Research Article

Implementation of Energy Efficient BIST Architecture by Using Verilog

R.Banupriya[a], R.Nagarajan[b], and S.Kannadhasan[c]

[a]
Assistant Professor, Department of EEE, PGP College of Engineering and Technology, Tamilnadu, India.
[b]Professor, Department of EEE, Gnanamani College of Technology, Tamilnadu, India.
[c]Assistant Professor, Department of ECE, Cheran College of Engineering, Tamilnadu, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: In a random testing environment, a significant amount of energy is wasted in the LFSR and in the CUT by useless patterns that do not contribute to fault dropping. Another major source of energy drainage is the loss due to random switching activity in the CUT and in the scan path between applications of two successive vectors. In this work, a new built-in self-test (BIST) scheme for scan-based circuits is proposed for reducing such energy consumption. A mapping logic is designed which modifies the state transitions of the LFSR such that only the useful vectors are generated according to a desired sequence. Further, it reduces test application time without affecting fault coverage. Experimental results on circuits reveal a significant amount of energy savings in the LFSR during random testing.

Keywords: Logic Built-In Self-Test, Power-On Self-Test, Automatic Test Equipment, Switching Activity.

1. Introduction

Every IC in the industry follows Moore’s law. According to Moore’s law, number of transistors (transistor density) in an IC doubles in every 1.5 years. With the recent advances in the technology, device shrinks to nanometer scale, but density and complexity of the ICs keep on increasing. This may result in many manufacturing faults and device failure. To accommodate more number of transistors, the device feature size is reduced. Reduction in the feature sizes results in increasing the manufacturing faults and fault detection becomes very difficult. VLSI testing is becoming more and more important and challenging to verify whether a device functions properly or not. Conventional automatic test equipment (ATE) based testing method is no longer able to handle the ever-growing test challenges. Logic built-in self-test (LBIST) is widely being adopted as the testing technique for most current day scan based designs. Logic BIST does not alter the scan structure of the designs permitting them to have both ATE based testing and also Logic BIST [1]-[6].

BIST for random logic is becoming an eye-catching substitute in IC testing, although logic BIST is a recent subject which is under research over more than 3 decades. This project provides the use of a deterministic logic BIST structure up on state-of-the-art industrial circuits. Nevertheless, new innovations throughout deep-submicron IC process engineering as well as core-based IC design and design engineering will surely lead to more popular using logical BIST due to the fact outer assessment actually becoming a lot more difficult as well as high- priced. Logic built-in self-test (BIST) depend on the fundamental design for test methodology [7]-[12].

The VLSI circuits are tested using BIST technique which avoids the requirement of external testing equipment. This method achieves simultaneous testing of the circuits under online mode. Various types of 6T SRAM cell layout architectures and corresponding 16-bit arrays have been implemented and compared at the 32 nm, in terms of area, power dissipation and read/write delay. The thin cell topology has proved to be the best design on all aspects. The recently proposed ultrathin cell provides a more lithographically friendly alternative to the thin cell but introduces a significant penalty in area and power/delay performance, presenting overall worse results than the conventional designs [13]-[16].

The impact of global and local variations and NBTI on 6T SRAM core cell are not negligible, but still do not dramatically impact the SRAM design. This is because the two pMOS pullup transistors are the only ones to be affected by NBTI. The situation including PBTI is much worse: this effect works on the nMOS pull-down as well as the access transistors, and the sensitivity especially of the pull-down transistor is much higher: with the same threshold voltage drift, the impact on read stability is twice as high for pull-down compared to pull-up devices. Moreover, during hold state of the memory, the two effects are adding, so that the cell weakens drastically in terms of read stability. The cell loses approx. 30% for 100mV NBTI and PBTI induced threshold voltage drift on pull-up and pull-down transistor. Here is the biggest threat of undesired write during a read cycle, which is equal to data loss. BTI behavior like distribution and annealing also plays an important role, especially if a long hold is directly followed by a read cycle, because then annealing is not possible [17]-[20].

The aim of proposed project is to design the logic module with SRAM cells to store input test vectors and to reduce the switching activity with reduced testing time and concurrent test latency. The proposed scheme is suitable for all types of IC’s. Input vector monitoring concurrent BIST method performs testing during the normal operation of the circuit without imposing a need to set the circuit into offline to perform the testing process, therefore they can have some problems appearing in offline BIST techniques like delay in testing process and more power consumption. The evaluation criteria are the hardware overhead and the CTL, i.e., the time required for the test to complete, while the circuit operates in normal mode. A concurrent BIST architecture for online testing based on the use of a SRAM-cell like structure for storing the data whether an input vector has appeared or not during normal operation of the circuit.

