Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

LFSR BASED TEST PATTERN GENERATOR FOR COMPLEX CIRCUITS

Mrs. N. MALATHI M. Tech, Assistant Professor, Department of ECE, NRI Institute of Technology
P. SIVA VEDA SRI, S.MONIKA, Y. RAKESH SHIVA SATYANARAYANA, N.S.S.S.PAVAN
B. Tech, Student, Department of ECE, NRI Institute of Technology

ABSTRACT

Testing of the circuits plays a crucial role for maintaining the better yield ratio and to minimize the loses. Apart from the test strategy test time is also considered to be the important parameter in the overall process. Among the most common ways of testing the circuit Built-in-Self-Test (BIST) is commonly used due to its efficiency which contains three different blocks, Random Number Generation, Circuit Under Test, Comparator. To test the circuit in different ways under different inputs we mainly concentrate on Random Number Generation where the commonly used circuit is Linear Feedback Shift Register (LFSR). LFSR is used to generate Pseudo random sequences of bits which can be used in testing a logical circuit. The main objective of this work is to increase the length of the Pseudo random sequences generated by a Test Pattern Generator by combining internal and external LFSR using a control signal in a single module. In order to obtain better results when compared with existing , a hybrid LFSR was designed which improves fault coverage with less number of bit patterns so that, the long test patterns can be generated by using lower bit LFSR.

Keywords:
VLSI, LFSR, BIST, CUT, Testing, fault coverage, Pseudo Random Sequences.

I. INTRODUCTION

The emergence of 3-D stacked ICs helps us to integrate multiple silicon dies as a stack using Through Silicon Vias (TSVs). In contrast with traditional stacking the 3-D stacked designs involve less power consumption, decrease in interconnect area, decreased wire load and high percentage of stacking density. Now a days the use of 3-D Stacking, such as 3-D memory-on-processor, 3-D FPGA , 3-D NoC , and 3-D logic Stacks as also been predicted. VLSI chip tests can be performed in a few different types with different environments.

If a new chip is designed and made before it reaches the customer, we need to ensure its accuracy design and testing process. IC testing plays an important role in ensuring that the circuit has no errors. There are many ways to test a circuit and it has also been improved over time. There are different types of testing such as verification test, production test, acceptance test etc. Test-Based IC-3-D test where several circuits are set as background and stacked one over the other using Through-Silicon-Vias (TS V's) before. But because of this we get too many irregularities in the cycle which can lead to many errors and can lead to cord lengths problem. So instead, we can connect them using Controller schemes to connect the layers.

When we use this type of pf program, we cannot use additional hardware circuitry as it may cause a larger surface area. So, we need to find a test hardware that can be built in a circuit. Among them is the BIST is another, widely used. BIST stands for Built-In-Self-Test which is used test the circuit accordingly. Computer hardware resulting from BIST additions may be empty and may be tested to a large extent to several times where necessary.

It even reduces the cost of testing and testing time.

BIST(Built-in-Self-Test)

In VLSI Design testing plays a prominent role. The correctness of IC behavior can be assured from the Testing phase hence it plays a crucial role. Testing in VLSI refers to testing of these designs such that they are fabricated correctly and designed correctly. With increase in number of levels in 3D Stacked ICs the testing cost for these designs increases, hence need an architecture which can minimizes the test cost and increases the functionality. With Built-in-self-Test (BIST) framework which is a design technique where a part of a circuit can be used to test the circuit by itself with minimal hardware and less cost.

The BIST framework consists of Pseudo- Random Pattern Generator, Circuit under Test (CUT), Response Analyzer and a BIST Controller to control the flow in the circuit. BIST is a design technique that allows a circuit to test itself. In this project the test performance achieved with the implementation of BIST is proven to be adequate to offset the disincentive of the hardware overhead produced by the additional BIST circuitry. The technique can provide shorter test time compared to an externally applied test.

