BIST Based PRPG with Test Compression Capabilities

M.Ayesha[1], P.Jayarami Reddy[2]
[1]PG Scholar, Dr. K.V. Subbareddy Institute of Technology, Kurnool, AP, India,
[2]Assistant Professor, Dr. K.V. Subbareddy Institute of Technology, Kurnool, AP, India.

Abstract: This paper describes a low-power (LP) programmable generator capable of producing pseudorandom test patterns with desired toggling levels and increased fault coverage gradient compared with the best-to-date built-in self-test (BIST)- primarily Based Pseudorandom test pattern generators. It is comprised of a linear finite state machine (a linear feedback shift register or a hoop generator) driving an appropriate phase shifter, and it comes with a variety of features permitting this device to supply binary sequences with preselected toggling (PRESTO) activity. We tend to introduce a technique to automatically choose many controls of the generator offering easy and precise tuning. The identical technique is subsequently used to deterministically guide the generator toward test sequences with improved fault-coverage-to pattern- count ratios. Furthermore, this paper proposes an LP test compression technique that permits shaping the test power envelope in a fully predictable, accurate, and flexible fashion by adapting the PRESTO-based logic BIST (LBIST) infrastructure. The proposed hybrid scheme efficiently combines test compression with LBIST, where both techniques can work synergistically to deliver top quality tests. Experimental results obtained for industrial styles illustrate the feasibility of the proposed test schemes and are reported herein.

Index Terms: Built-in self-test (BIST), low-power (LP) test, pseudorandom test pattern generators (PRPGs), and test data volume compression

I. INTRODUCTION

The test pattern generator produces test vectors that are applied to the tested circuit during pseudo-random testing of combinational circuits. The nature of the generator thus directly influences the fault coverage achieved the influence of the type of pseudo-random pattern generator on stuck-at fault coverage. Linear feedback shift registers (LFSRs) are mostly used as test pattern generators, and the generating polynomial is primitive to ensure the maximum period. We have shown that it is not necessary to use primitive polynomials, and moreover that their using is even undesirable in most cases. This fact is documented by statistical graphs. The necessity of the proper choice of a generating polynomial and an LFSR seed is shown here, by designing a mixed-mode BIST for the ISCAS benchmarks as the complexity of VLSI circuits constantly increases, there is a need of a built-in self-test (BIST) to be used. Built-in self-test enables the chip to test itself and to evaluate the circuit's response. Thus, the very complex and expensive external ATE (Automatic Test Equipment) may be completely omitted, or its complexity significantly reduced. Moreover, BIST enables an easy access to internal structures of the tested circuit, which are extremely hard to reach from outside. There have been proposed many BIST equipment design methods. In most of the state-of-the-art methods some kind of a pseudorandom pattern generator (PRPG) is used to produce vectors to test the circuit. These vectors are applied to the circuit either as they are, or the vectors are modified by some additional circuitry in order to obtain better fault coverage. Then the circuit's response to these vectors is evaluated in a response analyzer. Usually, linear feedback shift registers (LFSRs) or cellular automata (CA) are used as PRPGs, for their simplicity. Patterns generated by simple LFSRs or CA often do not provide a satisfactory fault coverage. Thus, these patterns have to be modified somehow. One of the most known approaches is the weighted random pattern testing. Here the LFSR code words are modified by a weighting logic to produce a test with given probabilities of occurrence of 0's and 1's at the particular circuit under test (CUT) inputs. Many papers dealing with the computation of the weights and the design of the weighting logic have been published.

