ENGINEERING IN ADVANCED RESEARCHSCIENCE AND TECHNOLOGY

ISSN 2352-8648
Vol.03, Issue.01
June-2022
Pages: -548-560

AN AREA - EFFICIENT AND LOW - LATENCY DESIGN OF A WEIGHTED PTPG FOR A BIST ARCHITECTURE

[1]AKHILA BEGUM, [2]Dr. P. H. S. TEJO MURTHY, [3]CH. JAYA PRAKASH, [4]Dr. I. HEMALATHA

[1]PG Student, Dept. Of ECE, Sir C.R. Reddy College of Engineering, Eluru, A.P
[2]Professor, Dept. of ECE, Sir C.R. Reddy College of Engineering, Eluru, A.P
[3]Assistant Professor, Dept. of ECE, Sir C.R. Reddy College of Engineering, Eluru, A.P
[4]Associate Professor, Dept. of ECE, Sir C.R. Reddy College of Engineering, Eluru, A.P

ABSTRACT: With growing complexity of integrated circuits and systems, the cost of testing has become ever more significant. Design-for-testability (DFT) is become more and more important and is the direction of the development of the IC test industry. Built-In Self-Test (BIST) is increasingly being applied as an effective means to reduce the cost of testing. BIST is a DFT technique in which testing (test generation and test application) is accomplished through built-in hardware features. It can potentially eliminate the need for external test equipment and introduce the capability for testing devices after the circuit is integrated in a system. BIST uses a Pseudo-Random Pattern Generator to generate the random test patterns that are applied to the test circuit. In conventional BIST architectures, the linear feedback shift register (LFSR) is commonly used in the test pattern generators and the major drawback of these architectures is that the pseudorandom patterns generated by the LFSR lead to significantly high switching activities in the CUT which can cause excessive power dissipation. They can also damage the circuit and reduce product yield and lifetime. In addition, the LFSR usually needs to generate very long pseudorandom sequences in order to achieve the target fault coverage in nanometer technology. Thus, the nature of the generator 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 test mode 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 existing method propose a new weighted TPG for BIST architecture to generate efficient weighted patterns for enabling scan chains with reduced power consumption and area. 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. The proposed MSIC (Multiple Single Input Change)-TPG consists of a Johnson counter, a seed generator and an XOR gate network. The Johnson counter generates Johnson vectors that has low transition properties. The Johnson sequence is a single input change sequence (SIC) i.e., the vector generated by the next clock in the sequence is a one-bit change from the previous clock generation vector. First, the SIC vector is decompressed to its multiple code words. Meanwhile, the generated code words will bit-XOR with the seed vectors and will be applied to all scan chains. By using this method, the area and the propagation delay can be reduced further. This process is observed with a circuit under test as their scan chains. The simulation results are tested using Xilinx and the results of the proposed TPG design are compared with existing potential TPG design.

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 scan chain-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 scans 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-pattern scan-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.

LITERATURE SURVEY: For large embedded memory cores, the manufacturing yield can be unacceptable low (e.g., for a 24Mbits memory core, the yield is around 20% [5]). Hence, to achieve a certain manufacturing yield, in addition to diagnosis support, it is also beneficial to introduce self-repair features comprising redundant memory cells. TPG methods like exhaustive, pseudo-random and fault simulation techniques (Hurst, 1998; Roth, 1998) are used in the test vector generation process. TPG is the process of generating the test vectors required to stimulate a circuit at the primary inputs so that effect of the considered fault (the fault effect) is propagated to the primary outputs. A difference between the fault free and faulty circuit can then be detected. It is common to derive a minimal set of test vectors as it will reduce the overall test set size and hence test time. In SOC, designers can specify the test speed, fault coverage, diagnostic options and test length for testing any random logic block. Power dissipation during the testing is one of most important issue (Crouch at. el.1999). Generally, a system or SOC block consumes more power in a test mode than the normal mode. It is observed that the power dissipation during test mode is 200% more than in normal mode (Whetsel, 2000). The test efficiency has been shown by (Zorian, 1993b) to have a high correlation with the toggle rate. The number of switching activities is more in the test mode than normal mode at all the nodes in the circuits/system. The DFT circuits like BIST is embedded in a system to reduce the test complexity and cost (Rajsuman, 2000). In SOC, to reduce the test time, the parallel testing is frequently employed, which generally results in excessive energy and power dissipation.The successive functional input vectors applied to a given circuit in normal mode have a significant correlation, while the correlation between consecutive test patterns is very low (Wang and Gupta, 1997). This causes large switching activity in the circuit during test than its normal operation. Power dissipation in CMOS circuits is proportional to switching activity. The excessive switching activity during test is responsible for cost, reliability, performance verification, power dissipation and technology related problems (Girard, 2002). Hence, it is important to optimize power during testing. (Corno et al., 1997) proposed an environment to address testability analysis and TPG on VHDL descriptions at the RT-level. The proposed approach, based on a suitable fault model and an ATPG algorithm, was experimentally shown to provide a good estimate

