LOW POWER AND HIGH THROUGHPUT CLOCK SPLITTING BASED ATPG FOR BIST

1. S. SAMBASIVA RAO DANNINA
1. Research Scholar, Dept. of ECE, VEL Tech University, Chennai, Tamil Nadu

ABSTRACT:
The design for low power has become one of the greatest challenges in high-performance very large scale integration (VLSI) design. It has been found that the power consumed during test mode operation is often much higher than during normal mode operation. The main objective of this project is to design a low power Test pattern generator. LFSR (Linear Feedback Shift Register) is commonly employed in various BIST, cryptography applications to generate pseudo-random numbers. This concept explores the avenues in power minimization during test application in CMOSVLSI circuits since power consumption during testing is high when compared to normal operation. Design of low transition Test Pattern Generators is one usual method adopted to reduce power consumption. In the Proposed Modified Low Transition Linear Feedback Shift Register, power dissipation during testing is reduced by minimizing the switching activity between successive test vectors by comparing the two consecutive test vectors. Further power is reduced by using clock splitting technique.

KEYWORDS: Built in Self-Test, Automatic Test pattern Generator, Low Transition Low power LFSR, Clock Splitting,

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. Over the last decade, network-on-chip (NoC) has emerged as a better communication infrastructure compared with bus-based

communication network for complex chip designs overcoming the difficulties related to bandwidth, signal integrity, and power dissipation [1]. However, like all other systems-on-a-chip (SoCs), NoC-based SoCs must also be tested for defects. Testing the elements of the NoC infrastructure involves testing routers and interrouter links. Significant amount of area of the NoC data transport medium is occupied by routers, which is predominantly occupied by FIFO buffers and routing logic. Accordingly, the probabilities of run-time faults or defects occurring in buffers and logic are significantly higher compared with the other components of the NoC. Thus, test process for the NoC infrastructure must begin with test of buffers and routing logic of the routers. In addition, the test must be performed periodically to ensure that no fault gets accumulated. The occasional run-time functional faults have been one of the major concerns during testing of deeply scaled CMOS-based memories. These faults are a result of physical effects, such as environmental susceptibility, aging, and low supply voltage and hence are intermittent (nonpermanent indicating device damage or malfunction) in nature [2]. However, these intermittent faults usually exhibit a relatively high occurrence rate and eventually tend to become permanent [2]. Moreover, wear-out of memories also cause intermittent faults to become frequent enough to be classified as permanent. Thus, there is a need for online

test technique that can detect the run-time faults, which are intermittent in nature but gradually become permanent over time. Chip integration has reached a stage where a complete system can be placed in a single chip. When we say complete system, we mean all the required ingredients that make up a specialized kind of application on a single silico n substrate. This integration has been made possible because of the rapid developments in the field of VLSI designs. This is primarily used in embedded systems. Thus, in simple terms a SoCcan be defined as "an IC, designed by stitching together multiple s Tand -alone VLSI designs to provide full functionality for an application." A NoC is perceived as a collection of computational, storage and I/O resources on -chip that are connected with each other via a network of routers or switches instead of being con nected with point to point wires. These resources communicate with each other using data packets that are routed through the networkin the same manner as is done in traditional networks. It is clear from the definition that we need toemployhighly sophis ticated and researched methodologies from traditional computer networks and implement themon chip.we have to explore the motivating factors that are compelling the researchers and designers to move toward the adoption of NoC architectures for future SoCs. The area of NoC is still in its infancy, which is one of the reasons why there are various names for the same thing; some call it on chip networks, some networks on silicon, but the majority agrees upon "Networks on Chips" (NoCs). However, we will be using these terminologies interchangeably throughout our tutorial.NOC is Integrating various processors and on chip memories into a single chip . 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 intro- duced 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].

BUILT IN SELF TEST:

When testing is built into the hardware, it has the potential of being not only fast and efficient but also hierarchical. In other words, in a well-designed testing strategy, the same hardware can test chips, boards, and system. The cost benefits, which may not seem significant at the chip level, are enormous at the system level.

ARCHITECTURE OF BIST:

BIST is basically same as off-line testing using ATE where the test pattern generator and the test response analyzer are on-chip circuitry. As equipments are replaced by circuitry, so it is obvious that compressed implementations of test pattern generator and response analyzer are to be designed. The basic architecture of BIST is shown in Figure 1.