II. BASIC ARCHITECTURE

An n-bit PRPG connected with a phase shifter feeding scan chains forms a kernel of the generator producing the actual pseudorandom test patterns. A linear feedback shift register or a ring generator can implement a PRPG. More importantly, however, n hold latches are placed between the PRPG and the phase shifter. Each hold latch is individually controlled via a corresponding stage of an n-bit toggle control register. As long as its enable input is asserted, the given latch is transparent for data going from the PRPG to the phase shifter, and it is said to be in the toggle mode. When the latch is disabled, it captures and saves, for a number of clock cycles, the corresponding bit of PRPG, thus feeding the phase shifter (and possibly some scan chains) with a constant value. It is now in the hold mode. It is worth noting that each phase shifter output is obtained by XOR-ing outputs of three different hold latches. Therefore, every scan chain remains in a low-power mode provided only disabled hold latches drive the corresponding phase shifter output the toggle control register supervises the hold latches. Its content comprises 0s and 1s, where 1s indicate latches in the toggle mode, thus transparent for data arriving from the PRPG. Their fraction determines a scan switching activity.

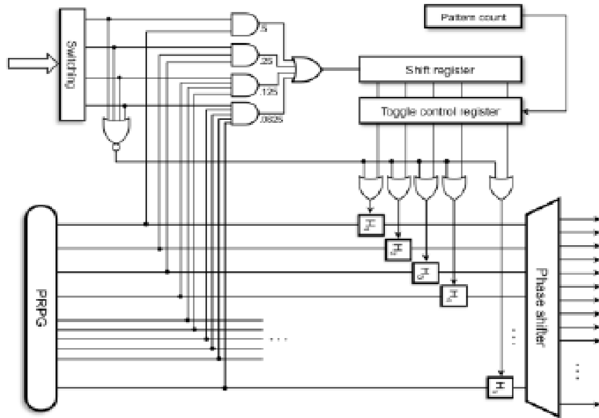

Fig.1. Basic architecture of a PRESTO generator

The control register is reloaded once per pattern with the content of an additional shift register. The enable signals injected into the shift register are produced in a probabilistic fashion by using the original PRPG with a programmable set of weights. The weights are determined by four AND gates producing 1s with the probability of 0.5, 0.25, 0.125, and 0.0625, respectively. The OR gate allows choosing probabilities beyond simple powers of 2. A 4-bit register Switching is employed to activate AND gates, and allows selecting a user-defined level of switching activity. For example, the switching code 0100 will set to 1, on the average, 25% of the control register stages, and thus 25% of hold. Latches will be enabled. Given the phase shifter structure, one can assess then the amount of scan chains receiving constant values, and thus the expected toggling ratio. An additional 4-input NOR gate detects the switching code 0000, which is used to switch the LP functionality off. It is worth noting that when working in the weighted random mode, the switching level selector ensures statistically stable content of the control register in terms of the amount of 1s it carries. As a result, roughly the same fraction of scan chains will stay in the LP mode, though a set of actual low toggling chains will keep changing from one test pattern to another. It will correspond to a certain level of toggling in the scan chains. With only 15 different switching codes, however, the available toggling granularity may render this solution too coarse to be always acceptable. Section III presents additional features that make the PRESTO generator fully operational in a wide range of desired switching rates.

While preserving the operational principles of the basic solution, this approach splits up a shifting period of every test pattern into a sequence of alternating hold and toggle intervals. To move the generator back and forth between these two states, we use a T-type flip-flop that switches whenever there is a 1 on its data input. If it is set to 0, the generator enters the hold period with all latches temporarily disabled regardless of the control register content. This is accomplished by placing AND gates on the control register outputs to allow freezing of all phase shifter inputs. This property can be crucial in SoC designs where only a single scan chain crosses a given core, and its abnormal toggling may cause locally unacceptable heat dissipation that can only be reduced due to temporary hold periods if the T flip-flop is set to 1 (the toggle period), then the latches enabled through the control register can pass test.

Data moving from the PRPG to the scan chains two additional parameters kept in 4-bit Hold and Toggle registers determine how long the entire generator remains either in the hold mode or in the toggle mode, respectively. To terminate either mode, a 1 must occur on the T flip-flop input. This weighted pseudorandom signal is produced in a manner similar to that of weighted logic used to feed the shift register. The T flip-flop controls also four 2-input multiplexers routing data from the Toggle and Hold registers. It allows selecting a source of control data that will be used in the next cycle to possibly change the operational mode of the generator .For example, when in the toggle mode, the input multiplexers observe the Toggle register. Once the weighted logic outputs 1, the flip-flop toggles, and as a result all hold latches freeze in the last recorded state. They will remain in this state until another 1 occurs on the weighted logic output. The random occurrence of this event is now related to the content of the Hold register, which determines when to terminate the hold mode.