The proposed scheme is more efficient than previous method of input vector monitoring concurrent BIST techniques in testing of VLSI circuits. The research presents a simulation workflow to estimate lifetime for a variety of wear out mechanisms, including negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), gate oxide breakdown (GOBD), hot carrier injection (HCI), backend time dependent dielectric breakdown (BTDDB), electro migration (EM), and stress-induced voiding. Taking into account the detailed thermal and electrical stress profiles of microprocessor systems while running real-world applications, a methodology is developed to accurately assess microprocessor lifetime based on each wear out 20 mechanisms. In addition, this research presents a way to establish the link between the device-level wear out models, the electrical stress profile, the thermal profile, and system performances for both logic and memory blocks. For BTDDB, the impact of line ends was studied and found to be clearly significant.

These irregular geometries can potentially impact chip lifetime and need to be separately extracted and included in the reliability simulator. The work identified the first block that is likely to fail in a system and takes into account a variety of use scenarios, composed of a fraction of time in operation, a fraction of time in standby, and a fraction of time when the system is off.Since the memory blocks within the microprocessor were found to be more vulnerable than the other units, the research also provide a methodology to analyze memory performance degradation due to the frontend wear out mechanisms with studying DC noise margins in conventional 6T SRAM cells as a function of NBTI, PBTI, HCI and GOBD degradation. This provides insights on memory reliability under realistic use conditions. Present a comparison study of four topologies, designed under the 32 nm rules. Proper layouts are designed and presented with detailed information on transistor sizing and interconnections implementation. Furthermore, simulations demonstrate the performance of each topology regarding area, power dissipation, and read/write delay.

2. Proposed System

With the emergence of mobile computing and communication devices, design of low-energy VLSI systems has become a major concern in circuit synthesis. A significant component of the energy consumed in CMOS circuits is caused by the total amount of switching activity (SA) at various circuit nodes during operation. The energy dissipated at a circuit node is proportional to the total number of 0 → 1 and 1 → 0 transitions the logic signals undergo at that node multiplied by its capacitance, which depends on its fan-out and its transistor implementation. Energy consumption in an IC may be significantly higher during testing due to increased SA than that needed during normal (system) mode, which can cause excessive heating and degrade circuit reliability. The average-power optimization help extend the battery life in mobile applications. The maximum-power, sustained or instantaneous, may cause excessive heating or undesirable logic swing.

Conventional BIST schemes with random patterns may need an excessive amount of energy because of the test length and randomness of the consecutive test vectors. Further, a significant amount of energy may be wasted during just the scan operations. Built-In-Self-Test is used to make faster, less-expensive integrated circuit manufacturing tests. The IC has a function that verifies all or a portion of the internal functionality of the IC. In some cases, this is valuable to customers, as well. For example, a BIST mechanism is provided in advanced field-bus systems to verify functionality. At a high level this can be viewed similar to the PC BIOS's power-on self-test (POST) that performs a self-test of the RAM and buses on power-up.

Assume a test-per-scan BIST scheme as in the STUMPS architecture is shown in Figure. 1. A modulo-m bit counter keeps track of the number of scan shifts, where m is the 41 length of the longest scan path. Since the number of useful patterns is known to be a very small fraction of all generated patterns, a significant amount of energy is still wasted in the LFSR while cycling through these useless patterns even though they are blocked at the inputs to the CUT. Further, test-vector reordering in a pseudorandom testing environment is a challenging task. In this paper, propose a new BIST design that prevents the LFSR from cycling through the states generating useless patterns, as well as reorders the useful test patterns in a desired sequence to minimize total energy demand. To estimate energy loss, we compute the total SA as the number of 0 → 1 and 1 → 0 transitions in all the circuit nodes including the LFSR, CUT, and the scan path over a complete test session.