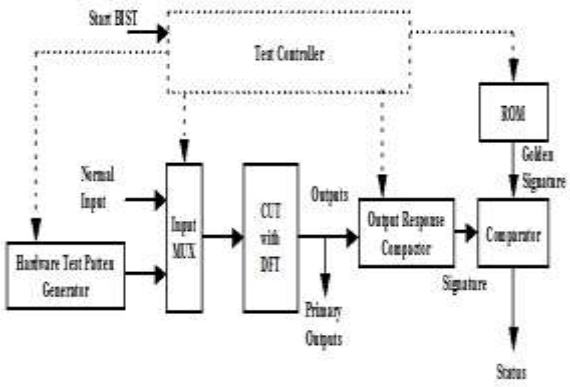


Figure 1. Basic architecture of BIST

As shown in Figure 1, BIST circuitry comprises the following modules (and the following functionalities)

1. *Hardware Test Pattern Generator:* This module generates the test patterns required to sensitize the faults and propagate the effect to the outputs (of the CUT). As the test pattern generator is a circuit (not equipment) its area is limited. So storing and then generating test patterns obtained by ATPG algorithms on the CUT (discussed in Module XI) using the hardware test pattern generator is not feasible. In other words, the test pattern generator cannot be a memory where all test patters obtained by running ATPG algorithms (or random pattern generation algorithms) on the CUT are stored and applied during execution of the BIST. Instead, the test pattern generator is basically a type of register which generates random patterns which act as test patterns. The main emphasis of the register design is to have low area yet generate as many different patterns (from 0 to 2^n , if there are *n* flip-flops in the register) as possible

2. *Input Mux:* This multiplexer is to allow normal inputs to the circuit when it is operational and test inputs from the pattern generator when BIST is executed. The control input of the multiplexer is fed by a central test controller.

3. *Output response compactor:* Output response compacter performs lossy compression of the outputs of the CUT. As in the case of off-line testing, in BIST the output of the CUT is to be compared with the expected response (called golden signature); if CUT output does not match the expected response, fault is detected. Similar to the situation for test pattern generator, expected output responses cannot be stored explicitly in a memory and compared with the responses of the CUT. So CUT response needs to be compacted such that comparisons with expected responses (golden signatures) become simpler in terms of area of the memory that stores the golden signatures.

4. *ROM:* Stores golden signature that needs to be compared with the compacted CUT response.

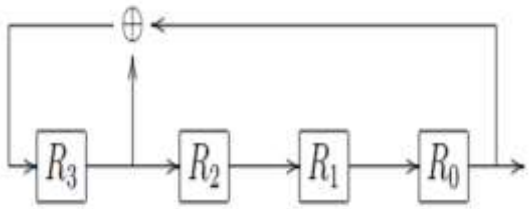
5. *Comparator:* Hardware to compare compacted CUT response and golden signature (from ROM).

6. *Test Controller:* Circuit to control the BIST. Whenever an IC is powered up (signal start BIST is made active) the test controller starts the BIST procedure. Once the test is over, the status line is made high if fault is found. Following that, the controller connects normal inputs to the CUT via the multiplexer, thus making it ready for operation. Among the modules discussed above, the most important ones are the hardware test pattern generator and the response compactor. The other ones are standard digital blocks [1]. In the next two sections we will discuss these two blocks in details.

LINEAR FEED BACK SHIFT REGISTER:

Linear feedback shift registers (LFSR's) are an efficient way of describing and generating certain sequences in hardware implementations. A linear feedback shift register is composed of a shift register R which contains a sequence of bits and a feedback function f which is the bit sum (xor) of a subset of the entries of the shift register. The shift register contains n memory cells, or stages, labelled Rn−1,. . . , R1, R0, each holding one bit. Each time a bit is needed the entry in stage R0 is output while the entry in cell Ri is passed to cell Ri−1 and the top stage Rn−1 is updated with the value f(R).