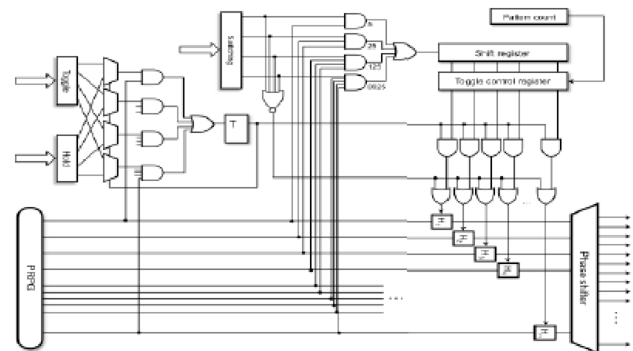

Fig.2. Fully operational version of PRESTO

A. Improving Fault Coverage Gradient

A quest to achieve higher BIST fault coverage with shorter test application time generated an immense amount of research in the past. Typically, LFSR-based pseudorandom test sequences were modified either by placing a mapping logic between the PRPG outputs and inputs of a circuit under test , or by adjusting the probabilities of outputting 0s and 1s so that the resultant vectors capture characteristics of test patterns for hard-to-detect faults, as done in various for weighted-random testing. Test patterns leaving a PRPG can also be transformed in a more deterministic fashion. Along the same lines, we will demonstrate that PRESTO-produced LP test patterns are also capable of visibly improving a fault coverage- to-pattern-count ratio. Assuming that the toggle control registers can also be driven by deterministic test data (see location of an additional multiplexer in the front of a shift register. test patterns can be produced with better-than-average fault coverage. The proposed method begins by computing the PRESTO parameters Given the PRESTO switching code, our goal is now to find the corresponding distribution of 1s in the control register that maximizes the fault detection probability.

The procedure starts by reducing each ATPG-produced test cube to a set of scan chains containing more than one specified bit. This set will be further referred to as a base. For example, let a test cube feature the following specified scan cells whereas is a scan chain, and c is a cell location within the scan chain.

The base is thus given by {4, 14}; note that chain 45 is not included as it features only one specified scan cell. A good chance (50%) of producing a given logic value in a purely pseudorandom fashion is a rationale behind excluding from any base scan chains hosting a single specified bit. As a result, more bases can be subsequently combined together to produce a single control setting. Given the phase shifter architecture, one can determine, for each base, the minimal number of phase shifter inputs—or equivalently the number of 1s in the toggle control register— required to activate the specified scan chains. These inputs are obtained by solving the minimum hitting set problem, where we find, in a greedy fashion, the minimal set of phase shifter inputs that intersects all subsets of phase shifter inputs capable of activating specified scan chains of a given base. Recall that the number of such inputs (and thus the number of 1s in the control register) is further constrained by the preselected switching code.

Let C be an initially empty set of bases. Once all weights are determined, we add to C a minimum-weight base. Next, every remaining base B is assigned a cost value, which is equal to the smallest number of 1s in the control register that would be required to activate all scan chains in {C ∪ B}.A minimum-cost base (or a minimum-weight base if there are two or more bases with the same minimal cost) is then added to C, and costs associated with the remaining bases are recomputed accordingly. The procedure continues until either the limit of 1s in the control register is reached or all bases are already in C. The control register content that activates all scan chains from C is then provided to PRESTO. For each control register setting, PRESTO is run to produce a certain number of pseudorandom test patterns. These patterns are subsequently fault-simulated, and detected faults are dropped from the list. Experimental results demonstrating feasibility of this method.

III. PROPOSED ARCHITECTURE

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.

