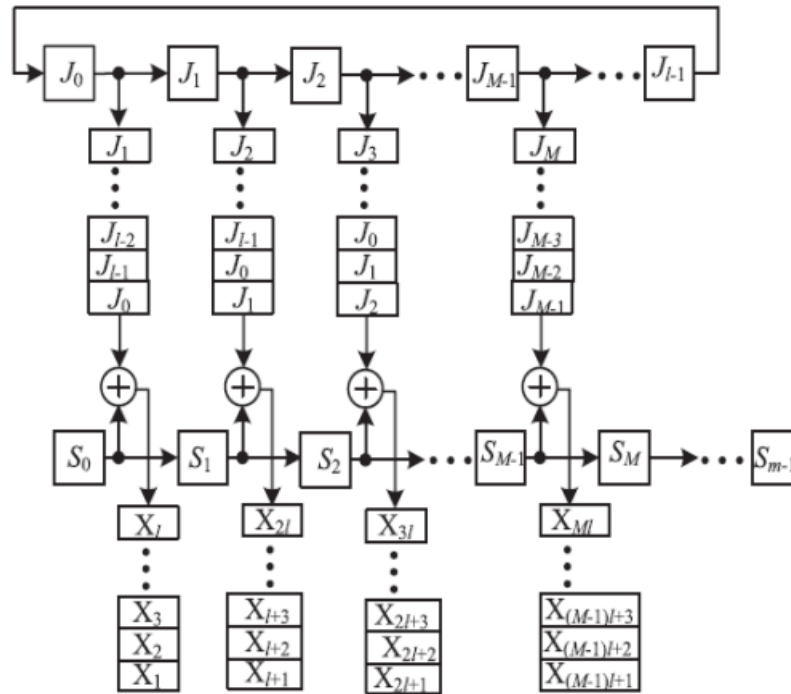


Figure3: Symbolic representation of an MSIC pattern

JOHNSON COUNTER:

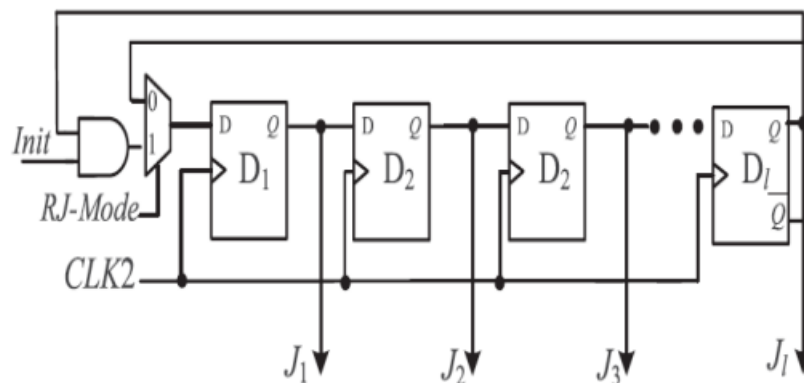


Figure4: Reconfigurable Johnson counter.

Reconfigurable Johnson Counter is preferably used when the scan length is less. The reconfigurable Johnson counter consists of a mux and an AND gate to make it operate in three different modes. The control signals for the reconfigurable Johnson counter are Init and RJ_Mode. The reconfigurable Johnson counter has three modes of operation: 1)Initialization: When Init is set to logic 0 and RJ_Mode is set to 1, the reconfigurable Johnson counter will be initialized to all zero states by clocking CLK2 more than 1 times. 2)Circular shift register mode: When Init and RJ_Mode are set to logic 1, each stage of the Johnson counter will output a Johnson codeword by clocking CLK2 1 times. 3)Normal mode: The reconfigurable Johnson counter will generate $2l$ unique SIC vectors by clocking CLK2 $2l$ times when RJ_Mode is set to logic 0