The following is a schematic of a linear feedback shift register:



In the above LFSR, let the initial entries of stages Ri be si , for $0 \leq i \leq n$. For each of the following initial entries below:

	s_3	s_2	s_1	s_0
$a)$	0	1	1	0
$b)$	1	1	1	0
$c)$	1	0	1	0
$d)$	1	1	0	0

DRAWBACKS IN LFSR:

1) To test the large circuit, the circuits are partitioned to save the test time, but the parallel testing results in excessive energy and power dissipation.
2) Due to lack of at speed equipment availability, the delay is introduced in the circuit during testing. This will cause the power dissipation.
3) Due to lack of correlation between the successive patterns, there exists a large switching activity in test mode. Since the power dissipation in CMOS circuits is directly proportional to the switching activity, so, the excessive switching activity may be responsible for cost, reliability, performance verification and technology related problems. For testing complex circuits, hierarchical approach is used. The advantage of hierarchial approach is that, every block is tested separately. That is test input is given to each block and output of each block is observed and verified. DFT(design for testability) is a technique used in chip design process to enhance the ability to generate the test vectors, to achieve measured quality level and to reduce the cost of testing. Generally, we use the conventional DFT approaches such as scan and BIST

PROPOSED TECHNIQUE:

MODIFIED LOW TRANSITION TPG WITH CLOCK SPLITTING LOGIC:

The primary advantage of our proposed technique is that it can exist for both combinational and sequential circuits, and the randomness quality of the models does not degenerate. There are many suggested methods of random pattern generators that only reduce the transitions either within the shapes or between the patterns by an n-bit LFSR.

MODIFIED LOW TRANSITION TPG:

Because of simplicity of the circuit and less area occupation, linear feedback shift register [LFSR] is used at the maximum for generating test patterns. This project proposes a novel architecture which generates the test patterns with reduced switching activities. LP-TPG structure consists of modified low power linear feedback shift register (LP- LFSR).

A new TPG Modified Low Transition LFSR (MLT LFSR) has been dealt, in which two consecutive test vectors will be compared. When there is a transition between corresponding bits of pattern, a new intermediate pattern is inserted by generating a random bit and positioning it in the corresponding bit of change from either first or second half of LFSR. The advantage of our proposed technique is without affecting the randomness, transitions between patterns get reduced and it can be used for both combinational and sequential circuits. The circuit diagram of proposed BIST is shown in below Figure which is the modified by combining LFSR and Bipartite LFSR. When compared to LT LFSR (Low Transition LFSR) [9] shown in Fig. the number of gates required, control signal needed and extra LFSR used to generate random bit are reduced in the proposed method. Random Injection (RI) circuit in existing LT LFSR which is composed of 3 AND gates and an OR gate is replaced by single EXOR gate in proposed MLT LFSR. EXOR gate is used to check the change of bits in consecutive patterns, if there is no change, output of EXOR gate which is connected to a selection line of multiplexer will be ‘0’ so the same bit will be passed to the output. When there is a change of bits, output of EXOR gate will be ‘1’ so random bit will be injected from first or second half of LFSR which is an advantage of proposed method, but in the existing LTLFSR separate LFSR is used for Random Injection. For the implementation of BIST circuit, a test pattern generator with random output value is required. The present work gives a design of a test pattern generator with reduced logic hardware. Bit TPG logic is implemented using a gate level architecture.

Proposed MLT LFSR has an FSM unit, LFSR and a data selector unit. FSM will generate control signal EN1 and EN2. LFSR polynomial used is 1+x+xN. Operation of MLT LFSR for 32 bit circuit is given below