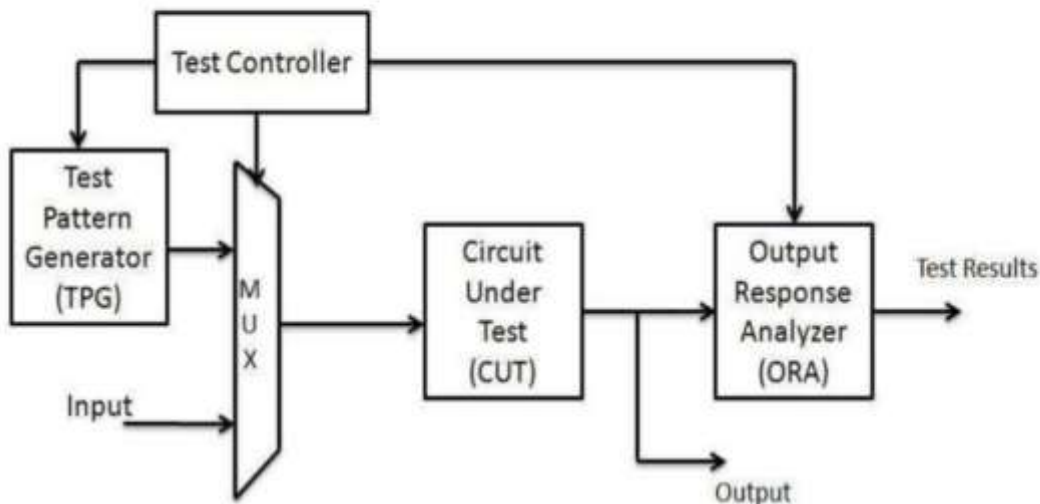

Fig1. BIST Framework

Linear Feedback Shift Registers

Among the available pattern generators used Linear Feedback Shift Register (LFSR) is most used for pseudo-random pattern generation. These patterns have random numbers that are generated algorithmically through a characteristic polynomial. LFSR is a shift register whose present input is a linear function of previous state by an XOR operation between states. In computing, a linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its prior state. The most used linear function of single bits is the special or (XOR) function. Thus, an LFSR is often a shift register driven by the XOR of some bits of the input bit overall shift register value.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

The initial value of the LFSR is called the seed, and since the operation of the record is deterministic, the stream of values produced by the record is entirely determined by its current (or previous) state. Similarly, since the register has a finite number of possible states, it must eventually enter an iterative loop. However, an LFSR with a well-chosen feedback function can produce a series of bits that seem random and have a very long loop.

The mathematics of a cyclic redundancy check used to provide a quick check for transmission errors are closely related to those of an LFSR.[1] Overall, the arithmetic behind LFSRs makes them very elegant as an object to study and implement. Relatively complex logics can be produced with simple building blocks. However, other less elegant but better performing methods should also be considered.

The rightmost bit string is called the output stream. Bits in the LFSR state that affect the input are called taps. A max-length LFSR produces an m-sequence (i.e. it cycles through all possible 2m - 1 states in the shift register as long as it doesn't contain all zeros), in which case it never changes. As an alternative to XOR-based feedback in an LFSR, XNOR can also be used.[2] This function is an affine map, not strictly a linear map, but results in an equivalent polynomial counter whose state is the complement of the state of an LFSR. An all-zero condition is illegal when using XNOR feedback, just as an all- zero condition is illegal when using XOR. This is considered illegal as the meter will remain "locked" in this state. This method can be advantageous in hardware LFSRs using flip-flops that start in a zero state as it does not start in a lock state, i.e. the register does not need to be seeded to start operating.

The number sequence produced by an LFSR or XNOR equivalent can be considered a binary number system as valid as the gray code or natural binary code. The arrangement of steps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod 2. This means that the coefficients of the polynomial must be 1s or 0s. This is called the feedback polynomial or the mutual characteristic polynomial. For example, if the taps are at bits 16, 14, 13, and 11 (as shown), the feedback polynomial x^{16}+x^{14}+x^{13}+x^{11}+1 The "one" in the polynomial does not correspond to a step - it corresponds to the input of the first bit (i.e., x0, which is equivalent to 1). The powers of the terms, counting from the left, represent the clicked bits. The first and last bits are always connected as an input and output tap respectively. A Fibonacci 31-bit linear feedback shift register has steps at positions 28 and 31, giving this speed a maximum cycle and duration of about 6.7 years.

