

Implementation of Efficient LFSR with Real Time BIST Application

Beril Susan Philip¹, Vuduthala Srinivas²
Assistant Professor¹, Associate Professor²

Department of Electronics and Communication Engineering
Malla Reddy Engineering College, Secunderabad

Abstract- In VLSI technology area, delay, power consumption and chip cost were major considerations. In this paper to reduce the power consumption three intermediate patterns were introduced for pattern generation. The main purpose of this injected logic to reduced usage of the power. The primary inputs which obviously reduced the activity of switching pattern. So by this way the total switching during the test mode reduces. Obviously, the consumption of power was automatically diminished. The power of a circuit during test by generating 3 intermediate patterns among the random patterns by dropping the hardware utilization. The objective of having intermediate patterns were to reduce the transitional activities of Primary Inputs (PI) which ultimately reduced the switching activities inside the Circuit under Test (CUT) and hence power consumption was also reduced. The experimental results for c17 benchmark, shows with and without fault confirm and the fault coverage of the proposed circuit being tested with less power.

Index Terms- VLSI technology, Circuit under Test, switching pattern.

I. INTRODUCTION

The main challenging areas in VLSI are performance, cost, and power dissipation. Due to switching i.e., the power consumed testing, due to short circuit current flow and charging of load area, reliability and power. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation VLSI circuits. The power dissipation during test mode is 200% P more than in normal mode. Hence it is important aspect to optimize power during testing. Power optimization is one of the main challenges. Test Pattern generation has long been carried out by using conventional Linear Feedback Shift Registers (LFSR's). LFSR's are a series of flip-flop's connected in series with feedback taps defined by the generator polynomial. The seed value is loaded into the outputs of the flip-flops. The only input required to generate a random sequence is an external clock where each clock pulse can produce a unique pattern at the output of the flip-flops. This random sequence at the output of the flip-flops can be used as a test pattern. The number of inputs required by the circuit under test must match with the number of flip-flop outputs of the LFSR. This test pattern is run on the circuit under test for desired fault coverage. The power consumed by the chip under test is a measure of the switching activity of the logic inside the chip which depends largely on the randomness of the applied input stimulus. Reduced correlation between the successive vectors of the applied stimulus into the circuit under test can result in much higher power consumption by the device than the budgeted power. A new low power pattern

generation technique is implemented using a modified conventional Linear Feedback Shift Register.

The first semiconductor chips held one transistor each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors. At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like Ultra-large-scale Integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use. Even VLSI is now somewhat quaint, given the common assumption that all microprocessors are VLSI or better. As of early 2008, billion-transistor processors are commercially available, an example of which is Intel's Montecito Itanium chip. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). Another notable



example is NVIDIA's 280 series GPU. This microprocessor is unique in the fact that its 1.4 Billion transistor count, capable of a teraflop of performance, is almost entirely dedicated to logic (Itanium's transistor count is largely due to the 24MB L3 cache). Current designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM cell, however, are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability). Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications.

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. For example, a chip designed solely to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products (ASSPs). As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks. Such an ASIC is often termed a SoC (system-on-a-chip). Designers of digital ASICs use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs. Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

II. EXISTING WORK OR LITERATURE SURVEY

The main challenging areas in VLSI are performance, cost, power dissipation is due to switching i.e. the power consumed testing, due to short circuit current flow and charging of load area, reliability and power. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation VLSI circuits. The power dissipation during test mode is 200% P more than in normal mode. Hence it is important aspect to optimize power during testing. Power optimization is one of the main challenges. BIST architecture: A typical BIST architecture consists of TPG - Test Pattern Generator TRA - Test Response Analyzer Control Unit. Tapping can be taken as we wish but as per tapping change

the LFSR output generate will change & as we change in no of flip-flop the probability of repetition of random number will reduce. The initial value loading to the LFSR is known as seed value.