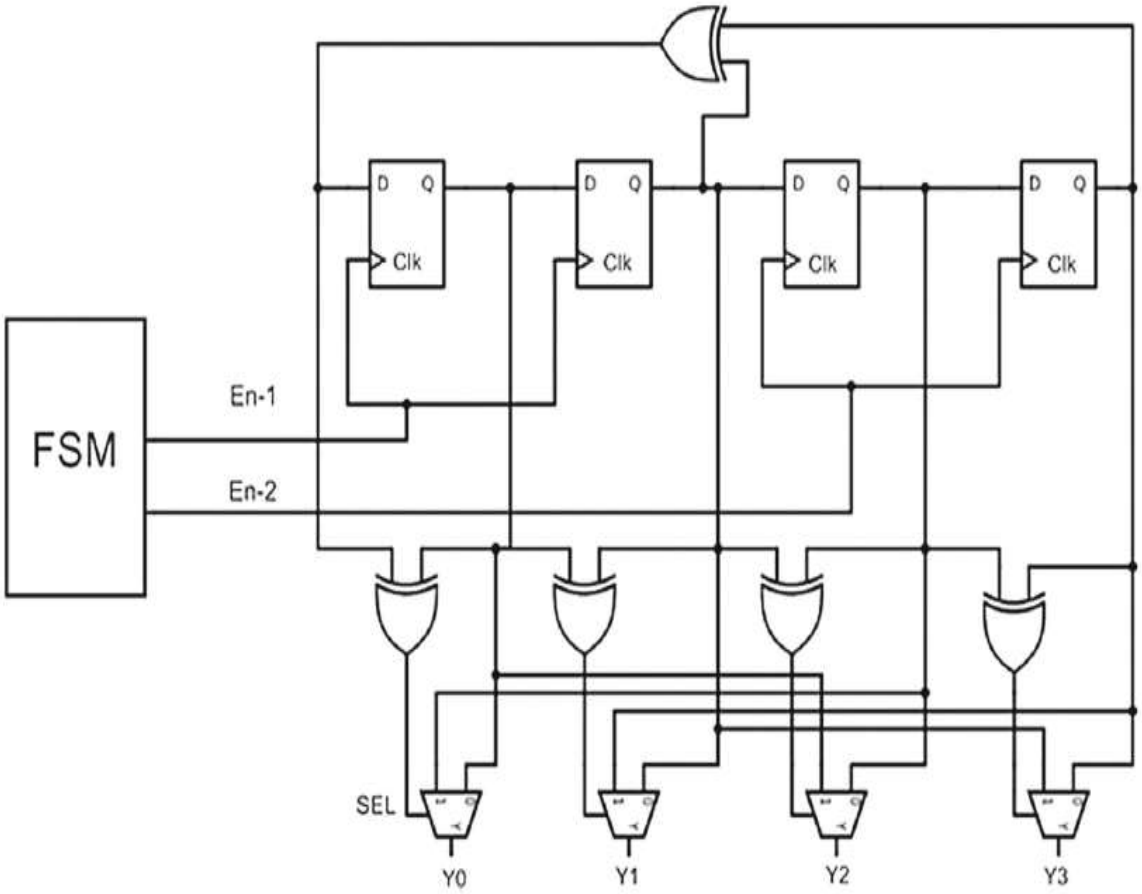


Fig: LT LOW POWER LFSR

Random pattern injection in modified low transition test pattern generator

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31
Ti	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	0	0	0	0	1	0	1	1	1	1	0	0	1	0
	R	1	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	R	R	0	0	0	R	R	1	1	1	1	0	0	1	0
Ti+1	1	1	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	1	1	0	0	1	0
	1	1	R	R	R	R	R	1	R	R	R	0	0	0	R	R	R	0	1	0	0	0	0	1	1	1	1	R	0	R	R	R
Ti+2	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1	0	0	1	0	1

R indicates Random bit
T_i indicates first pattern
T_{i+1} indicates next consecutive pattern

Table Random pattern injection in modified low transition test pattern generator

Step 1 When EN1 = 1, first half of LFSR (D0 to D15) will be working and second half (D16 to D31) will be in idle mode
Step 2 When EN2 = 1, first half of LFSR (D0 to D15) will be idle and second half (D16 to D31) will be in the working condition.
Step 3 XOR gate is used to check the transitions in the pattern, if there is no transition, output of XOR gate will be 0 and the same bit will be passed to output Y

Step 4 If transitions exist, output of XOR gate will be 1 therefore random bit will be inserted from LFSR. Table 1 and Table 2 shows output of proposedMLT LFSR and existing LT LFSR for 32 bit and transitions of successive patterns Ti to Ti+1.

Below Table shows random bit injection in the test patterns when the conditionTij _=Ti+1j satisfied [9] as given in Eq. (1)

$$T_j^{i1} = T_j^i, T_j^i = T_j^{i+1} = R, T_j^i \neq T_j^{i+1} \quad (1)$$

where Ti1j is an intermediate pattern between Tij and Ti+1j . The shaded portions indicate the change in bits from Ti to Ti+1 in column C0, C17, C18, C22 and C23. The columns which has the transitions are filled with random bits (R) generated by first and second half of LFSR. The MLT LFSR exploits the available LFSR alone for the generation of random bits R which makes its area and power efficient. Similarly for random bits are inserted Ti+1 to Ti+2.