A. BIST Architecture

A typical BIST architecture consists of

- TPG - Test Pattern Generator
- TRA – Test Response Analyzer
- Control Unit

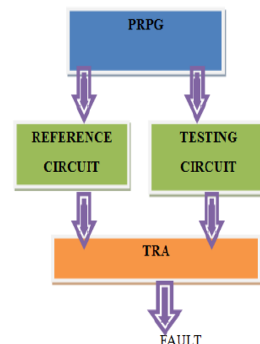

Fig.3.Test Pattern Generator

It generates test pattern for CUT. It will be dedicated circuit or a micro processor. Pattern generated may be pseudo random numbers or deterministic sequence. Here we are using a Linear Feedback Shift Register for generating random number. The Architecture for LFSR is as shown below.

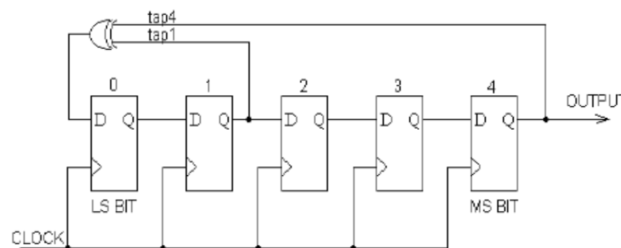

Fig.4. The Architecture for LFSR

Tapping can be taken as we wish but as per taping 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.

Test Response Analyzer (TRA): TRA will check the output of MISR & verify with the input of LFSR & give the result as error or not.

Circuit under Test (CUT): CUT is the circuit or chip in which we are going to apply BIST for testing stuck at zero or stuck at one error.

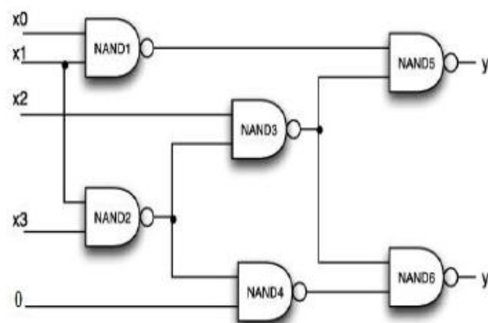

Fig.5. C14 Bench mark

B. 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 major problems detected so far are as follows:

- Test generation problems
- Gate to I/O pin ratio

• **Test Generation Problems:** The large number of gates in VLSI circuits has pushed computer automatic-test-generation times to weeks or months of computation. The numbers of test patterns are becoming too large to be handled by an external tester and this has resulted in high computation costs and has outstripped reasonable available time for production testing.

• **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.

C. Cyclic Analysis Test System (CATS)

Cyclic analysis test system (CATS) is a typical example of circular BIST. The architecture of CATS is shown in Fig.5. In test mode, the outputs are fed back to the inputs directly. The responses are used as the test vector without modification. If there are more inputs than outputs, one output may drive multiple inputs. If there are more outputs than inputs, we can use XOR gates to do space compression, as the one shown in Fig.5. The hardware overhead is very low. However, the fault coverage is circuit dependent. The recycling of test responses might create the fault masking effects. Note that, fault masking here is different from the aliasing discussed earlier. Here, the faulty and fault-free circuits have different test patterns.

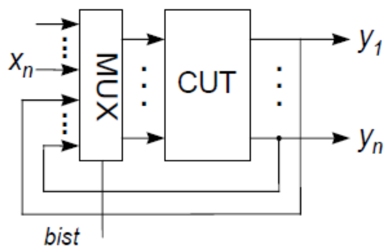

Fig.6. Cyclic analysis test system architecture

D. Random Test Data (RTD)

Random test data (RTD) transforms internal flip-flops into MISR. The circuit structure is shown in Fig.5. In normal mode, the MISR is operated as latches. In test mode, it operates as MISR. Both internal responses are compressed into and the internal test vectors are generated from the MISR. RTD is able to do one test per clock cycle. As compare to CATS, the hardware overhead is much higher. However, due to the extensive use of MISR, the test responses are scrambled before being used as the test patterns. Hence, the self masking probability can be lowered.