Need for using BIST technique: Today's highly integrated multi-layer boards with fine-pitch ICs are virtually impossible to be accessed physically for testing. Traditional board test methods which include functional test, only accesses the board's primary I/Os, providing limited coverage and poor diagnostics for board-network fault. In circuit testing, another traditional test method works by physically accessing each wire on the board via costly "bed of nails" probes and testers. To identify reliable testing methods which will reduce the cost of test equipment, a research to verify each VLSI testing problems has been conducted.

The Gate to I/O Pin Ratio Problem: As ICs grow in gate counts; it is no longer true that most gate nodes are directly accessible by one of the pins on the package. This makes testing of internal nodes more difficult as they could neither no longer be easily controlled by signal from an input pin (controllability) nor easily observed at an output pin (observe ability). Pin counts go at a much slower rate than gate counts, which worsens the controllability and observe ability of internal gate nodes. Integrated Circuits (IC's) are designed and manufactured to meet application specific functional requirements. Some examples of applications are camera-on-a-chip, MP3 player, etc. These functional requirements often need to be balanced with the desired IC performance, maximum allowable power consumption and overall packaged and tested IC cost. Generally total power consumption of the device is a sum of the power consumed by the core and I/O's. Appropriate package attributes such as the material and thermal properties need to be selected to ensure maximum heat dissipation of the die through the package. Implementation of Efficient LFSR with Real Time BIST Application: This increase in the power consumption of the IC in the test mode is well known in the industry to cause sudden un-repairable device failures resulting in significant manufacturing fall-out directly impacting the cost of the IC. Today a combination of external Automated Test Equipment (ATE) and internal BIST (Built-In-Self-Test) techniques are used to ensure the highest possible fault coverage of the device at the lowest possible cost IC testing using exclusively external ATE's can require SOC architects to allocate a fairly large number of pins of the device to invoke the test procedure and run vectors into and through the various blocks of the device such as memory, user defined logic, dedicated functional macros, etc. Combination of external ATE's and internal BIST however can result in, utilizing far fewer external pins on the IC but at the cost of embedding test logic inside device. SOC's typically integrate multiple Microprocessors, various types of memories such as SRAM, ROM, Flash, user defined logic, etc. Most SOC's are heavily populated with multiple instances of memory. Also included can be IP macros such as Digital Signal Processors, Analog to digital converters, etc. SOC's typically contain multiple types of I/O's ranging from standard CMOS, LVTTTL to high speed I/O's such as LVDS (Low voltage differential signal). The pin count can



range from a few hundred pins to over a thousand with custom designed packages including multiple layers of substrate. SOC's are solution driven with the intent of providing a single chip solution for particular applications such as digital cameras, MP3 players, storage drives, printers, networking, etc. SOC's for the Consumer electronics typically include mixed signal components such as A/D, D/A, multiple instances of SRAM, Flash, ROM and user defined logic. Networking and Storage applications tend to be extremely compute intensive and therefore it is not uncommon to find these devices containing multiple microprocessor cores along with many megabits of memory and high speed I/O's operating in the giga-bit per second range.

