A Survey Paper on Design of STUMPS based Logic BIST Architecture

Meha M. Panchal[1] Kirit V. Patel[2]
[1, 2]Electronics & Communication Engg. Department
[1, 2]V.G.E.C, Chandkheda.

Abstract---Built-in self-test for logic circuits or logic BIST, is an effective solution for the test cost, test quality, and test reuse problems. The scheme is based on STUMPS (Self Test Using MISR and Parallel Shift register) architecture which uses an on-chip circuitry to generate the test patterns and analyze the responses with no or little help from an ATE. External operations are required only to initialize the Built-in tests and to check the test results. Expected output includes the designed STUMP based architecture, Which is test per scan using verilog HDL and simulation result having information of design is defect free or not.

Key words: BIST, LFSR, MISR, SRSG, SISR, CUT, TPG, ORA, Logic BIST, STUMPS, at speed testing, scan test, PRPG, fault coverage, DFT.

I. INTRODUCTION

As[6] the VLSI designs increase in integration density, associated modules becomes inaccessible and testing of the chip becomes more challenging. The test industry is facing with varied problems such as increase in test time and test volume when external ATE is employed for testing. To deal with this, in-system test generation and test application were introduced through built in self test architectures by the testing community. Test generation hardware and embedded algorithms reduce the need for test access for system-wide built-in self-test (BIST), which is a preferred DFT approach.

II. LOGIC BIST

Logic built-in self-test, or LBIST, is a mechanism that permits an integrated circuit to test the integrity of its own digital logic structures. LBIST operates by stimulating the logic-based operations of the integrated circuit and then detecting if the logic behaved as intended. LBIST is one of several approaches to Design For Test (DFT). It is one of numerous types of built-in self-test. It typically tests a wide range of logic structures wired in a highly randomized fashion, and so requires more complex test circuitry.

A. STUCK AT FAULTS

The most popular model [7], called the 'Stuck- At' model, models a faulty gate input as a "stuck at zero" or "stuck at one". These faults most frequently occur due to thin-oxide shorts or metal-to-metal shorts.

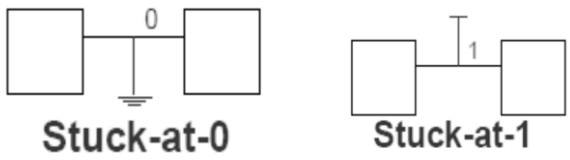


Fig.1: Stuck at Faults

B. BRIDGING FAULTS

This type of fault occurs when signals are connected together when they should not be. There are many implementations of LBIST, but almost all depend on generation of a pseudo-random sequence as stimulus for the design, while the response to this stimulus is captured in a MISR (Multiple Input Shift Register). This MISR is used to generate a "signature". This signature is unique in the sense that each failure in the device would lead to a different value at the end of the whole process.

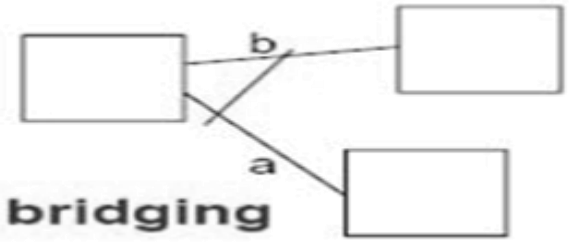


Fig. 2: Bridging Fault

III. BIST ARCHITECTURE

The components that the logic BIST is comprised of pseudo-random pattern generator (PRPG) and the phase shifter circuit. The output response analysis block - composed of multiple input signature register (MISR) and the signature analyzer. The most popular among the L-BIST architectures is the STUMPS (Self-Test Using a MISR and Parallel Shift register sequence generator) architecture. Logic Built-In Self-Test (BIST) schemes based on the STUMPS structure use on-chip circuitry to generate test stimuli and analyze test responses, with little or no help from an ATE. The STUMPS structure applies pseudo-random patterns generated by a PRPG (Pseudo-Random Pattern Generator) to a full-scan circuit in parallel and compacts the test responses into a signature with a MISR (Multiple-Input Signature Register). This approach has advantages such as simple test interface, better test quality, lower test cost, and higher reliability.