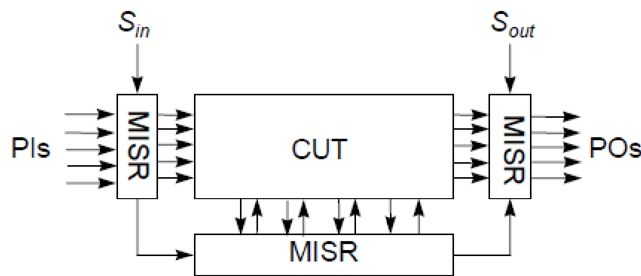

Fig.7. Random test data architecture

IV. RESULTS

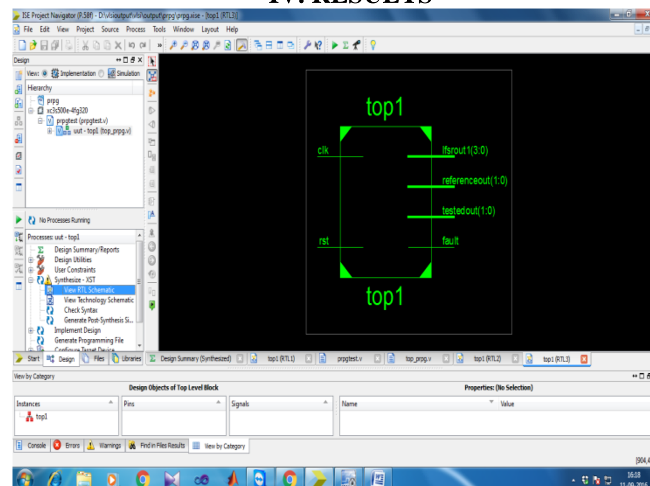

Fig.8. Schematic Diagram.

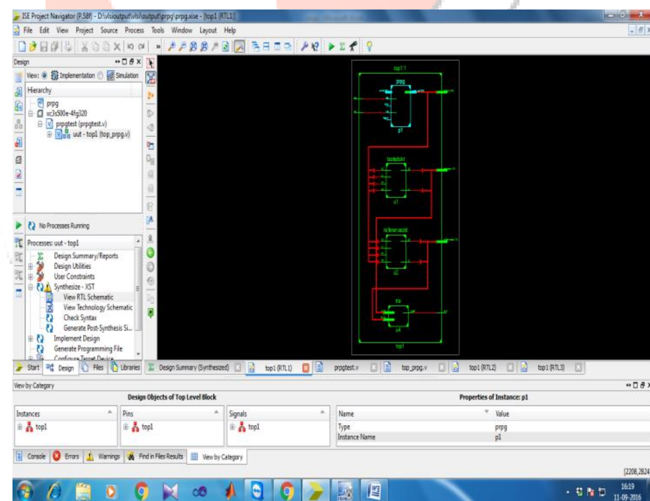

Fig.9. RTL Schematic.

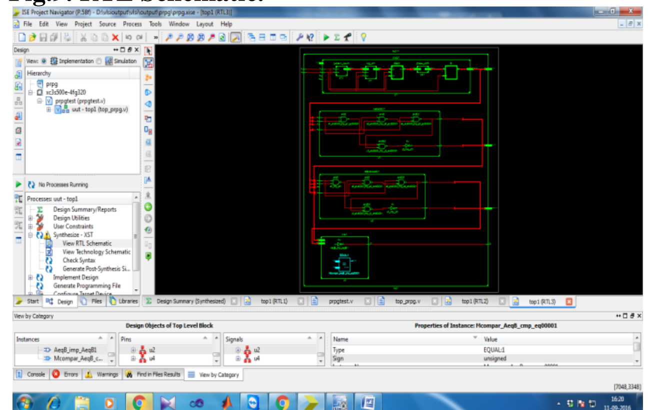

Fig10. Internal Structure of RTL Schematic.