The LFSR is of maximum length if and only if the corresponding feedback polynomial is primitive. This means that the following conditions are necessary (but not sufficient): The number of taps is equal. The set of taps is co-prime; that is, there should be no divisors other than 1 common to all taps. An LFSR in a Galois configuration, also known as modular, internal XORs or one-to-many LFSRs, is an alternative structure that can produce the same output stream (but offset over time) as a conventional LFSR.[3] In the Galois configuration, the non-tap bits are shifted one position to the right when the system is clocked. Taps, on the other hand, are XORed with the output bit before being stored in the next location. The new output bit is the next input bit. The effect of this is that when the output bit is zero, all the bits in the register are left shifted unchanged and the input bit is zero. When the output bit is one, all the bits in the stage are inverted (0 becomes 1 and 1 becomes 0) and then the entire register is shifted to the right and the input bit becomes 1.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

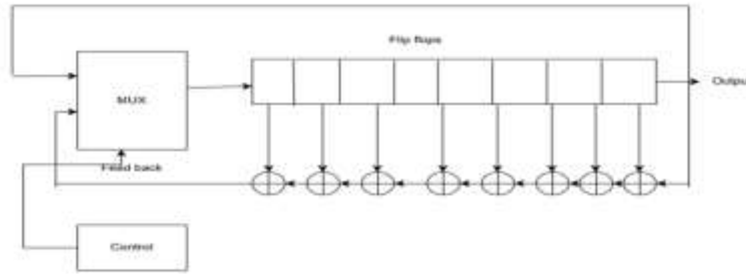

Fig 2. Internal LFSR for function F(x) = 1+x+x^3+x^4

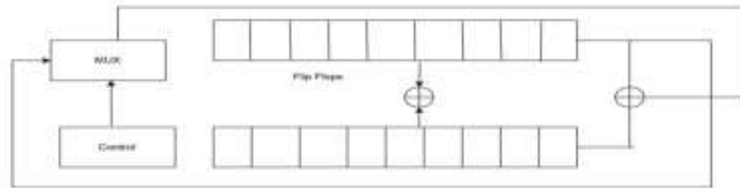

Fig3. External LFSR for function F(x) = 1+x+x^3+x^4

II. LITERATURE SURVEY

Faults may occur due to failure in the implementation of the design. It explains the various faults that are caused during manufacturing process due to improper placement of photo resist layer. Among all other kinds of faults Stuck-at-1 and Stuck-at-0 are typical faults that need to detect and removed for the proper functioning of the chip. The prior work on 3-D Stacked ICs is based on Through Silicon Vias (TS V's) which include the analog layout structure of designing 3-D Stacks from interconnects. In Xin Zhao et.al showed the way to partition the TSV using clock partitioning algorithm by reducing the wire length and clock power consumption used. In B.Noia et.al proposed an new Test Access Mechanism (TAM) to transport test data on to cores, they have divide dies on to 3-types such as has, soft and firm dies to decrease the wire length used in each of the module. In Rajit Karmakar et.al describes the way to test the 3-D architecture considering power and time constraints. He has defined the scheduling strategy which could schedule the constraints based on power has the radiated power need to be less for perfect functioning of the system. In S. Deutsch et.al proposed an ILP model to robust the optimization which takes in to account of variations input parameter and provides an solution to reduce test time.

In [4] shows the way to generate LFSR polynomials to get Maximum number of test patterns for the circuit by considering the period and shows the maximum number of primitive polynomials that has to be used to get test patterns. In [3] shows the gate level description of the different gates by converting them into sequential circuits by adding D-flip-flops.

In [5] Bogdan Dugonik et.al proposed to select the LFSR Test Pattern Generator seed to achieve highest fault coverage from given set of different characteristic polynomials. In the application of seed selection in combination with phase shifter to evaluate the fault coverage for different circuits. This also shows the construction of LFSR polynomial by having multiple taps at specified spaces which could yield maximum possible test vectors for testing the circuit

Valentin Muregan et. [6] proposed a way to improve test concurrency with power dissipation limits using extended tree growing algorithm by considering sequence of list and distributed graph. In Amit Kumar et.al proposed an hyper graph partition tool to connect the interconnects between the layers to reduce the wire length between them. In Mukesh Agrawal et.al shows that the minimization of Wire Connect is equivalent to graph-theoretic and have proposed a technique for reuse-based method at post bond testing.