of the final gate-level fault coverage, and to give test patterns with excellent fault coverage properties. The approach, being based on an abstract representation, was particularly suited for large circuits, where gate level ATPGs is often inefficient. (Corno et al., 1998) proposed an ATPG technique that reduces power dissipation during the testing of sequential circuits. The proposed approach exploited some redundancy introduced during the TPG phase and selected a subset of sequences able to reduce the consumed power, without reducing the fault coverage. The method was composed of three independent steps: redundant TPG, power consumption measurement, and optimal test sequence selection. The experimental results gathered on the ISCAS benchmark circuits show that average power consumption is decreased by 70% with respect to the original test pattern, which also reduced the heat dissipation problem. (Lee and Touba, 2007) also proposed a new low power test data compression scheme based on LFSR reseeding. A drawback of compression schemes based on LFSR reseeding was that the unspecified bits were filled with random values, which results in a large number of transitions during scan-in, thereby causing high-power dissipation.

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.

Linear Feedback Shift Register (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.

- The number of LFSR bits.
- The length of the primary input sequence.
- The specific gates used for modifying the LFSR sequence into the sequence.
- The specific gate used for selecting the functional broad- side tests that will be applied to the circuit based on.
- 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 (2m). 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.

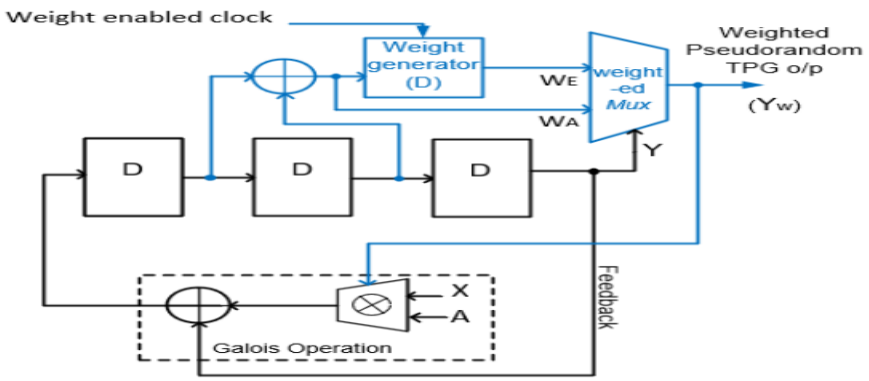
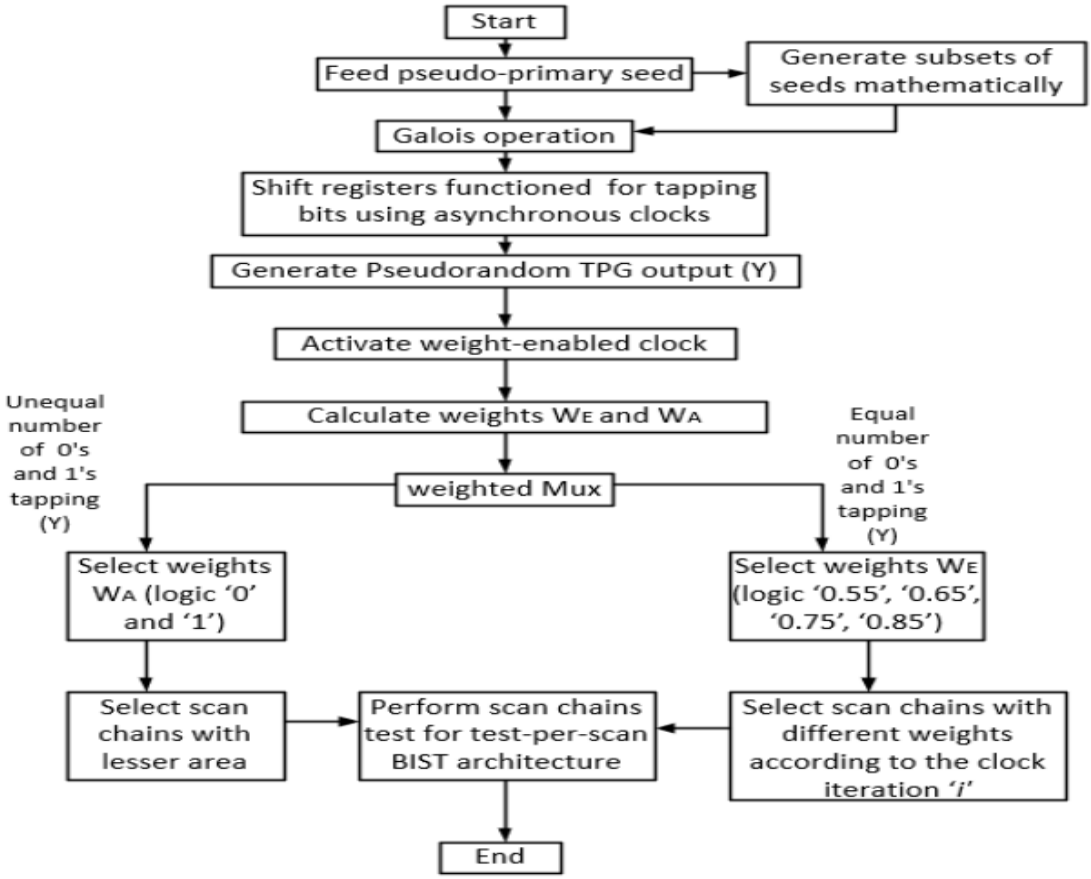


FIGURE1. Existing 3-bit weighted pseudorandom TPG

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