BIST architecture [7] most widely used to apply patterns and observe responses on a chip is the self-testing using MISR and parallel shift (STUMPS) register sequence generator.

The basic mechanism uses a pseudo-random pattern generator (PRPG) to generate patterns which form the inputs to the device's internal scan chain, initiate a functional cycle to capture the response of the device, and then compress the captured response with the help of a multiple input signature register (MISR). The compressed response that comes out of the MISR is called the signature. Any corruption in the output signature indicates a defect in the device.

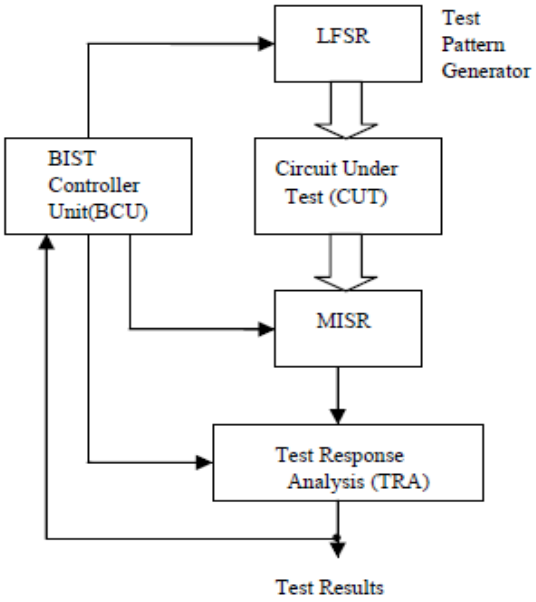


Fig.3:BIST Implementation

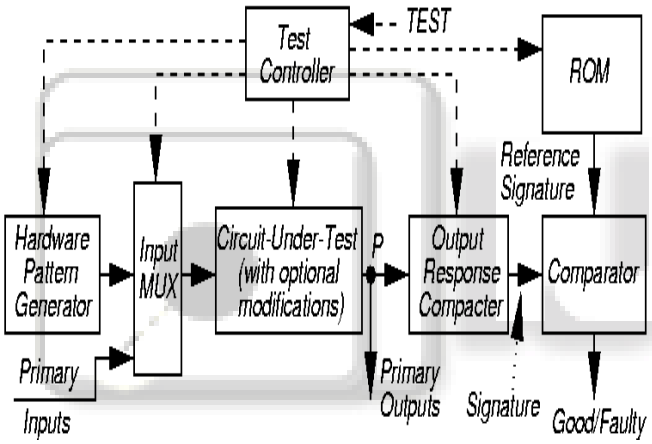


Fig.4: BIST Architecture

Four primary parameters must be considered in developing a BIST methodology for embedded systems.

- *Fault coverage*: This is the fraction of faults of interest that can be exposed by the test patterns produced by pattern generator and detected by output response monitor. In presence of input bit stream errors there is a chance that the computed signature matches the golden signature, and the circuit is reported as fault free. This undesirable property is called masking or aliasing.
- *Test set size*: This is the number of test patterns produced by the test generator, and is closely linked to fault coverage: generally, large test sets imply high fault coverage.
- *Hardware overhead*: The extra hardware required for BIST is considered to be overhead. In most embedded systems, high hardware overhead is not acceptable.
- *Performance overhead*: This refers to the impact of BIST hardware on normal circuit performance such as its worst-case (critical) path delays. Overhead of this type is sometimes more important than hardware overhead.