The BIST scheduling approach given in [7] is one of the first to consider the power dissipation during test scheduling at block level. It performs global optimization considering also other factors such as block type, adjacency of blocks (device floor plan), but the latter are hardly known at high-level. [8] makes for the first-time a thorough theoretical analysis of the power-constrained test scheduling (PTS) problem at IC level. It proposes a compatible test clustering technique which is an NP-complete approach.

Linear Feedback Shift Register is a shift register whose input is a linear function of previous state. The linear function thus obtained by xor operation between states. Linear Feedback Shift Register are of mainly two types External and Internal Shift Register's in a technique to generate seeds from inverting the logic value of some bits from its next state by eliminating the ROM from storing seeds which results in fast computation of seeds. In a way to implement test patterns from limiting the power in the circuit by building a controller module. In Dhanesh P et.al prosed a Dual Threshold Bit Swapping LFSR to reduce the transition power between vectors by considering different threshold values while swapping the bits. In LFSR generating polynomial that could increase the fault coverage by eliminating the total number of invalid patterns generated due to LFSR.

.

III. OVERVIEW OF PROPOSED SYSTEM

During BIST operations the patterns are produced by a pseudo-random generator will not be enough to produce a high number of errors, they can find errors until a certain set of other patterns may not produce errors. Below under such cases we can use reseeding techniques so that the random generator starts to change from that point re-inserted can therefore produce a higher amount of coverage for the limit number of patterns. For example, suppose CUT has 4 inputs so we use 4-bit LFSR too the first seed as 1101. Let the Polynomial aspect of LFSR be $x4 + x + 1$. Next, he says of LFSR taken from the polynomial work and sequence represented in Consider the vectors 0x01, 1000 and 1111 required for solid errors. So, we take a closer look that in addition to reorganization, the first 13 vectors are required for circuit inspection. During the generation the 4th vector includes a central pattern; error 0x01 can be checked.