F IGURE2. Flowchart of the existing 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 WE and WA with self-control, as the weighted patterns WE and WA can be shifted to the output Yw according to the pseudorandom test patterns (Y). The scan chains are identified by the weighted patterns WE and WA

using the following essential properties. 1) First, the weighted patterns WA 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 pseudo randomness of the seed inputs. Consider a case in which Y0 will be swapped with Y1, Y2 until Yn, according to the value of the Galois operation (Z) in the proposed 3-bit TPG. Here, Y2 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 swapped with Y1, Y2 until Yn, according to the value of the Galois operation (Z) in the proposed 3-bit TPG. Here, Y2 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 swapping through the shift registers.

Table 1: Operation of the existing 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						$W_E(i+j)$				
			D1	*D2*	*D3*	*Y*	$W_A(i)$	*i+1*	*i+2*	*i+3*	*i+4*	Y_w
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 w0, w1, w2, ... wn ∈ {0.55, 0.65, 0.75,0.85}, which are the probability distributions. The respective weights are assigned to the scan chains as S0, S1, S2, ..., Sn. 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 [$\sum xn\ m-1\ i$=0], indicated in equation (4), is assumed to be W[x0] = WA in the TPG. Initially, the weight WA

accumulates in the weighted Mux, and later, at the $(i + 1)$-th iteration, the weight WE is achieved. Although WA generates both even and odd parities, the additional WE bits are continuously defined for accurate weights in the TPG output. This is represented as WE $[i + 1]$ 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 WA and WE 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 WE 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: The following are the drawbacks of the existing TPG.