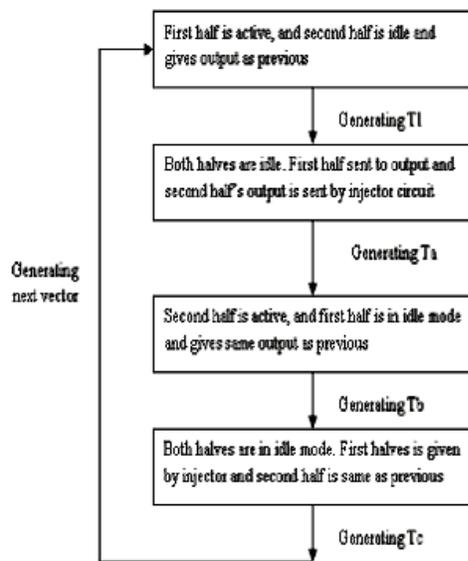
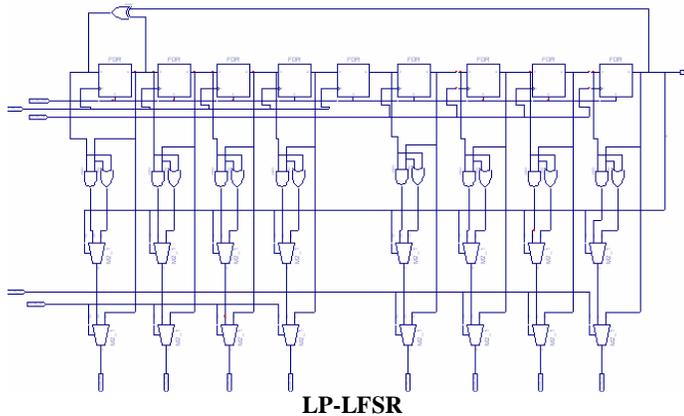
SOC Design Tools and Methodology: Generally, the architecture of the entire SOC is designed and simulated by chip architects. Front end designers are involved in converting the architecture level IC requirements to detailed circuit level descriptions using design description languages such as Verilog, VHDL, etc. Depending on the complexity of the overall IC project, it is not uncommon to find front end design teams ranging from tens of engineers to a few hundred spending anywhere from 6 months to a year.

III. WRITE DOWN YOUR STUDIES AND FINDINGS

One way to improve the correlation between the bits of the successive vectors is to avoid frequent transitioning of the logic levels of the primary inputs. The new approach entails inserting 3 intermediate vectors between every two successive vectors. The total number of signal transitions between these 5 vectors is equal to the total number of signal transitions between the 2 successive vectors generated using the conventional approach. This reduction of signal transition activity in the primary inputs reduces the switching activity inside the design under test and therefore results in reduced power Consumption by the device under test. The additional circuitry used to accomplish the generation of the 3 intermediate vectors is minimal at best consisting of few logic gates. The number of LFSR outputs required is driven by the number of test inputs required for circuit under test. The technique of inserting 3 intermediate vectors is achieved by modifying the conventional LFSR circuit with two additional levels of logic between the conventional flip-flop outputs and the low power outputs.

The first level of hierarchy from the top down includes logic circuit design for propagating either the present or the next state of the flip-flops to the second level of hierarchy. The second level of hierarchy is a multiplexer function that provides for selecting between the two states (present or next) to be propagated to the outputs as low power output. Minimal at best consisting of few logic gates. In the simulation environment, the outputs of the flip-flops are loaded with the seed vector. The feedback taps are selected pertinent to the characteristic polynomial $x^8 + x + 1$. Only 2 inputs pins, namely test enable and clock are required to activate the generation of the pattern as