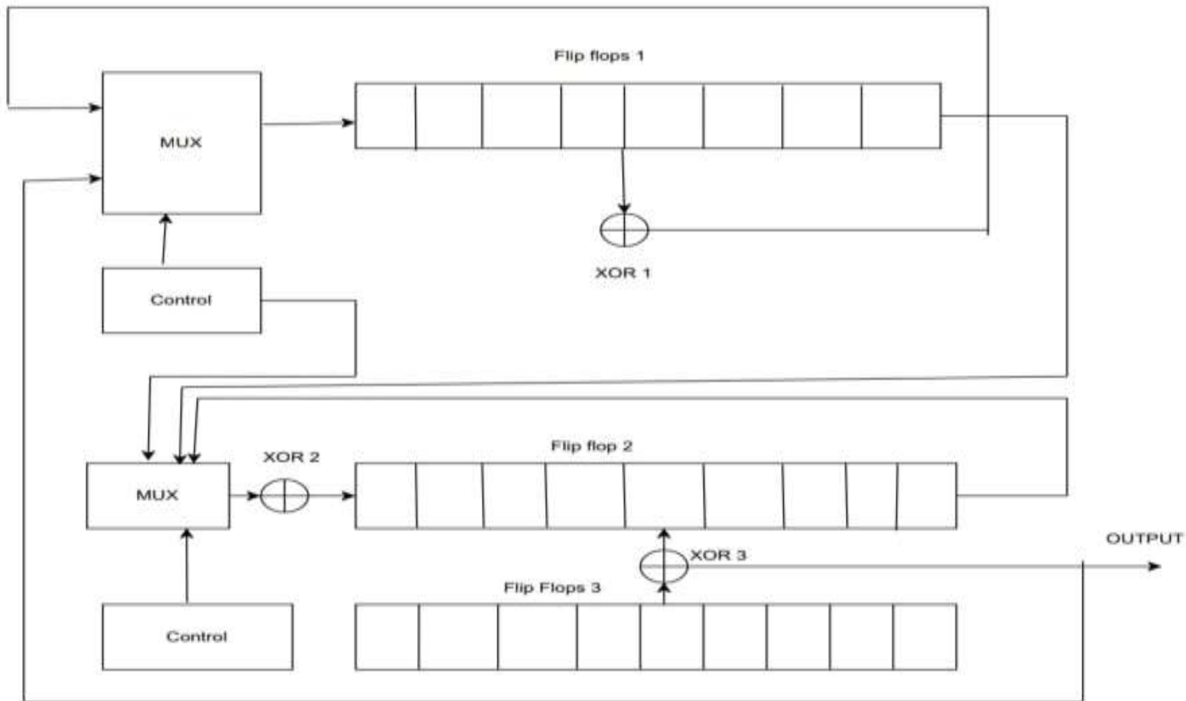

Fig 3. Proposed Block Diagram

The counter will count to the required value. The user wants to create test patterns. For example, for what the counter that generates test patterns for n-bit inputs should not count value up to 2n-1 based on max length formulas used to automatically generate continuous seed production fed into LFSR to generate the required pseudo-random test patterns will be supplied to the tested circuit output. The CUT will be compared to the correct output tested circuits stored in response analyzer called gold signature using signature analysis.

Used in the system to produce counters seed value. The seed value produced must be constant for the 2n-1 clock cycle. circuit in the proposed system It works up to 32 bits. The system can accept input Value up to 32 bits. That is, the counter must have a value Stable for 232-1 clock cycle count. counter 1 and counter 2 is compressed into a single block called counter counter 2 to stabilize the counter value, Used.

The counter will work according to 2 clock cycles will take increments for each clock cycle. after counter 2 reaches the required value. Counter 1 will be updated. Counter 1 value is given to LFSR reseeded as seed value for generating Pseudo-Random Test Patterns for specific seed value. The counter starts counting 1. Value from 0000 for 4-bit input. For 0000 seed value the LFSR will produce all 0 test patterns. this test Patterns will not be used to test the CUT because a lock status. The LFSR consists of a primitive polynomial. It will help to create all possible combinations with according to seed value. Seed value given to LFSR taken from the counter 1.

According to the given seed value test patterns are created. Test patterns are provided to the CUT to test the design. From the new seed the LFSR operation will be started again this solves the issue of repeated or regenerative patterns so it can generate more sufficiently valued patterns with lesser clock cycles. We are providing the enable signal to initiate the signal and that will be given to the counter (N bit).

The design is employed to check circuits uses less storage and time design that includes LFSR and counter. Counter can count until the specified worth that the user needs to get look at patterns. The output of the CUT is going to be compared with the output of the properly tested circuits that are kept in response analyzer referred

to as a golden signature uses signature analysis.

The LFSR circuits which were used in BIST circuits that are initially implemented in Model sim and the outputs from these LFSR's are given to the hope simulator to check the fault coverage and efficiency of the circuit. The LFSR's thus chosen is given to the BIST architecture and analysed the performance using comparator as an Output Response Analyzer and operation of it is monitored.