- Area overhead
- More latency
- Performance penalties
- High density

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) = S0(t)S1(t)S2(t), . . ., Sm−1(t) (hereinafter referred to as the seed), and the vector generated by an l-bit Johnson counter can be expressed as J (t) = J0(t)J1(t)J2(t), . . ., Jl−1(t). In the first clock cycle, J = J0 J1 J2, . . ., Jl−1 will bit-XOR with S = S0S1S2, . . ., SM−1, and the results X1Xl+1X2l+1, . . ., X(M−1)l+1 will be shifted into M scan chains, respectively. In the second clock cycle, J = J0 J1 J2, . . ., Jl−1 will be circularly shifted as J = Jl−1 J0 J1, . . ., Jl−2, which will also bit-XOR with the seed S = S0S1S2, . . ., SM−1. The resulting X2Xl+2X2l+2, . . ., 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 S0S1S2, . . ., Sm−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 developed as shown in Figure 3 to generate an SIC sequence in time domain. A 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.

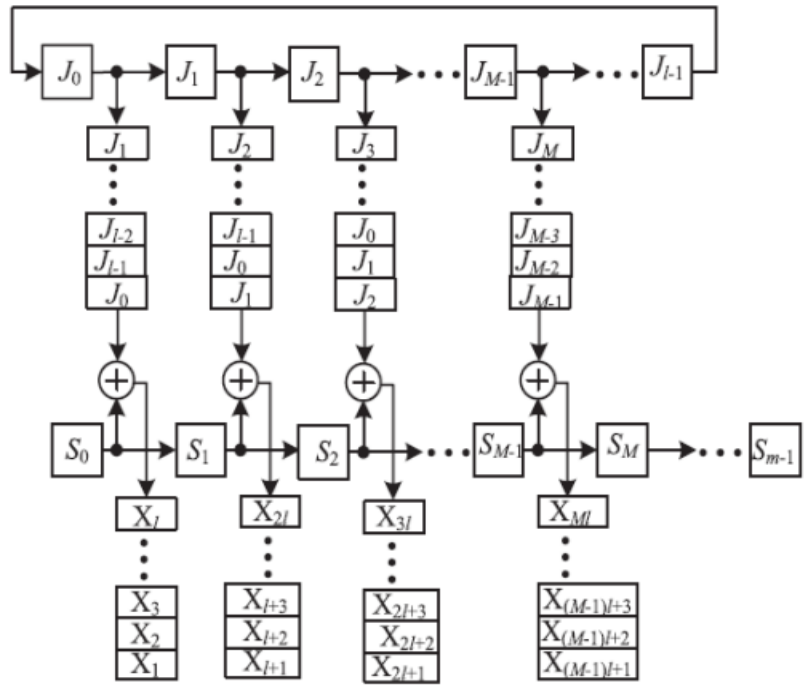


Figure3: Symbolic representation of an MSIC pattern

JOHNSON COUNTER:

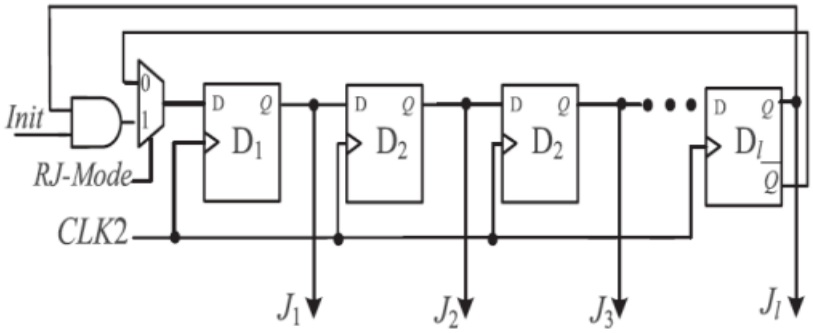


Figure4: Reconfigurable Johnson counter.

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:

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 l times.

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 l times.

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

CIRCUIT UNDER TEST - BAUGHWOOLEY MULTIPLIER:

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 Baugh Wooley 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 compliment numbers.