CLOCK SPLITTING WITH MTLLFSR:

Portable devices like mobiles, iPods, and laptops consume more energy which can exhaust battery charge within a short duration. Most of the power dissipation is of the dynamic type which necessities the reduction in switching power dissipation. In portable devices there is an opportunity for switching of a part of circuit that is not in use for a certain time. This results in the reduction in dynamic power dissipation of the device there by reducing overall power. Clock splitting is the technique in which part of the design can be gated, that is registers that don't change their state are not given clock signals. By this technique the power consumption in storing the same bit to memory of the flip-flop is reduced. There are different types in which the clock splitting can be applied to a design. System level, combinational clock splitting and sequential clock splitting. In system level clock splitting a module in a design can be gated when not in use. When a mobile is left idle the mobile switches off the display and some other features that are not used always results in considerable reduction in power. Sequential clock splitting switches off the clock, applied to the flip-flops in a pipelined design are not in use during that stage. The sequential clock splitting is difficult to achieve and tools are not provided with the capability to implement this feature hence we have to predict and verify the result which is very difficult to achieve. RTL clock splitting is a technique in which the architecture is analyzed for certain condition if the registers satisfy the condition then the registers can be clock gated. The gating can be done in the architecture during code insertion or clock splitting components are inserted during the synthesis of the design. The conditions for clock splitting a design is: It should have a feedback system for registers, the enable signal activation logic of the mux should be determined, logic conditions that provide the output should be known. The gating elements are inserted according to the conditions where insertion of clock splitting elements is possible and results in considerable power reduction. The clock tree structure obtained along with the clock synthesis report gives vital information about the skew and slack in the clock distribution network. There is a problem of high fan outs and clock gating of flip-flop by individuals gating instances which results in slack and skew problem. By using the splitting and merging techniques appropriately the power dissipation can be reduced. Split is the process by which the clock gating instances are divided among a fixed number of registers. Merge is the process by which flipflops in the gates at the same time are merged under a common clock gating instance.

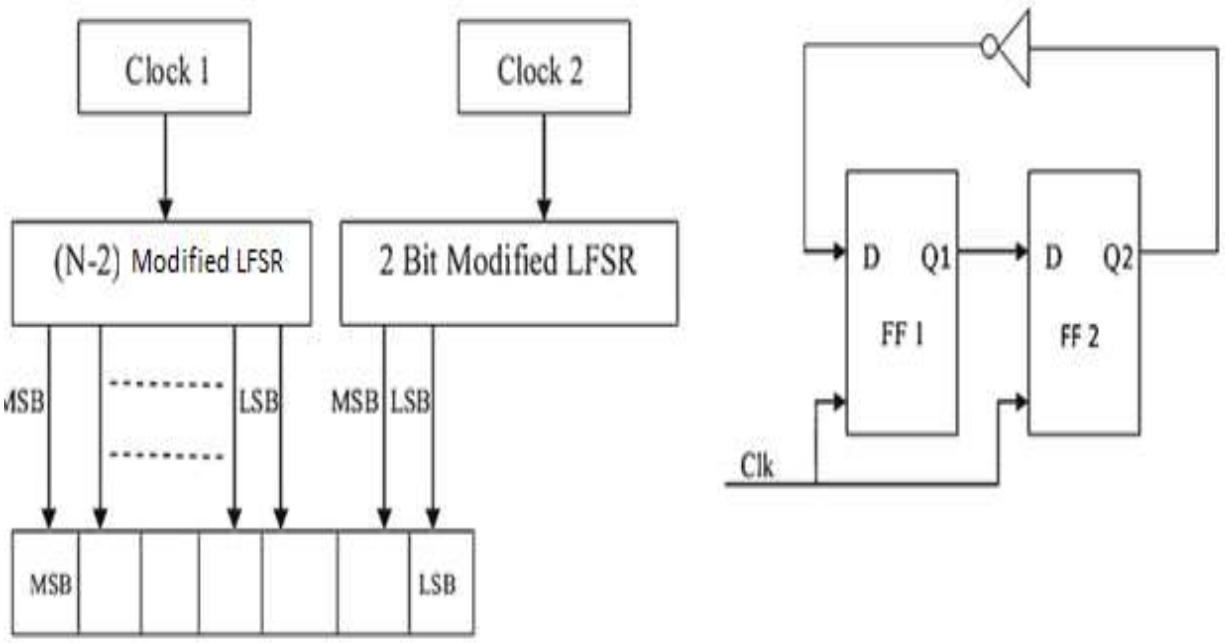


Fig: Proposed clock splitting

we are separating the LFSR into (N-2) ,2 bit. The splits logic of the 2 bit LFSR here we are applying . Two different clock pulses are applied to the 2 bit LFSR and n-2 bit lfsr. When the initial stage of first Flipflop equal to 1 then the next flipflop also getting 1 as a output.