IV. BIST Pattern Generation

1) *LFSR:* Another method is to use a linear feedback shift register (LFSR) to generate pseudo-random tests. This frequently requires a sequence of 1 million or more tests to obtain high fault coverages, but the method uses very little hardware and is currently the preferred BIST pattern generation method.
2) *LFSR and ROM:* One of the most effective approaches is to use an LFSR as the primary test mode, and then generate test-patterns with an ATPG program for the faults that are missed by the LFSR sequence. These few additional test-patterns can either be stored in a small ROM on the chip for a second test epoch, they can be embedded in the output of the LFSR, or they can be embedded in a scan chain in order to augment the stuck-fault coverage to 100%.

A. Linear Feedback Shift Register

A linear feedback shift register (LFSR) can solve some of the problems with the counters. When an *n*-bit binary counter produces 2*n* unique input vectors, an LFSR can be made to generate up to2^n -1 unique pseudo-random test vectors .An LFSR consists of a series of flip-flops wired as a shift-register with feedbacks through XOR gates. The XOR gates are modulo-2 adders, and the flip-flops are considered as delay elements.

B. LFSR Characteristic Equation

Data produced by an LFSR are based on what is referred to as its characteristic equation, which is defined by the way its feedback is formed. We represent the time component coefficient of each data by *x*, thus *xn* becomes the coefficient of data at time *n*, and *x*0 becomes the coefficient of data *n* clock cycles back. Therefore, the characteristic

Polynomial that defines the data produced by this register becomes:

$$P(x) = x0 + xn = 1 + xn \quad (1)$$

Based on the above discussion, the polynomial for the LFSR shown in Fig.5 [4] is:

$$P(x) = 1 + x2 + x3 \quad (2)$$

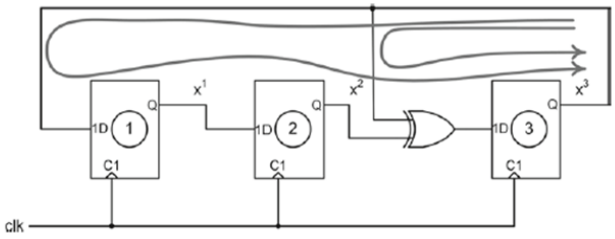


Fig.5:A third degree standard LFSR

As shown in this figure, in addition to the feedback from the right-most flip-flop (number 3) to the left-most flip-flop, there is also a feedback that adds the output of flip-flop 2 to data coming into flip-flop 3. Considering that the XOR behaves as a modulo-2 adder, data at the output of flip-flop 3 become the superposition of all data arriving at this point through various feedback paths. This explains the addition of *x*2 in E.q. 2, when compared with E.q. 1.

C. Standard LFSR

Figure 6 [4] shows an LFSR type that is referred to as a standard or external-XOR LFSR. This circuit has feedbacks from flip-flop stages back to the left-most flip-flop. Because the XOR gates are outside of the shift-register, this structure is also referred to as external-XOR LFSR .The structure in this figure has XOR gates (modulo-2 adders) in the feedback path and switches that turn feedback contributions on or off. An XOR gate with a switch input in the off (0)position is effectively removed from the feedback path.

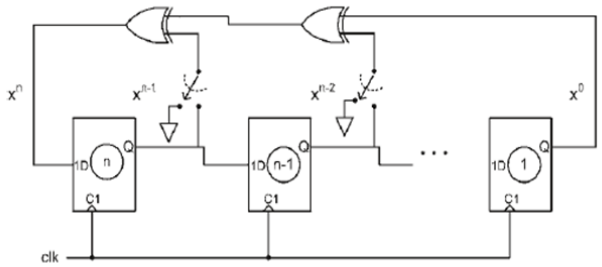


Fig .6: Standard LFSR

Equation 3 is the general form of the polynomial for this LFSR. The h parameters represent the switch positions, where h_n and h_0 are always 1.

$$P(x) = x^n + h_{(n-1)}x^{(n-1)} + h_{(n-2)}x^{(n-2)} + \ \dots\ + 1. \tag{3}$$

Figure 6 shows an LFSR of this type and its corresponding polynomial. Starting with any non-zero initial value, this LFSR cycles through all 3-bit combinations. An example is shown in this figure.