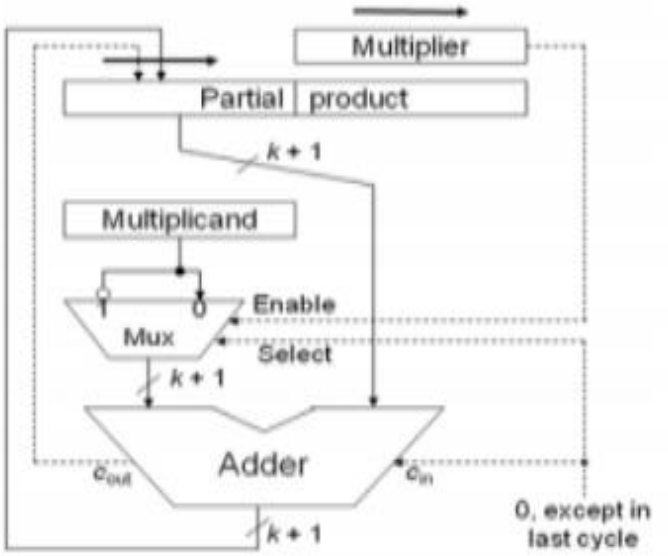
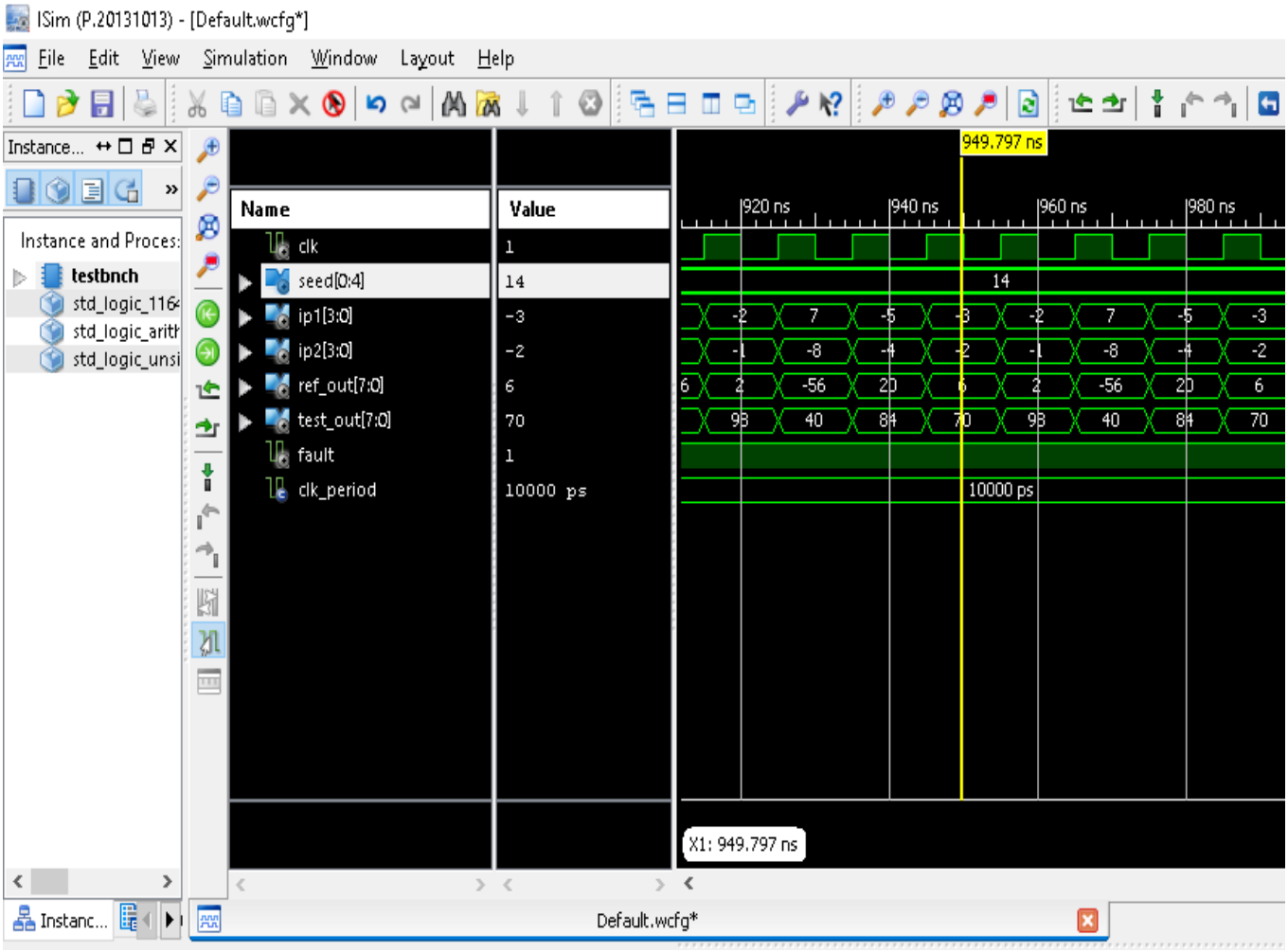