well as simulation of the design circuit. It is also noteworthy here that the intermediate vectors in addition to aiding in reducing the number of transitions can also empirically assist in detecting faults just as good as the conventional LFSR patterns. Description of the technique to produce low power pattern for BIST The following is a description of a low power test pattern generation technique as depicted in the 9-bit LFSR based schematic. Verilog based test bench as shown in Appendix B is used in assigning the initial output states (0100 1011) of the 9-bit LFSR. The feedback taps are designed for maximal length LFSR generating all zeros and all one's as well. The first step is to generate. The first step is to generate T1, the first vector by enabling (clocking) the first 4-bits of the LFSR and disabling (not clocking) the last 4 bits. This Shifts the first 4 bits to the right by one bit. The feedback bits of the LFSR are the outputs of the 8th and the first flip-flop. The output of the 8th flip-flop is 1 and the output of the first flip-flop is 0. The exclusive-or of the 8th-flip-flop (logic 1 in this case) and the first flip-flop(logic 0 in this case) is input (1 EXOR 0 = 1 into the first D flip-flop. The new pattern in the first four bits of the LFSR is 1010. Note that the shaded register is clocked along with the first 4 bits of the LFSR. So the input of the shaded flip-flop is the output of the 4th flip-flop which in this case is 0. Also note that prior to the first clock, the input of the shaded register was the seed value of the 4th flip-flop at the output of the 4th flip-flop which in this case is 0. So after the first clock this value of 0 will now appear at the output of the shaded flip-flop. In other words the value of the 4th output is stored in this shaded register and is used in the next few steps. The first 4 shifted bits of the LFSR and the last 4 unshifted bits (i.e. the seed value) are propagated as T1 (1010 1011) to the final outputs. Next few steps involve generating the 3 intermediate patterns from T1. These patterns are defined as Ta, Tb and Tc shown in below flow. Ta is generated by maintaining (disabling the clock to the first 4 bits) the first four bits of the LFSR outputs (as is from T1) as the final first four low power outputs 1010. Note that the clock to the last four bits of the LFSR is also disabled. The last four bits however are the outputs from the injector circuits. The injector circuit compares the next value (the input of the D-flip-flop) with the current value (the output of the D-flip-flop). According to T1, the outputs (current values) of the last 4 bits of the LFSR are 1011. The next values are the values at the inputs of the D-flip-flops which in this case are 0101. Compare the current values (1011) bit by bit with the next values (0101). If the values bit by bit are not the same then use the random generator feedback R (in this case is logic 1) as the bit value as shown in the schematic above. If however both values bit by bit are the same then propagate that bit value to output as opposed to the R bit. This bit by bit comparison gives us the last four bits of Ta to be 1111. Therefore Ta = 1010 1111. Next step is to generate Tb. Shift the last 4 flip-flops to the right one bit but do not shift the first 4 flip-flops to the right. The clock to the first 4 bits plus the shaded flip flop is disabled. The clock to the last 4 bits is enabled. Propagate the outputs of the flip-flops of the entire LFSR as opposed to the outputs of the injection circuit to the outputs (low power).