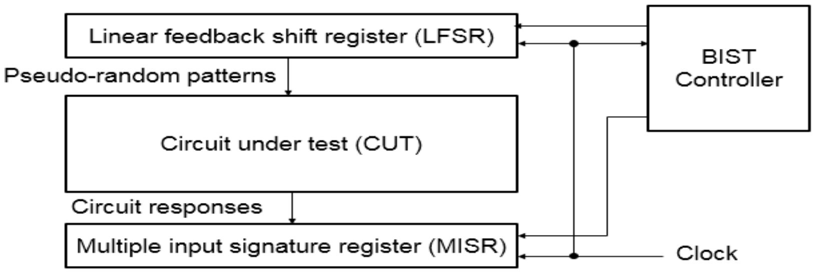

Figure. 1. BIST Architecture

The various steps of the proposed method are now summarized below:

i. A pseudorandom test sequence is generated by an LFSR, and its single stuck-at fault coverage in the CUT is determined through forward and reverse fault simulation; let S denote the test sequence up to the last useful vector (beyond which fault coverage does not improve significantly).
ii. Identify the set U of useless patterns in S that do not contribute to fault dropping.
iii. For all ordered pairs of test vectors in the reduced set Sr=(S\U), determine the switching activity (SA) in the scan path and the CUT.
iv. Reorder the vectors in Sr to estimate an optimal order so to minimize energy.
v. Modify the state table of the LFSR such that it generates the new sequence Sǫ.
vi. Synthesize a mapping logic (ML) with minimum cost, to augment the LFSR; the state transitions of the LFSR are modified under certain conditions to serve two purposes: (a) to prevent it from cycling through the states generating useless patterns and (b) to reorder Sr to Sǫ; for all other conditions, the LFSR runs in accordance to its original state transition function. Figure.2 shows a simple MUX-based-design that can be used for this purpose. A similar idea of skipping LFSR states is used earlier for embedding a set of deterministic tests.

The following example of a TPG illustrates the idea of states kipping technique. The LSB of the LFSR is shifted serially into the scan path generating a test sequence S. Some component of SA is intrinsic (invariant over a full test session), and the rest is variable. Hence, SA can be represented as a directed complete graph called activity graph (Fig. 4b), where each node represents a test vector, and the directed edge (eij) represents application of the ordered test pair (ti, tj). The weight w(eij) on the edge eij denotes the variable component of SA corresponding to the ordered pair of tests (ti, tj). The edge weights are represented as an asymmetric cost matrix, as the variable component of SA strongly depends on ordering of test pairs. Thus, for the test sequence S (t1→ t2 → t3 → t4 → t5), the variable component of switching activity is 37. Now, if t3 is found to be a useless test pattern, it along with all incident edges, can be deleted. An optimal ordering of test vectors that minimizes the energy consumption is a min-cost Hamiltonian path: Sǫ (t1→t2 → t5 → t4), the path cost being equal to 23.

Thus, in the new sequence Sǫ, for the ordered pair (t1 → t2), no action is required, as t2 is generated by the LFSR as a natural successor of t1. So, for s9 (end state of t1), we set the Y-outputs of the mapping logic (see Fig. 2) to don't cares (d), and the control line C to 0. However, we need an additional transition from s14 (end-state of t2) to s8 (start-state of t5), and similarly from s11 (end-state of t5) to s6 44 (start-state of t4). For these combinations, the Y outputs are determined by the corresponding start states, and C is set to 1. For all other remaining combinations, all outputs are not cares. These transitions generate the useful test patterns in a desired sequence, and prevent the LFSR to cycle through the states that generate useless patterns (in this example, test t3). Further, the output M of the modulo-m bit counter assumes 1 only when scan path (whose length is m) is filled, i.e., at the end-states of the test vectors. Thus, in order to generate the sequence so, we need to skip the natural next state of the LFSR and jump to the start state of the desired next test pattern.