Figure 5: Baugh-Wooley multiplier block diagram

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 twos compliments. 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 compliment and one's compliment representations are commonly used since arithmetic units are simpler to design.

RESULTS:



The behavioral simulation results of BIST with proposed TPG when there is fault i.e., when there is difference between the actual output and the reference output of the circuit under test is shown in above Figure.

The comparison of gate count and time delay of BIST with Existing and Proposed TPG is shown in below table. From the below table we can conclude that the gate count of BIST with proposed TPG is reduced by 47.95% when compared to the gate count of BIST with existing TPG. Hence, we can say that the area is further reduced in proposed method. Similarly, we can observe that the time delay of BIST with proposed TPG is reduced by 0.951ns.

Table 2: Comparison of gate count and time delay of BIST with Existing and Proposed TPG

Parameter	Existing	Proposed
Area (Gate count)	269	129
Delay(ns)	11.463	10.512

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. Johnson counter can be easily implemented and flexible to test-per-scan schemes. The proposed weighted TPG is enhanced to implement a 4-bit TPG, guaranteeing low latency with lower area overhead. Correspondingly, the proposed TPG achieves approximately 47.95% reduction in the gate count and the delay is reduced by 0.951ns compared with the existing TPGs. Further modification can be done to the proposed method for the generation of TPG so that we can reduce area overhead and latency. This can be achieved by replacing the logic memory cells with static or dynamic memory cells.

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 regis-ters," in Proc. IEEE Int. Test Conf., 1992, pp. 120–129

[5] N. A. Touba and E. J. McCluskey, "Altering a pseudo-random bit se- quence for scan-based BIST," in Proc. IEEE Int. Test Conf., 1996, pp. 167–175.

[6] S. Hellebrand, S. Tarnick, J. Rajski, B. Courtois, and T. I. M. Imag, ``Multiple-polynomial linear feedback shift registers," 1992, pp. 120129.

[7] X. Lin and J. Rajski, ``Adaptive low shift power test pattern generator for logic BIST," in Proc. Asian Test Symp., 2010, pp. 355360, doi:10.1109/ATS.2010.67.

[8] A. S. Abu-Issa, ``Energy-efcient scheme for multiple scan-chains BIST using weight-based segmentation," IEEE Trans. Circuits Syst. II, Exp.Briefs, vol. 65, no. 3, pp. 361365, Mar. 2018, doi: 10.1109/TCSII.2016.2617160.

[9] G. S. Sankari and M. Maheswari, ``Energy efcientweighted test pattern generator based bist architecture," in Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC, 2019, pp. 448453, doi: 10.1109/I-SMAC.2018.8653768.

[10] R. Kapur, S. Patil, T. J. Snethen, and T. W. Williams, ``A weighted random pattern test generation system," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 15, no. 8, pp. 10201025, Aug. 1996, doi:10.1109/43.511581.

[11] A. Jas, C. V. Krishna, and N. A. Touba, ``Weighted pseudorandom hybrid BIST," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 12, pp. 12771283, Dec. 2004, doi: 10.1109/TVLSI.2004.837985.