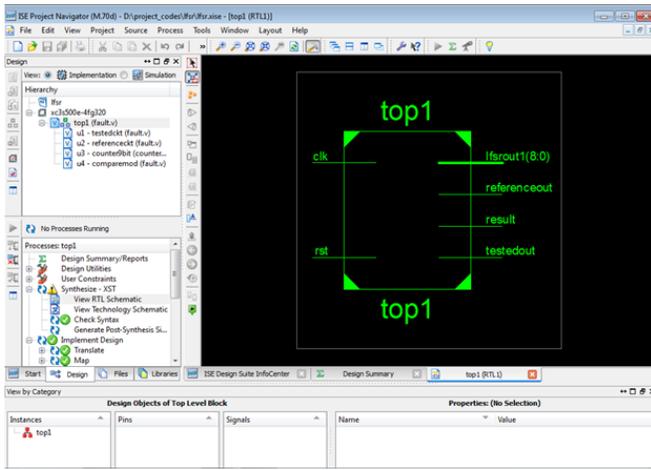


Proposed algorithms for low power LFSR

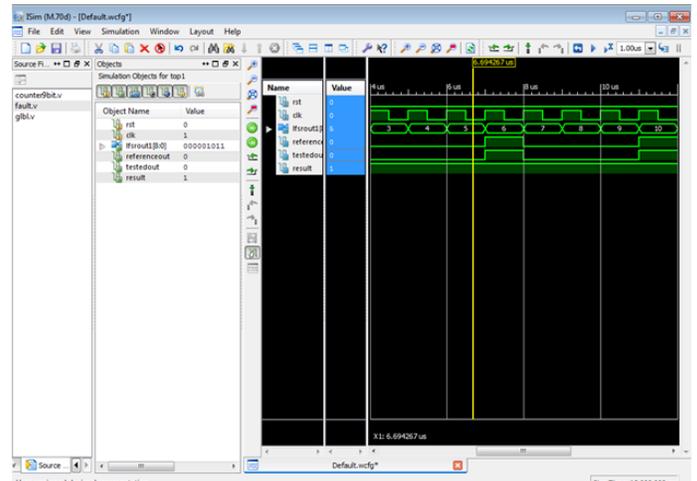
Ta is generated by maintaining (disabling the clock to the first 4 bits) the first four bits of the LFSR outputs (as is from T1) as the final first four low power outputs 1010. Note that the clock to the last four bits of the LFSR is also disabled. The last four bits however are the outputs from the injector circuits. The injector circuit compares the next value (the input of the D-flip-flop) with the current value (the output of the D-flip-flop). According to T1, the outputs (current values) of the last 4 bits of the LFSR are 1011. The next values are the values at the inputs of the D-flip-flops which in this case are 0101. Compare the current values (1011) bit by bit with the next values (0101). If the values bit by bit are not the same then use the random generator feedback R (in this case is logic 1) as the bit value as shown in the schematic above. If however both values bit by bit are the same then propagate that bit value to output as opposed to the R bit. This bit

by bit comparison gives us the last four bits of Ta to be 1111. Therefore Ta = 1010 1111. Next step is to generate Tb. Shift the last 4 flip-flops to the right one bit but do not shift the first 4 flip-flops to the right. The clock to the first 4 bits plus the shaded flip flop is disabled. The clock to the last 4 bits is enabled. Propagate the outputs of the flip-flops of the entire LFSR as opposed to the outputs of the injection circuit to the outputs (low power). The injection circuits are disabled. As in Ta, maintain the first four LFSR outputs (1010) as the low power outputs. Again from Ta, the inputs of the last four D flip-flops from the previous step (generating Ta) are 0101. Also note that the output of the shaded register is 0 from the previous step (generating Ta). Therefore the input of the 5th flip-flop is a 0. The outputs of the last 4 flip-flops are 0101 resulting in Tb = 1010 0101. The 3rd intermediate vector Tc is generated via disabling the clock to the entire LFSR. Generating injection circuit outputs for Tc is conceptually the same as explained above in generating Ta. Current values (the outputs of the flip-flops) of the first four flip-flops are compared with the next values (the inputs of the flip-flops) of the flip-flops. The feedback from the 8th flip-flop is 1 (please see generating Tb). Therefore the logical feed forward value of R is 1. The feedback value from the first flip-flop is also 1 as per the current values above. The exclusive or of two ones is a 0. Therefore the input to the first flip-flop is a 0 which is also the next state of the first flip-flop. Hence the next values are 0 for the first flip-flop and 101 for the 2nd, 3rd and 4th flip-flop respectively. The next values are 0101. The first four outputs from the injection circuit are 1111. The last 4 outputs are the same as Tb which are 0101 resulting in the 3rd and final intermediate vector Tc = 1111 0101. Generating T2 is quite similar to generating T1. As in Tc the outputs of the last four LFSR flops are 0101. The outputs of the first 4 flip-flops of the LFSR are the current values which are 1010. Therefore the seed vector for generating T2 is 1010 0101. Shift the first four bits of the LFSR plus the shaded flip-flop. Do not clock the last four flip-flops. Propagate the outputs of the entire LFSR to the final low power outputs. The output of the 8th flip-flop from the previous step (generating Tc) is a 1 and the output of the first flip-flop from the previous step (generating Tc) is also a 1. The exclusive or of the output of the 8th flip-flop and the first flip-flop is 0. Therefore the input to the first flip-flop will be a 0. The inputs to the 2nd, 3rd, 4th and the shaded flip-flops are 1010. These are also the current values from the previous step (generating Tc). Shifting the first four flip-flops of the LFSR to the right by one bit results in 0101 as the outputs of the first four flip-flops. Therefore T2 generated is 0101 0101.