The final output for the first cycle clock was Q1=0 then the second cycle output was Q2=1. Then these Q1 and Q2 are taken as the MSB of the final LFSR generated address and the second operation we needs to divide the clock for second stage LFSR and the resultant of the overall LSB and MSB of the LFSR generated outputs are considered as the final Address of testing of the memory processor.

CLOCK 1 and CLOCK2 diagrams are yielded as shown in below figure.

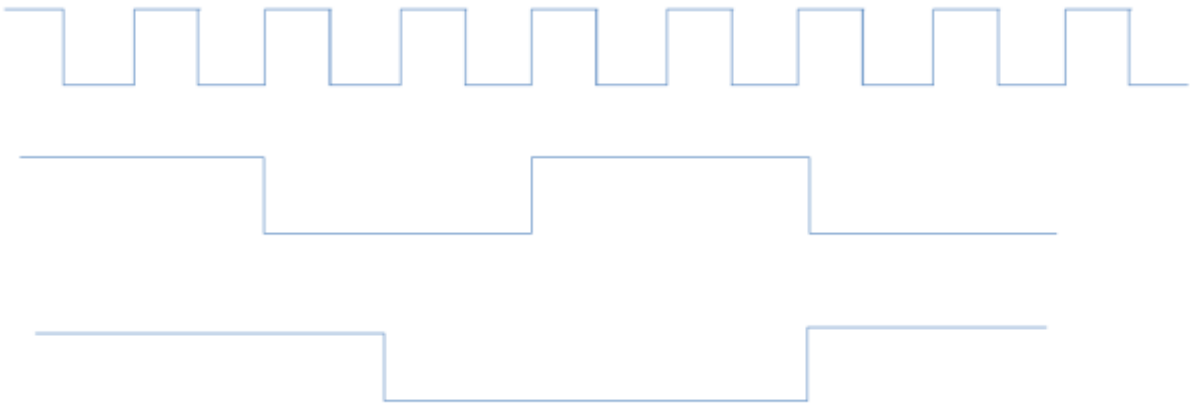


Fig: Clock splitting using single click

SEQUENTIAL BENCHMARK CIRCUIT s27:

S27 bench mark circuit is the standard sequential circuit. Here we are used s27 bench mark circuit for as a testing circuit. Applying test vectors as input to the s27 bench mark sequential circuit.

Logic Gates are taken at Primary input combination in (2^4=16).

S27 circuit have three scan circuits and then its scan inputs are 2^3=8.

Scan-in-state input are s0, s1, s2 and it's scanned out denoted as s

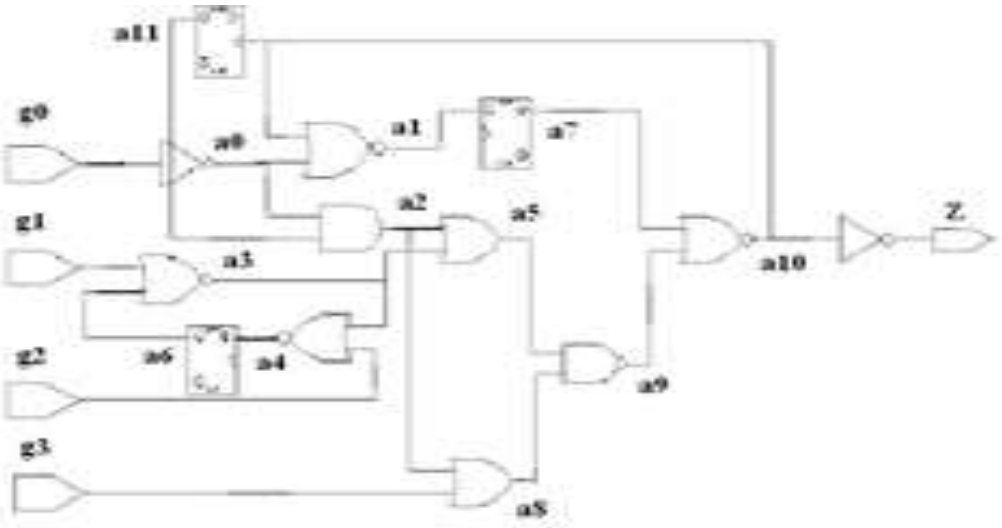


Fig: S27 CUT.