[12] D. Xiang, M. Chen, and H. Fujiwara, ``Using weighted scan enable signals to improve test effectiveness of scan-based BIST," IEEE Trans. Comput., vol. 56, no. 12, pp. 16191628, Dec. 2007.

[13] H.-C. Tsai, K.-T. Cheng, and S. Bhawmik, ``On improving test quality of scan-based BIST," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 19, no. 8, pp. 928938, Aug. 2000, doi: 10.1109/43.856978.

[14] N. A. Touba and E. J. McCluskey, ``Altering a pseudo-random bit sequence for scan-based BIST," in Proc. IEEE Int. Test Conf., Oct. 1996, pp. 167175, doi: 10.1109/test.1996.556959.

[15] G. Kiefer, H. Vranken, E. J. Marinissen, and H. J. Wunderlich, ``Application of deterministic logic BIST on industrial circuits," J. Electron.Test. Theory Appl., vol. 17, nos. 34, pp. 351362, 2001, doi:10.1023/A:1012283800306.

[16] R. Cited and A. Banerjee, U.S. Patent, 2019, vol. 2.

[17] B. W. Johnson, J. H. Aylor, and H. H. Hana, ``Efcient use of time and hardware redundancy for concurrent error detection in a 32-bit VLSI adder," Comput. Arith., vol. 23, no. 1, pp. 171178, 2015, doi:10.1142/9789814641470./

[18] D. Xiang, X. Wen, and L.-T. Wang, ``Low-power scan-based built-in selftest based on weighted pseudorandom test pattern generation and reseeding,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 3, pp. 942953, Mar. 2017, doi: 10.1109/TVLSI.2016.2606248.

[19] H. Shin, S. Choi, J. Park, B. Y. Kong, and H. Yoo, ``Area-efcient error detection structure for linear feedback shift register," Electronics, vol. 9, no. 1, pp. 110, 2020.

[20] H. Dau, I. M. Duursma, H. M. Kiah, and O. Milenkovic, ``Repairing ReedSolomon codes with multiple erasures," IEEE Trans. Inf. Theory,vol. 64, no. 10, pp. 65676582, Oct. 2018, doi: 10.1109/TIT.2018.2827942.

[21] P. Wohl, J. A. Waicukauski, G. A. Maston, and J. E. Colburn, ``XLBIST: X-tolerant logic BIST," in Proc. IEEE Int. Test Conf. (ITC), Oct. 2018, pp. 19, doi: 10.1109/TEST.2018.8624738.

[22] E. Moghaddam, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer, and J. Zawada, ``Logic BIST with capture-per-clock hybrid test points, "IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 6, pp. 10281041, Jun. 2019, doi: 10.1109/TCAD.2018.2834441.

[23] H. Tran, ``Demonstration of 5T SRAMand 6T dual-port RAMcell arrays, '' in IEEE Symp. VLSI Circuits, Dig. Tech. Paper, Jun. 1996, pp. 6869, doi:10.1109/vlsic.1996.507719.

[24] S. Roy, B. Stiene, S. K. Millican, and V. D. Agrawal, ``Improved randompattern delay fault coverage using inversion test points," in Proc. IEEE28th North Atlantic Test Workshop (NATW), May 2019, pp. 16, doi:10.1109/NATW.2019.8758727.

[25] C. Senthilpari, K. Diwakar, and A. Singh, ``Low energy, low latency andhigh speed array divider circuit using a Shannon theorem-based adder cell," Recent Patents Nanotechnol., vol. 3, no. 1, pp. 6172, Jan. 2009, doi: 10.2174/187221009787003311.

[26] C. Senthilpari, A. K. Singh, and K. Diwakar, ``Design of a low-power, high performance, 88 bit multiplier using a Shannon-based adder cell," Microelectron. J., vol. 39, no. 5, pp. 812821, May 2008, doi: 10.1016/j.mejo.2007.12.016.

[27] R. F. W. Coates, G. J. Janacek, and K. V. Lever, ``Monte Carlo simulation and random number generation," IEEE J. Sel. Areas Commun., vol. SAC-6, no. 1, pp. 5866, Jan. 1988, doi: 10.1109/49.192730.