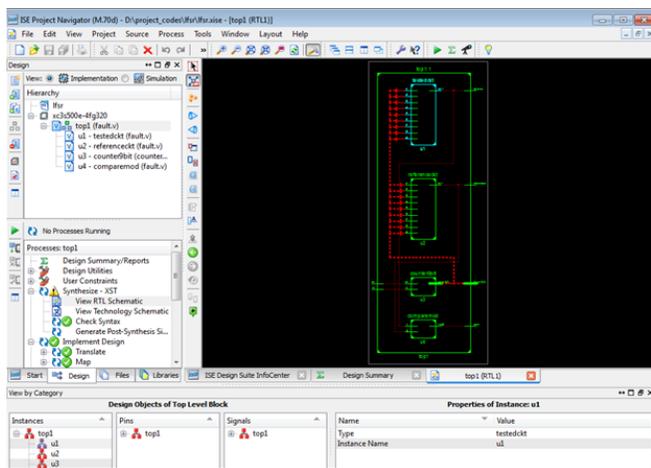
IV. RESULTS AND DISCUSSION



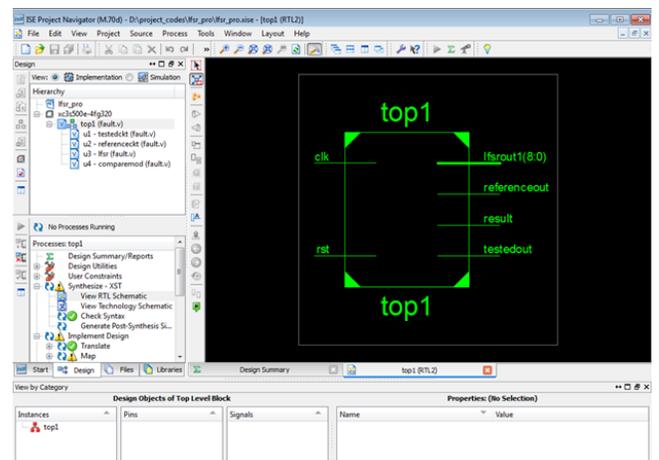
RTL Schematic view



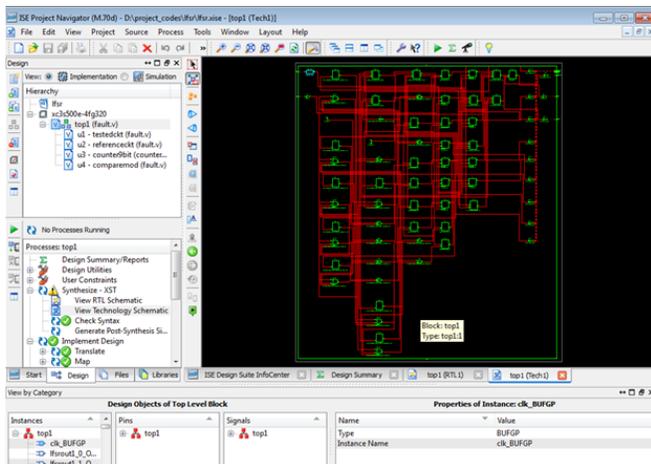
Simulation Wave Forms



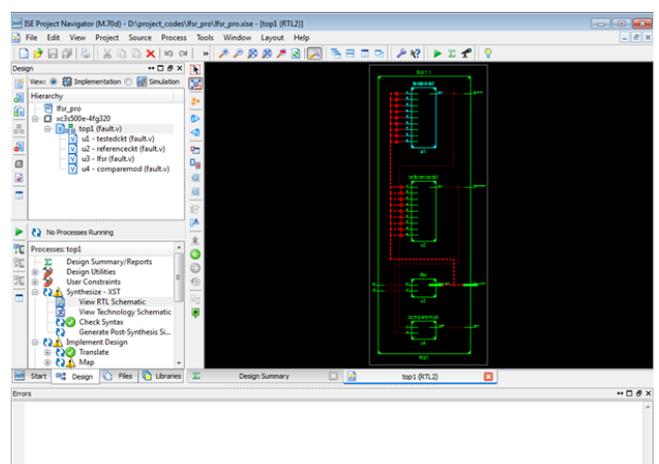
RTL Schematic view



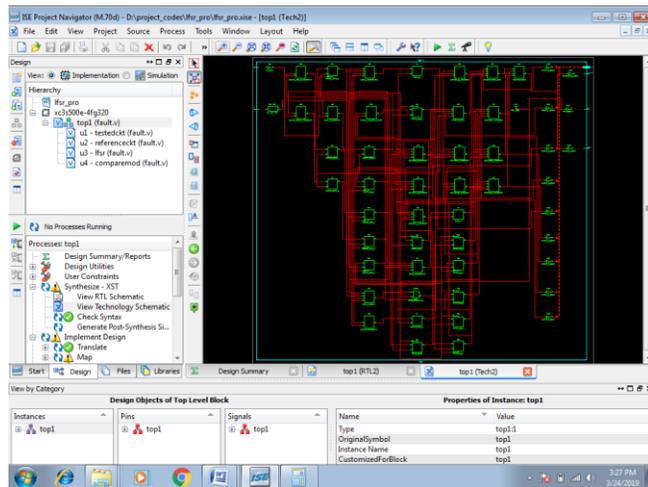
RTL Schematic view



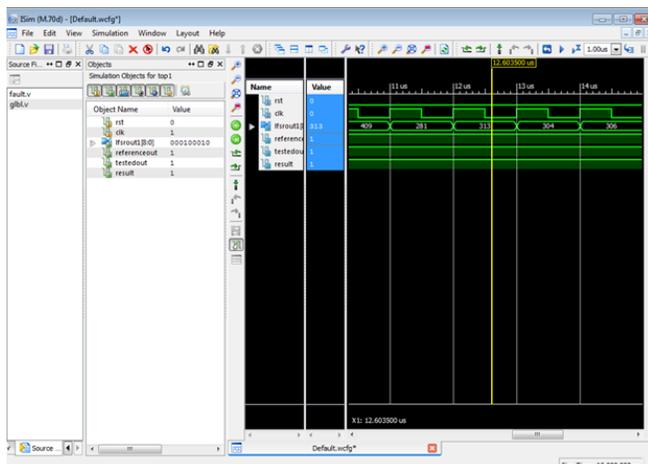
View technology Schematic



RTL Schematic view



View Technology Schematic



Simulation Wave Forms

V. CONCLUSION

In this project the proposed method was gave the test patters were generated with more correlation which missed in the conventional one. Based on the simulation results it would understand how the circuit was determining the fault coverage. The proposed system utilized only 21 LUT's (look up tables) consumed less area when compared with the conventional system, it utilized 27LUTs. In addition, the power consumption was also less in the proposed design which was 0.1719mw. The existed one utilized 0.2203mw power. In this paper VERILOG was used for implementing the RTL. XILINX ISE 12.3i is used for performing synthesis.