IV. RESULTS & ANALYSIS

Output

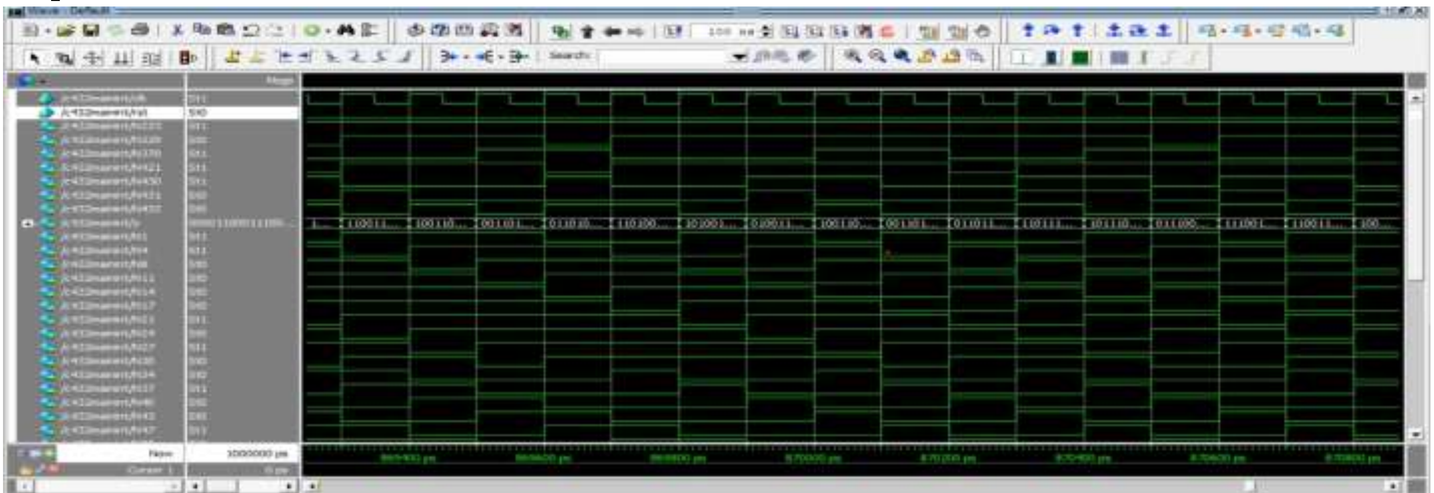

Fig 4. Output for internal LFSR

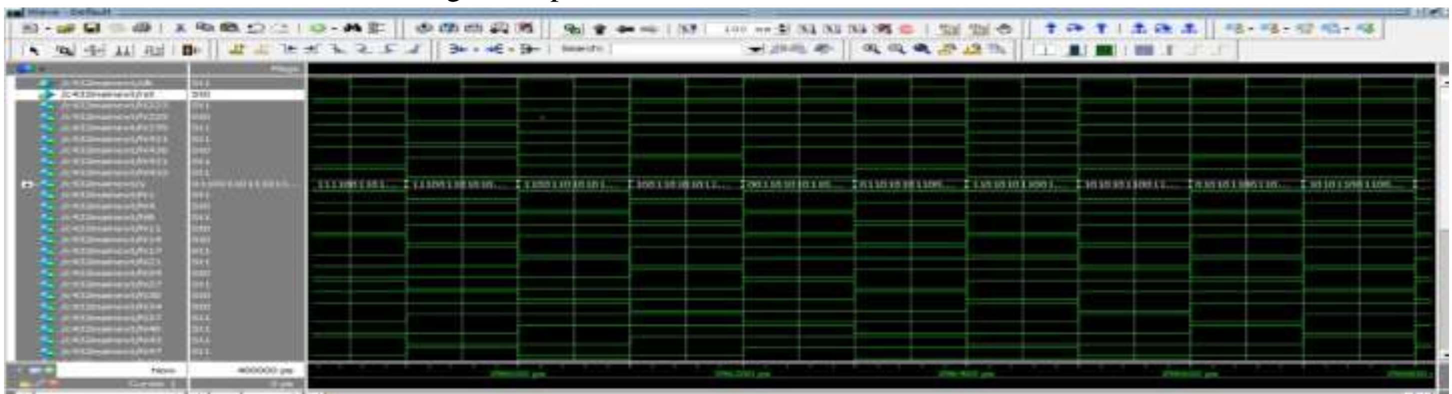

Fig 5. Output for external LFSR

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

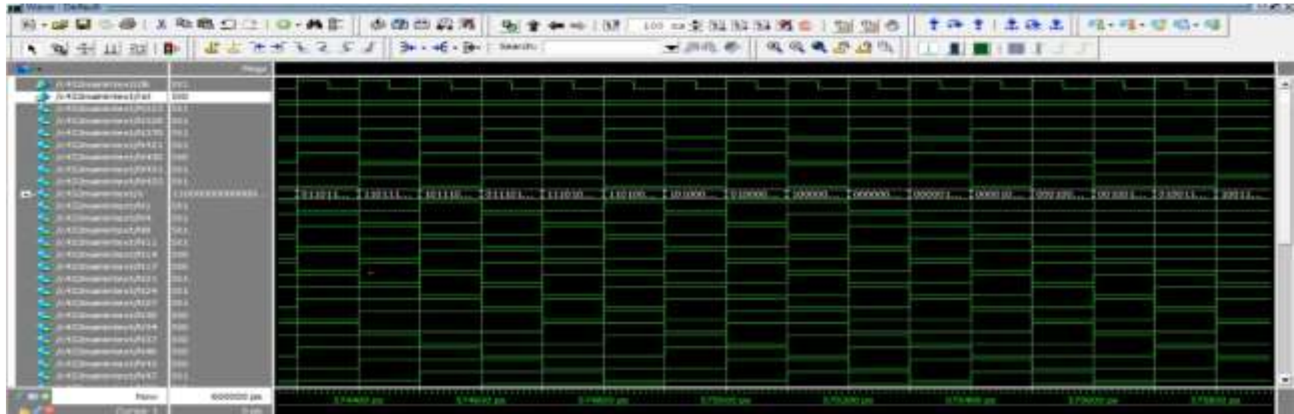

Fig 6. Output of Hybrid LFSR

```
1.  Circuit structure
    Name of circuit                                   : c17
    Number of primary inputs                          : 5
    Number of primary outputs                         : 2
    Number of flip-flops                              : 0
    Number of gates                                   : 6
    Level of the circuit                              : 3

2.  Simulator input parameters
    Simulation mode                                   : file (c17.test)

3.  Simulation results
    Number of test patterns applied                   : 31
    Fault coverage                                    : 100.000 %
    Number of collapsed faults                        : 22
    Number of detected faults                         : 22
    Number of undetected faults                       : 0

4.  Memory used                                       : 13393 Kbytes

5.  CPU time
    Initialization                                    : 23204512.750 secs
    Fault simulation                                  : 0.000 secs
    Total                                             : 23204512.750 secs
```

Fig 7. Output Summary of c17 (ISCAS 85) benchmark circuit

```
1.  Circuit structure
    Name of circuit                                   : 36_inputs
    Number of primary inputs                          : 36
    Number of primary outputs                         : 7
    Number of flip-flops                              : 0
    Number of gates                                   : 160
    Level of the circuit                              : 17

2.  Simulator input parameters
    Simulation mode                                   : file (c432.test)

3.  Simulation results
    Number of test patterns applied                   : 792
    Fault coverage                                    : 99.237 %
    Number of collapsed faults                        : 524
    Number of detected faults                         : 520
    Number of undetected faults                       : 4

4.  Memory used                                       : 16859 Kbytes

5.  CPU time
    Initialization                                    : 23169299.467 secs
    Fault simulation                                  : 0.000 secs
    Total                                             : 23169299.467 secs
```

Fig 8. Output summary of c432 (ISCAS 85) benchmark circuit

```
1.  Circuit structure
    Name of circuit                      : combinational_logic_example_"c1355"
    Number of primary inputs             : 41
    Number of primary outputs            : 32