RESULT:

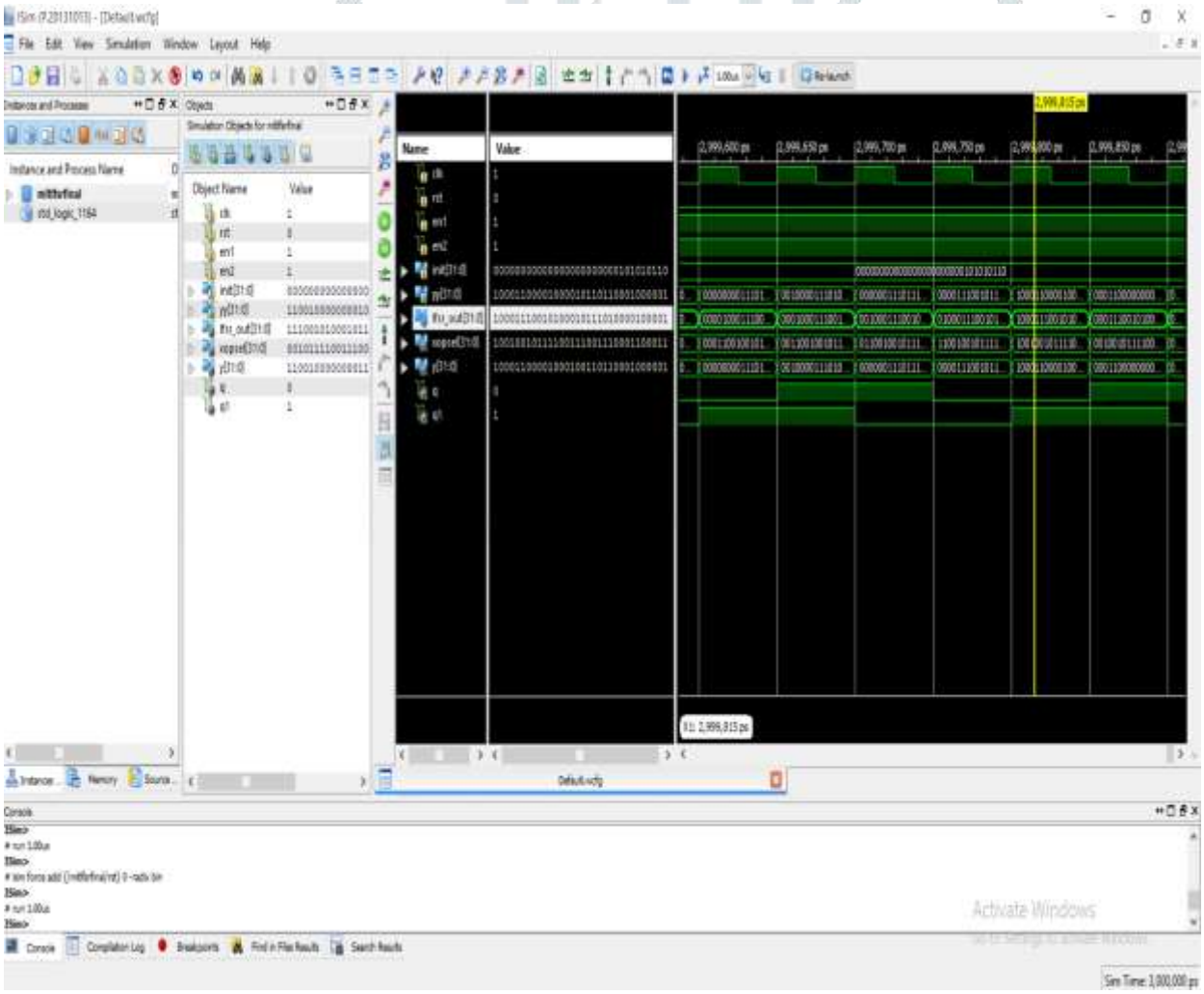


Fig: Simulation results

	Power (mW)	Time (ns)
EXISTING	560	12.328
PROPOSED	452	5.161

Fig: Synthesis result

CONCLUSION:

From the simulation analysis it can be seen that the LFSR with well chosen maximum feedback polynomial will generate random PN sequence with less amount of power. Allow-transition and highly correlated TPG to decrease power consumption of combinational and sequential benchmark has been presented. The power consumption is reduced due to low transition and increased correlation between the test vectors. Proposed MLT LFSR with clock splitting method shows good randomness with equal number of 0 s and 1 s when compared to other existing method which is an added advantage.

REFERENCES:

[1] F. Najm, "Transition Density, A New Measure of Activity in Digital Circuits", in IEEE Transaction on ComputerAided Design no.2, vol.12, pp.310-323, February 1993.

[2] A. Chandrasekaran, T. Sheng and R.Brodersen, "Low-Power CMOS Digital Design", in IEEE Journal of SolidState Circuits,

[3]. Mohammad Tehranipoor, Mehrdad Nourani, Nisar Ahmed, "Low Transition LFSR for BIST Based Applications

[4]. Fulvio Corno, Paolo Pinetum, Matteo Sonza Ford "Testability Analysis and ATPG on Behavioral RT-Level VHDL",

[5]. F.Corno, M.Rebaudengo, M.Sonza Reorda, "A Test Pattern Generation Methodology for Low Power Consumption", pp.1-5, 2008.

[6]. Dr. K. Gunavathi, Mr. K. Paramasiva, Ms.P. Subashini Lavanya, M.Umamageswaram, "A Novel BIST TPG for Testing of VLSI Circuits",

[7]. Balwinder Singh, Arun Khosla, Sukhleen Bindra, " Power optimization of LFSR for Low Power BIST", IEEE International Conference 2009, pp. 311-31.

[8] S. W. Golomb, Shift Register Sequences. Laguna Hills, CA: Aegean Park, 1982.

[9] T.-C. Huang and K.-J. Lee, "A token scan architecture for low power testing," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2001, pp. 660–669.

[10] E. J. McCluskey, "Verification testing—A pseudoexhaustive test techniques," IEEE Trans. Comput., vol. C-33, no. 6, pp. 541–546, Jun. 1984.

[11] R. Sankaralingam, B. Pouya, and N. A. Touba, "Reducing power dissipation during test using scan chain disable," in Proc. VLSI Testing Symp., Marina Del Rey, CA, 2001, pp. 319–324.

[12] J. Saxena, K. M. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2001, pp. 670–677.

[13] S. Wang, "Minimizing heat dissipation during test application," Ph.D. dissertation, EE-Systems, Univ. Southern California, Los Angeles, 1998.

[14] ——, "Generation of low power dissipation and high fault coverage patterns for scan-based BIST," in Proc. IEEE Int. Test Conf., Baltimore, MD, 2002, pp. 834–843.

[15] S. Wang and S. K. Gupta, "ATPG for heat dissipation minimization for scan testing," in Proc. IEEE-ACM Design Automation Conf., Anaheim, CA, 1997, pp. 614–619.

[16] ——, "DS-LFSR: A new BIST TPG for low heat dissipation," in Proc. IEEE Int. Test Conf., Washington, DC, 1997, pp. 848–857.

[17] ——, "ATPG for heat dissipation minimization during test application," IEEE Trans. Comput., vol. 47, no. 2, pp. 256–262, Feb. 1998.
[18] ——, "LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation," in Proc. IEEE Int. Test Conf., Atlantic City, NJ, 1999, pp. 85–94.
[19] ——, "An automatic test pattern generator for minimizing switching activity during scan testing activity," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 21, no. 8, pp. 954–968, Aug. 2002.
[20] ——, "LT-RTPG: A new test-per-scan BIST TPG for low switching activity," Univ. Southern California, Los Angeles, Tech. Rep. CENG 02-06, 2002.
[21] N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective, 2nd ed. Reading, MA: Addison-Wesley, 1992.
[22] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in Proc. VLSI Testing Symp., Atlantic City, NJ, 1993, pp. 4–9.