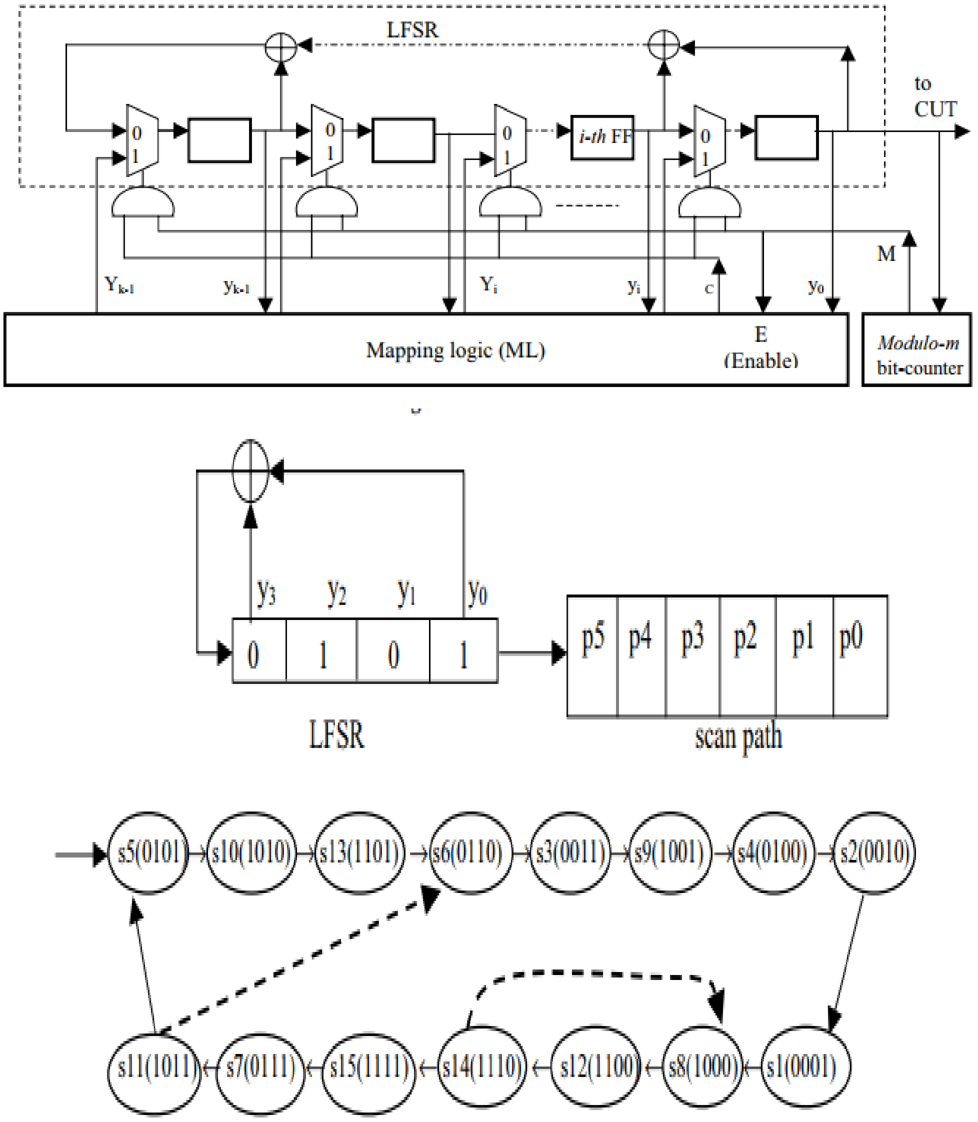

Figure: 2 Example LFSR and Its State Diagram

These state skipping transitions are shown with dotted lines in Figure. 3. In general, the mapping logic can be described as follows: given a seed, let S denote the original test sequence generated by the LFSR, and Sϱ = {t'1, t'2, … t'i, t'i +1, …} denote the optimally ordered reduced test sequence consisting of useful vectors only. Let yi denote the output of the i-th flip-flop of the LFSR, and Yi denote the output of the mapping logic feeding the i-th flip-flop through a MUX seen in Figure. 2. The ML is a combinational circuit with k inputs {y0, y1, …,yk-1}, and k+1 outputs {Y0, Y1, ..,Yk-1, C}, where k is the length of the LFSR, and C is a control output. For every test t'i in sϱ, there is a corresponding row in the truth table case (i) is applicable if the consecutive test pair (t'i, t'i +1) of Sϱ appears in consecutive order in the original sequence S as well; otherwise, case (ii) is applicable. Thus, the next-state of the LFSR follows the transition diagram of the original LFSR when either C = 0, or M = 0, and is determined by the outputs of the mapping logic if and only if CM = 1. Since these additional transitions emanate only from the end states of test patterns, their occurrences can be signaled by the M output, and also when C = 1. In order to prevent the SA from occurring in ML for every scan shift cycle, an enable signal E controlled by M is used. Thus, the y-inputs become visible to ML if and 45 only if M = 1. The test session terminates when the end-state of the last useful pattern in Sϱ is reached. Determination of optimal reordering of test patterns is equivalent to solving a traveling salesman problem (TSP), which being NP-hard, needs heuristic techniques for quick solution.