    Number of flip-flops                 : 0
    Number of gates                      : 546
    Level of the circuit                 : 24

2.  Simulator input parameters
    Simulation mode                      : file (c1355.test)

3.  Simulation results
    Number of test patterns applied      : 1700
    Fault coverage                       : 99.111 %
    Number of collapsed faults           : 1574
    Number of detected faults            : 1560
    Number of undetected faults          : 14

4.  Memory used                          : 12791 Kbytes

5.  CPU time
    Initialization                       : 23139766.133 secs
    Fault simulation                     : 0.533 secs
    Total                                : 23139766.667 secs
```

Figure: Output summary of c1355 (ISCAS 85) benchmark circuit.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 3, March : 2023

```
1.  Circuit structure
    Name of circuit                          : combinational_logic_example_"c3540"
    Number of primary inputs                 : 50
    Number of primary outputs                : 22
    Number of flip-flops                     : 0
    Number of gates                          : 1669
    Level of the circuit                     : 47

2.  Simulator input parameters
    Simulation mode                          : file (c3540.test)

3.  Simulation results
    Number of test patterns applied          : 3099
    Fault coverage                           : 95.741 %
    Number of collapsed faults               : 3428
    Number of detected faults                : 3282
    Number of undetected faults              : 146

4.  Memory used                              : 12750 Kbytes

5.  CPU time
    Initialization                           : 23147170.300 secs
    Fault simulation                         : 1.833 secs
    Total                                    : 23147172.133 secs
```

Figure: Output summary of c3540 (ISCAS 85) benchmark circuit

V. CONCLUSION & FUTURESCOPE

Overall, the design and analysis of a test pattern generator using internal and external LFSRs is an important technique for ensuring the reliability and functionality of digital circuits. By combining both internal and external LFSRs, designers can generate high-quality and efficient test patterns that can help detect and diagnose any faults in the circuit. The analysis of the test pattern generator involves evaluating the quality and efficiency of the test patterns generated. Quality metrics such as the fault coverage, the number of undetected faults, and the detection latency are used to assess the effectiveness of the test pattern generator. Efficiency metrics such as the test application time, the storage requirements, and the power consumption are used to evaluate the performance of the test pattern generator.