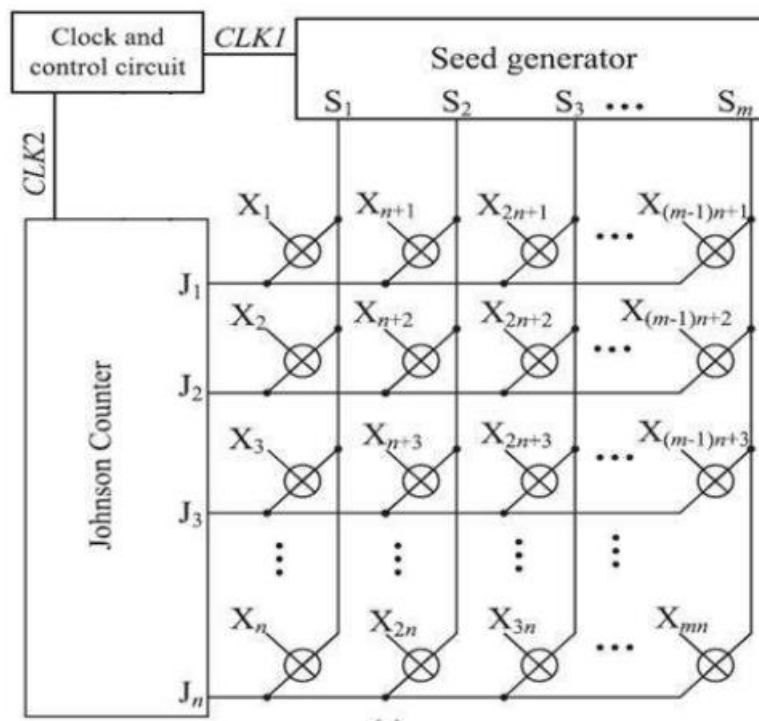


Figure5: MSIC-TPGs for test-per-clock scheme

The MSIC-TPG for test-per-clock schemes is illustrated in above Figure. The circuit under test's primary inputs $X_1 - X_{mn}$ are arranged as an $n \times m$ SRAM-like grid structure. Each grid has a two-input XOR gate whose inputs are tapped from a seed generator output and an output of the Johnson counter. The outputs of the XOR gates are applied to the circuit under test's primary inputs. A seed generator is an m -stage conventional LFSR, and operates at low frequency CLK1. The test procedure for the test per clock scheme is as follows: 1)The seed generator generates a new seed by clocking CLK1 one time. 2)The Johnson

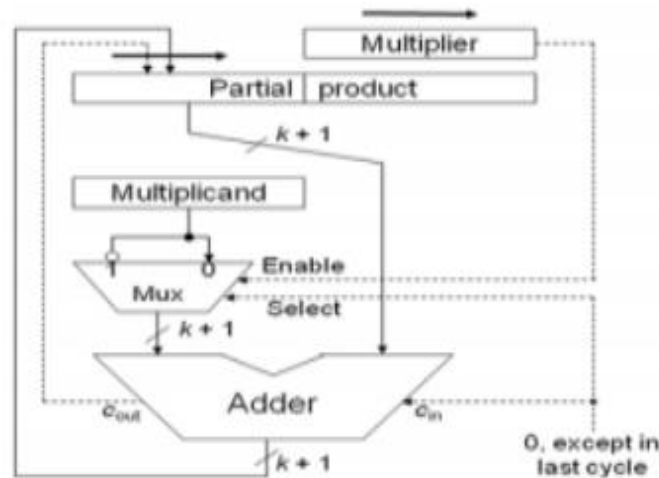
Copyright © 2022ijearst. All rights reserved.

INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY

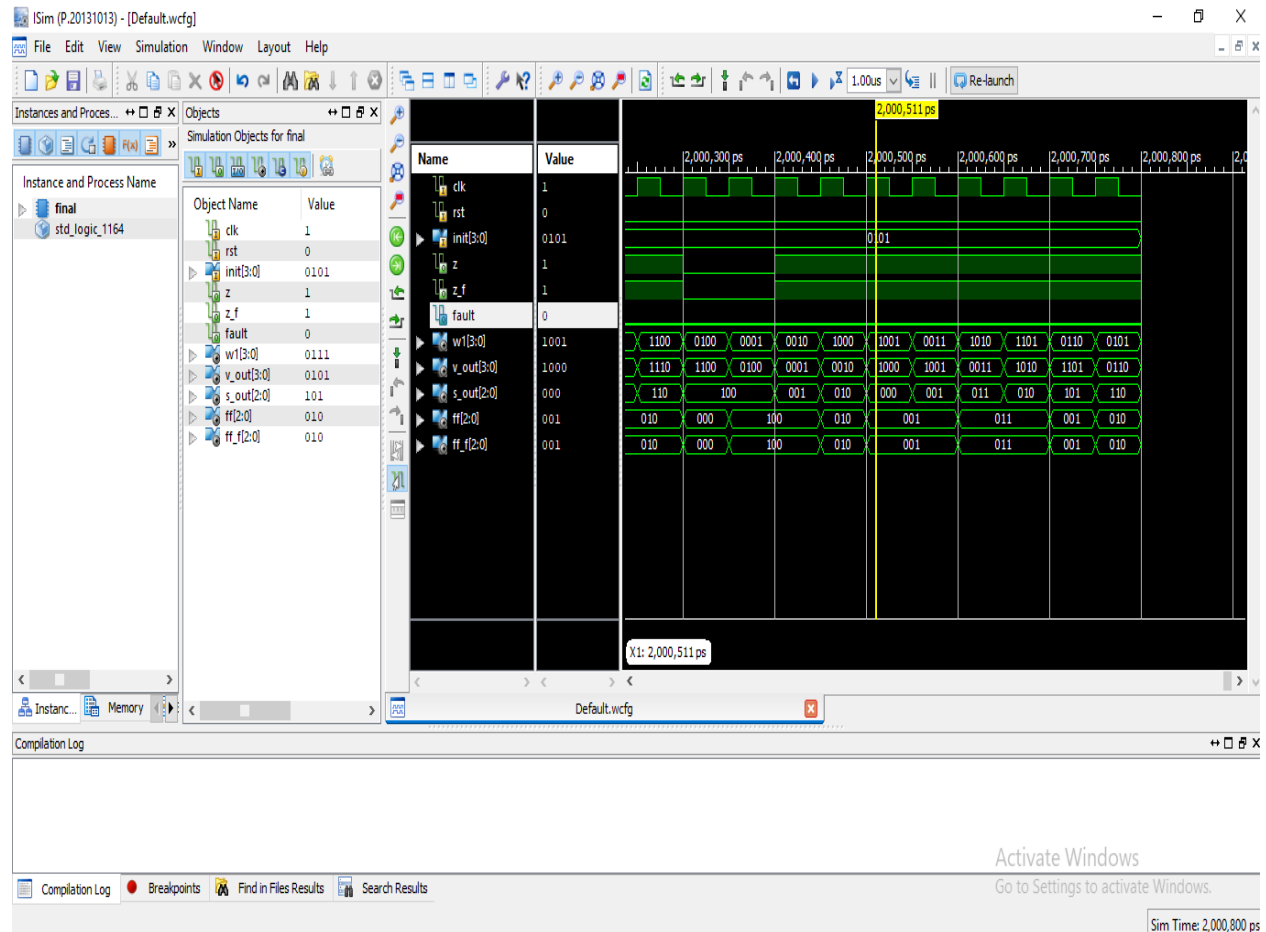
Volume.02, IssueNo.04, May-2022, Pages: 241-255

CIRCUIT UNDER TEST:

In signed multiplication the length of the partial products and the number of partial products will be very high. So an algorithm was introduced for signed multiplication called as BaughWooley algorithm. The Baugh-Wooley multiplication is one amongst the cost-effective ways to handle the sign bits. This method has been developed so as to style regular multipliers, suited to 2's complement numbers. Baugh-Wooley multiplier hardware architecture is shown in figure 1. It follows left shift algorithm. Mux can select which bit will multiply. Suppose we multiply +4 and -4 in decimal we get '0'. Now, after representing these numbers in two's complement form we get +4 as 0100 and -4 as 1100. On adding these two binary numbers we get 10000. Discard carry, then number is represented as '0'.



2's Compliments is the most popular method in representing signed integers in Computer sciences. It is also an operation of negation (Converting positive to negative numbers or vice versa) in computers which represent negative numbers using two's complements. Its use is so wide today because it does not require the addition and subtraction circuitry to examine the signs of the operands to determine whether to add or subtract. Two's complement and one's complement representations are commonly used since arithmetic units are simpler to design.