3. Results And Discussion

The masking properties of signature analyzers depend widely on their structure, which can be expressed algebraically by properties of their characteristic polynomials and the results and analysis are shown in Figure.3. There are three main ways of measuring the masking properties of ORAs:

a. General masking results either expressed by the characteristic polynomial or in terms of other LFSR properties;
b. Quantitative results, mostly expressed by computations or estimations of error probabilities;
c. Qualitative results, e.g. concerning the general possibility or impossibility of LFSR to mask special types of error sequences.

The first one includes more general masking results, which are based either on the characteristic polynomial or on other ORA properties. The simulation of the circuit and the compression technique to determine which faults are detected can 53 achieve. This method is computationally expensive because it involves exhaustive simulation. Smith's theorem states the same point as: Any error sequence E=(e1,...,et) is masked by an ORA S if and only if its "error polynomial" pE(x) = e1xt-1+...+et-1x+et is divisible by the characteristic polynomial pS(x).

The second direction in masking studies, which is represented in most of the papers concerning masking problems, can be characterized by "quantitative" results mostly expressed by some computations or estimations of masking probabilities. This is usually not possible and all possible outputs are assumed to be equally probable. But this assumption does not allow one to correlate the probability of obtaining an erroneous signature with fault coverage and hence leads to a rather low estimation of faults. If we suppose that all error sequences having any fixed length are equally likely the masking probability of any n-stage ORA is not greater than 2-n.

The third direction in studies on masking contains "qualitative" results concerning the general possibility or impossibility of ORAs to mask error sequences of some special type. Examples of such a type are burst errors, or sequences with fixed error-sensitive positions. Traditionally, error sequences having some fixed weight are also regarded as such a special type, where the weight w(E) of some binary sequence E is simply its number of ones. Masking properties for such sequences are studied without restriction of their length. In other words, if the ORA S is non-trivial then masking of error sequences having the weight 1 by S is impossible.

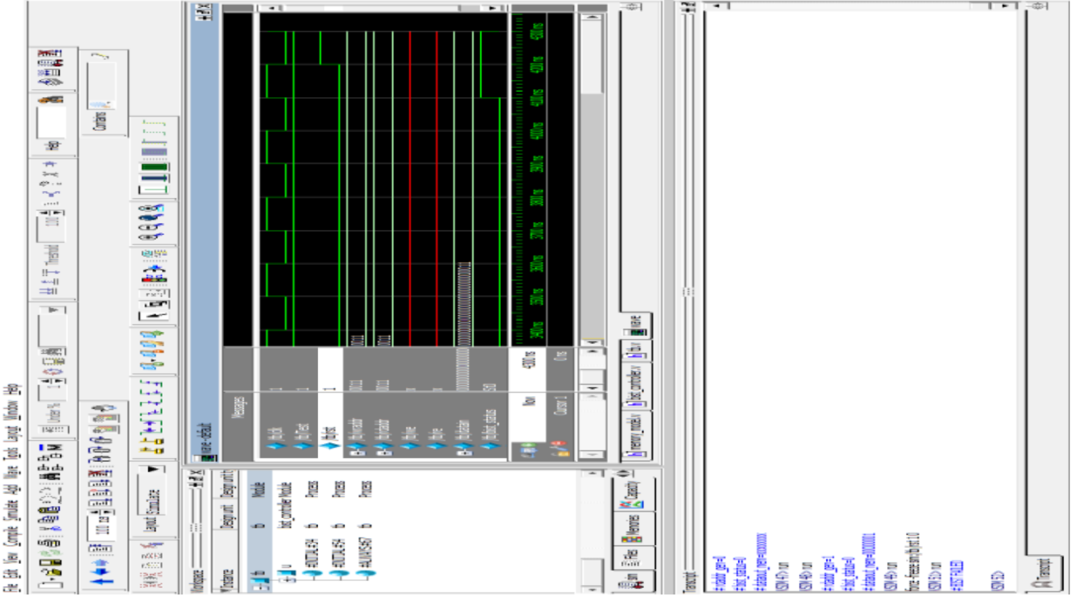

Figure.3. Results Analysis of the Proposed System

4. Conclusıon

The BIST architecture proposed is implemented using Verilog language and tested on various faulty circuits. Then design has been synthesized on fault has been created and simulated on Modelsim. A new BIST design is described for saving energy both in the LFSR and the CUT in a random testing environment. A significant component of the SA is observed to be intrinsic in nature, which given a test set, cannot be reduced by vector reordering. To reduce this component, either a different set of useful test vectors is to be selected from the random sequence, or the scan path architecture is to be radically redesigned. Ensuring reusability of mapping logic and BIST hardware for different cores on a chip is another open area to study. In this paper we have illustrated an implementation of BIST logic using Verilog. LFSR is used as a pseudorandom sequence generator. Signature analysis is used to make verification of the circuit. Signature mismatch with the reference signature means that the circuit is faulty. However, there is a small probability that the signature of a bad circuit will be the same as a good circuit. When longer sequences are used, signature analysis gives high fault coverage.