The design and analysis of a test pattern generator by combining internal and external Linear Feedback Shift Registers (LFSRs) has significant potential for future research and development. Researchers can explore ways to improve fault coverage by incorporating additional LFSRs or using more advanced algorithms for selecting feedback polynomials. Test pattern compression techniques can be explored to reduce the size of the generated test pattern sequences while maintaining the same fault coverage. The use of LFSRs in test pattern generation can be extended to IoT devices and cybersecurity applications. In these areas, the use of LFSRs can be useful in testing the security and reliability of the system. Integration of Artificial Intelligence (AI) algorithms can be done with LFSR-based test pattern generation to improve the test pattern generation and fault detection process. Future research can explore hybrid techniques that combine LFSRs with other test pattern generation methods such as BIST, SCAN, or ATPG to generate more efficient and reliable test patterns.The use of LFSRs in multi-core testing can be explored, where multiple cores can be tested simultaneously using different LFSRs. Overall, the future scope of design and analysis of a test pattern generator by combining internal and external LFSRs is wide and promising, and continued research in this area can lead to more efficient and reliable testing of digital circuits.

REFERENCES

1. V. Shivakumar , C. Senthilpari and Z. Yusoff, A Low-Power and AreaEfficient Design of a Weighted Pseudorandom Test-Pattern Generator for a Test-Per-Scan Built-in Self-Test Architecture,2021.
2. D. Datta, B. Datta and H. S. Dutta, Design and implementation of multibit LFSR on FPGA to generate pseudorandom sequence number
3. Naveen Balaji.G , Chenthur Pandian.S, Design of test pattern generator (TPG) by an optimized low power design for testability (DFT) for scan BIST circuits using transmission gates

4. P. S. Dilip, G. R. Somanathan and R. Bhakthavatchalu, Comparative Study of Test Pattern Generation Systems to Reduce Test Application Time, 2019.
5. Bogdan Dugonik, Zmago Brezocnik, Seed And Polynomial Selection Algorithm For Lfsr Test Pattern Generator In Built-In Self-Test Environment, Informacije MIDEM 34(2004)3, str. 141-149.
6. A combined tree growing technique for block-test scheduling under power constraints.
7. Y. ZORIAN: A Distributed BIST Control Scheme for Complex VLSI Devices - Proceediiigs of The 11th I€€€ VLSI Test Syi?posirtiii. pp. 4-9, Apr. 1993.
8. R.M. CHOU. K.K. SALUJA. V.D. AGRAWAL: Scheduling Tests for VISI Systems Under Power Constraints - IEEE Ti.nris. oii Very Large Sccrlc hitegratiori (VLSI) S.vsfciiis, Vol. 5. No. 2, pp. 175-185. Jun. 1997.
9. W.B. JONE. C. PAPACHRISTOU, M. PEREIRA: A Scheme for Overlaying Concurrent Testing of VLSI Circuits - Proceedings qf rlic 26rli Desirig Aittoinnriori Confewrice. pp. 531-536. 1989.
10. V. MURESAN. X. WANG. V. MURESAN. M. VLADUTIU: A Coniparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling - I€€€ Tcsi CoilfErcizce (ITC) 2000, pp. 882 - 891. Atlantic City. NJ. USA, Oct, 2000.
11. G.L. CRAIG. C.R. KIME, K.K. SALUJA: Test Scheduling and Control for VLSI Built-In Self-Test - IEEE Trnris. oii Coriipier. Vol. 37. No. 9. pp. 1099-1 109. Sep. 1988.
12. E. LARSSON, Z. PENG: Test Infrastructure Design and Test Scheduling Optimization - P roceediiigs of The IEEE Eirropenri Test Confererice, 2000.