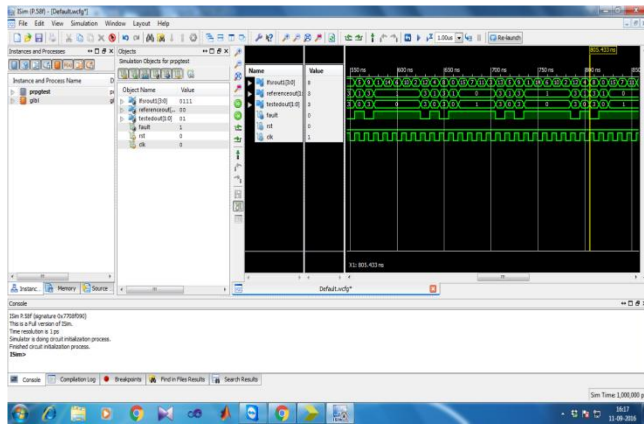

Fig11. Simulation Waveform.

V. CONCLUSION

The Project **"**Low Power Programmable PRPG with Test Compression Capabilities**"** is successfully designed and simulated. The proposed approach shows the concept of reducing the transitions in the test pattern generated. The transition is reduced by increasing the correlation between the successive bits. The simulation results shows that how the patterns are generated for the applied seed vector.

VI. REFERENCES

[1] Michał Filipek, Grzegorz Mrugalski, Senior Member, IEEE, Nilanjan Mukherjee, Senior Member, IEEE, Benoit Nadeau-Dostie, Senior Member, IEEE, Janusz Rajski, Fellow, IEEE, Je ̨drzej Solecki, and Jerzy Tyszer, Fellow, IEEE, "Low-Power Programmable PRPG With Test Compression Capabilities", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 23, No. 6, June 2015.

[2] A. S. Abu-Issa and S. F. Quigley, "Bit-swapping LFSR for low-power BIST," Electron. Lett., vol. 44, no. 6, pp. 401–402, Mar. 2008.

[3] C. Barnhart et al., "Extending OPMISR beyond 10x scan test efficiency," IEEE Design Test, vol. 19, no. 5, pp. 65–73, Sep./Oct. 2002.

[4] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy, "Low-power scan design using first-level supply gating," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 3, pp. 384–395, Mar. 2005.

[5] M. Chatterjee and D. K. Pradham, "A novel pattern generator for near perfect fault-coverage," in Proc. 13th IEEE Very Large Scale Integr. (VTSI) Test Symp., Apr./May 1995, pp. 417–425.

[6] F. Corno, M. Rebaudengo, M. S. Reorda, G. Squillero, and M. Violante, "Low power BIST via non-linear hybrid cellular automata," in Proc. 18th IEEE Very Large Scale Integr. (VTSI) Test Symp., May 2000, pp. 29–34.

[7] D. Das and N. A. Touba, "Reducing test data volume using external/LBIST hybrid test patterns," in Proc. Int. Test Conf. (ITC), 2000, pp. 115–122.

[8] R. Dorsch and H. Wunderlich, "Tailoring ATPG for embedded testing," in Proc. Int. Test Conf. (ITC), 2001, pp. 530–537.

[9] M. Filipek et al., "Low power de-compressor and PRPG with constant value broadcast," in Proc. 20th Asian Test Symp. (ATS), Nov. 2011, pp. 84–89.

[10] S. Gerstendorfer and H. Wunderlich, "Minimized power consumption for scan-based BIST," in Proc. Int. Test Conf. (ITC), 1999, pp. 77–84.

[11] V. Gherman, H. Wunderlich, H. Vranken, F. Hapke, M. Wittke, and M. Garbers, "Efficient pattern mapping for deterministic logic BIST," in Proc. Int. Test Conf. (ITC), Oct. 2004, pp. 48–56.

[12] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design," in Proc. 17th IEEE VLSI Test Symp. (VTS), Apr. 1999, pp. 407–412.