RESULTS:**Fig7: Proposed simulation result**

Conclusion and future scope: This paper presents a low hardware overhead TPG for scan-based BIST that can reduce switching activity in CUTs during BIST and also achieve very high fault coverage with a reasonable length of test sequence with fixed hardware architecture. Unacceptably long test sequences are often required to attain high fault coverage with pseudo-random test patterns for circuits that have many random pattern resistant faults. The main objective of most recent BIST techniques has been the design of TPGs that achieve high fault coverage at acceptable test lengths for such circuits. While this objective still remains important, reducing heat dissipation during test application is also becoming an important objective. Since the correlation between consecutive patterns applied to a circuit during BIST is significantly lower, switching activity in the circuit can be significantly higher during BIST than that during its normal operation. Excessive switching activity during test application can cause several problems. For further modification we can replace the logic memory cells with static or dynamic memory cells so that we can

Copyright © 2022ijearst. All rights reserved.

INTERNATIONAL JOURNAL OF ENGINEERING IN ADVANCED RESEARCH
SCIENCE AND TECHNOLOGY

Volume.02, IssueNo.04, May-2022, Pages: 241-255

reduce area overhead. The proposed BIST architecture is more efficient using cellular automata as a test pattern generator

REFERENCES:

- [1] P. H. Bardell, W. H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques. New York: Wiley, 1987.
- [2] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, "Built-In test for circuits with scan based on reseeding of multiple- polynomial linear feedback shift registers," IEEE Trans. Comput., vol. 44, no. 2, pp. 223-233, Feb. 1995.
- [3] N. Zacharia, J. Rajski, and J. Tyszer, "Decompression of test data using variable-length seed LFSRs," in Proc. IEEE 13th VLSI Test Symp., 1995, pp. 426-433.
- [4] S. Hellebrand, S. Tarnick, and J. Rajski, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers," in Proc. IEEE Int. Test Conf., 1992, pp. 120-129
- [5] N. A. Touba and E. J. McCluskey, "Altering a pseudo-random bit sequence for scan-based BIST," in Proc. IEEE Int. Test Conf., 1996, pp. 167-175.
- [6] M. Chatterjee and D. K. Pradhan, "A new pattern biasing technique for BIST," in Proc. VLSITS, 1995, pp. 417-425.
- [7] N. Tamarapalli and J. Rajski, "Constructive multi-phase test point in section for scan-based BIST," in Proc. IEEE Int. Test Conf., 1996, pp. 649-658.
- [8] Y. Savaria, B. Lague, and B. Kaminska, "A pragmatic approach to the design of self-testing circuits," in Proc. IEEE Int. Test Conf., 1989, pp. 745-754.
- [9] J. Hartmann and G. Kemnitz, "How to do weighted random testing for BIST," in Proc. IEEE Int. Conf. Comput.-Aided Design, 1993, pp. 568-571.
- [10] J. Waicukauski, E. Lindbloom, E. Eichelberger, and O. Forlenza, "A method for generating weighted random test patterns," IEEE Trans. Comput., vol. 33, no. 2, pp. 149-161, Mar. 1989.
- [11] H.-C. Tsai, K.-T. Cheng, C.-J. Lin, and S. Bhawmik, "Efficient test- point selection for scan-based BIST," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 6, no. 4, pp. 667-676, Dec. 1998.
- [12] W. Li, C. Yu, S. M. Reddy, and I. Pomeranz, "A scan BIST generation method using a markov source and partial BIST bit-fixing," in Proc. IEEE-ACM Design Autom. Conf., 2003, pp. 554-559.
- [13] N. Z. Basturkmen, S. M. Reddy, and I. Pomeranz, "Pseudo random patterns using markov sources for scan BIST," in Proc. IEEE Int. Test Conf., 2002, pp. 1013-1021.
- [14] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in Proc. VLSI Testing Symp., 1993, pp. 4-9.
- [15] S. W. Golomb, Shift Register Sequences. Laguna Hills, CA: Aegean Park, 1982.
- [16] C.-Y. Tsui, M. Pedram, C.-A. Chen, and A. M. Despain, "Low power state assignment targeting two- and multi-level logic implementation," in Proc. IEEE Int. Conf. Comput.-Aided Des., 1994, pp. 82-87.
- [17] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design," in Proc. VLSI Test. Symp., 1999, pp. 407-412.
- [18] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 17, no. 12, pp. 1325-1333, Dec. 1998.
- [19] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Scheduling tests for VLSI systems under power constraints," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 5, no. 2, pp. 175-185, Jun. 1997.