References

1. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability", Proc. of DAC, pp. 462-468, 1977.
2. V. D. Agrawal, C. R. Kime and K. K. Saluja, "A Tutorial on Buil-In Self-Test, Part 1: Principles", IEEE Design and Test of Computers, Vol. 10, Issue 1, pp. 73-82, March 1993.

A. Jas, C.V. Krishna, and N.A. Touba, “Weighted pseudorandom hybrid BIST,” IEEE Trans. VLSI, vol. 12, no. 12, pp. 1277-1283, Dec. 2004.

3. R. Kapur, S. Patil, T.J. Snethen, and T.W. Williams, “A weighted random pattern test generation system,” IEEE Trans. CAD, vol. 15, no. 8, pp. 1020-1025, Aug. 1996.
4. K.-H. Tsai, J. Rajski, and M. Marek-Sadowska, “Star test: the theory and its applications,” IEEE Trans. CAD, vol. 19, no. 9, pp. 1052-1064, Sept. 2000.
5. Pomeranz and S.M. Reddy, “Static test data volume reduction using complementation or modulo-M addition,” IEEE Trans. VLSI, vol. 19, no. 6, pp. 1108-1112, June 2011.
6. J. Rajski, J. Tyszer, M. Kassab and N. Mukherjee, “Embedded Deterministic Test", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 23, Issue 5, pp. 776-792, May 2004.
7. M. Abramovici, M. Breuer, and A. Friedman, “Digital Systems Testing and Testable Design”, IEEE Press, Piscataway, NJ, 1994.
8. R. D. Eldred, “Test Routines Based on Symbolic Logical Statements", Journal of the ACM, Vol. 6, pp. 33-36, 1959.
9. S.M. Reddy, “Test Drivers – Past, Present, and Future”, VTS 2018 Invited Keynote.
10. Z. Chen, D. Xiang, and B. Yin, (May. 2009) “The ATPG conflict-driven scheme for high transition fault coverage and low test cost,” in Proc. 27th IEEE VLSI Test Symp. pp. 146–151.
11. Z. Chen and D. Xiang, (May. 2010) “Low-capture-power at-speed testing using partial launch-on-capture test scheme,” in Proc. 28th IEEE VLSI Test Symp. pp. 141–146.
12. Z. Chen, K. Chakrabarty, and D. Xiang, (Nov. 2010) “MVP: Capture-power reduction with minimum-violations partitioning for delay testing,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, pp. 149–154.
13. M. Filipek et al., (Jun. 2015) “Low-power programmable PRPG with test compression capabilities,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1063–1076.
14. S. Gerstendörfer and H.-J. Wunderlich, (Jun. 1999) “Minimized power consumption for scan-based BIST,” J. Electron. Test., vol. 16, no. 3, pp. 203–212.
15. S Kannadhasan, M Shanmuganantham and R Nagarajan, ‘’System Model of VANET using Optimization-Based Efficient Routing Algorithm’’ IOP Conf. Series: Materials Science and Engineering, 1119 (2021) 012021, doi:10.1088/1757-899X/1119/1/012021
16. P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, (Oct. 2000) “Low power BIST design by hypergraph partitioning: Methodology and architectures,” in Proc. Int. Test Conf.pp. 652–661.
17. S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, (Feb. 1995)“Built-in test for circuits with scan based on reseeding of multiplepolynomial linear feedback shift registers,” IEEE Trans. Comput., vol. 44, no. 2, pp. 223–233.
18. S. Hellebrand, H.-G. Liang, and H.-J. Wunderlich, (Oct. 2000) “A mixed mode BIST scheme based on reseeding of folding counters,” in Proc. Int. Test Conf. pp. 778–784.
19. Y. Huang, I. Pomeranz, S. M. Reddy, and J. Rajski, (Nov. 2000) “Improving the proportion of at-speed tests in scan BIST,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, pp. 459–463