REFERENCES

- [1] Michael L. Bushnell, Vishwani D. Agawal, "Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits," Kluwer Academic Publishers, 2000.
- [2] I. Pomeranz and S. M. Reddy, "Static test compaction for scan-based designs to reduce test application time," in Proc. Asian Test Symp., Singapore, 1998, pp. 198–203.
- [3] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC scan: The war against unmodelled defects," in Proc. Int. Test Conf., Washington, DC, USA, 1996, pp. 250–258.
- [4] J. Rearick, "Too much delay fault coverage is a bad thing," in Proc. Int. Test Conf., Baltimore, MD, USA, 2001, pp. 624–633.
- [5] X. Lin and R. Thompson, "Test generation for designs with multiple clocks," in Proc. Design Autom. Conf., Anaheim, CA, USA, 2003, pp. 662–667.
- [6] G. Bhargava, D. Meehl, and J. Sage, "Achieving serendipitous N-detect mark-offs in multi-capture-clock scan patterns," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2007, pp. 1–7.
- [7] I. Park and E. J. McCluskey, "Launch-on-shift-capture transition tests," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2008, pp. 1–9.
- [8] E. K. Moghaddam, J. Rajski, S. M. Reddy, and M. Kassab, "At-speed scan test with low switching activity," in Proc. VLSI Test Symp., Santa Cruz, CA, USA, 2010, pp. 177–182.
- [9] I. Pomeranz, "Generation of multi-cycle broadside tests," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 8, pp. 1253–1257, Aug. 2011.
- [10] I. Pomeranz, "Multicycle tests with constant primary input vectors for increased fault coverage," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 31, no. 9, pp. 1428–1438, Sep. 2012.
- [11] I. Pomeranz, "Multi-cycle broadside tests with runs of constant primary input vectors," IET Comput. Digit. Tech., vol. 8, no. 2, pp. 90–96, Mar. 2014.
- [12] I. Pomeranz, "A multicycle test set based on a two-cycle test set with constant primary input vectors," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 34, no. 7, pp. 1124–1132, Jul. 2015.
- [13] B. Koenemann, "LFSR-coded test patterns for scan designs," in Proc. Eur. Test Conf., Munich, Germany, 1991, pp. 237–242.
- [14] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift register," in Proc. Int. Test Conf., Baltimore, MD, USA, 1992, pp. 120–129.
- [15] C. Barnhart et al., "OPMISR: The foundation for compressed ATPG vectors," in Proc. Int. Test Conf., Baltimore, MD, USA, 2001, pp. 748–757.
- [16] J. Rajski et al., "Embedded deterministic test for low cost manufacturing test," in Proc. Int. Test Conf., Baltimore, MD, USA, 2002, pp. 301–310.
- [17] A.-W. Hakmi, H.-J. Wunderlich, C. G. Zoellin, A. Glowatz, and F. Hapke, "Programmable deterministic built-in self-test," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2007, pp. 1–9.
- [18] D. Cyszcz et al., "Deterministic clustering of incompatible test cubes for higher power-aware EDT compression," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 8, pp. 1225–1238, Aug. 2011.
- [19] A. Chandra, J. Saikia, and R. Kapur, "Breaking the test application time barriers in compression: Adaptive scan-cyclical (AS-C)," in Proc. Asian Test Symp., New Delhi, India, 2011, pp. 432–437.
- [20] O. Acevedo and D. Kagaris, "Using the Berlekamp–Massey algorithm to obtain LFSR characteristic polynomials for TPG," in Proc. Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst., Austin, TX, USA, 2012, pp. 233–238.
- [21] X. Lin and J. Rajski, "On utilizing test cube properties to reduce test data volume further," in Proc. Asian Test Symp., Niigata, Japan, 2012, pp. 83–88.
- [22] T. Moriyasu and S. Ohtake, "A method of one-pass seed generation for LFSR-based deterministic/pseudo-random testing of static faults," in Proc. Latin-Amer. Test Symp., Puerto Vallarta, Mexico, 2015, pp. 1–6.
- [23] I. Pomeranz, "Computation of seeds for LFSR-based diagnostic test generation," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 34, no. 12, pp. 2004–2012, Dec. 2015.
- [24] I. Pomeranz, "Computing seeds for LFSR-based test generation from nontest cubes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 6, pp. 2392–2396, Jun. 2016.



International Journal For Advanced Research In Science & Technology

A peer reviewed international journal

www.ijarst.in

IJARST

ISSN: 2457-0362

- [25] P. H. Bardell, W. H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques. New York, NY, USA: Wiley, 1987.
- [26] I. Pomeranz and S. M. Reddy, "Forming N-detection test sets without test generation," ACM Trans. Design Autom., vol. 12, no. 2, 2007, Art. no. 18.