D. Period of LFSR

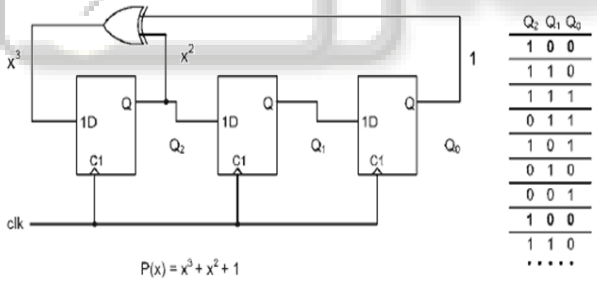
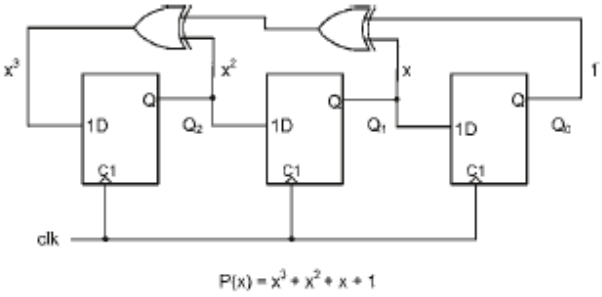


Fig.7**:**A third degree Standard LFSR

$$\begin{array}{r|l} & x^4+x^3+x^2+1 \\ \hline x^3+x^2+1 & \big) \ x^7+1 \\ & \ \ x^7+x^6+x^4 \\ \hline & \ \ x^6+x^4+1 \\ & \ \ x^6+x^5+x^3 \\ \hline & \ \ x^5+x^4+x^3+1 \\ & \ \ x^5+x^4+x^2 \\ \hline & \ \ x^3+x^2+1 \\ & \ \ x^3+x^2+1 \\ \hline & \ \ 0 \end{array}$$

Fig. 8**:** Polynomial division for periodic check

Polynomials can be divided and multiplied in modulo-2 system. If polynomial of an LFSR divides the polynomial, x^T+1 (remainder is 0), and T is the smallest positive number for which this is true, then T is the period of the polynomial. Figure 8 [4] shows dividing x^7+1 by x^3+x^2+1, which is the polynomial of LFSR of Fig. 4. Since this division has no remainder, the period of this LFSR is 7. This can be verified by the sequences shown in Fig. 5(the 100 pattern repeats after 7 clock cycles).



$$\begin{array}{c|c} Q_2\ Q_1\ Q_0 & \\ \hline 1\ 0\ 0 & \\ 1\ 1\ 0 & \\ 0\ 1\ 1 & \\ 0\ 0\ 1 & \\ 1\ 0\ 0 & \\ \vdots & \end{array} \qquad \begin{array}{r|l} & x+1 \\ \hline x^3+x^2+x+1 & \big) \ x^4+1 \\ & \ \ x^4+x^3+x^2+x \\ \hline & \ \ x^3+x^2+x+1 \\ & \ \ x^3+x^2+x+1 \\ \hline & \ \ 0 \end{array}$$

Fig. 9**:** Third degree polynomials with T=4

Figure 9 [4] shows another LFSR with a different polynomial. The sequences this LFSR cycles through are also shown. The smallest positive integer, T, with which x^T+1 can be formed such that it can be divided by the polynomial of Fig. 6 is 4. Thus the period of this LFSR is 4, which is also verified by the sequences shown. In an n-stage LFSR, $T = 2^n - 1$, then the LFSR with this period produces all possible n-bit combinations except 0. This kind of LFSR is called maximum length LFSR.

A maximum-length LFSR has a good randomness in terms of frequency of 1s and 0s for each bit, and for this reason is a good source of test data for data inputs of combinational circuits. The characteristic polynomial of a maximum-length LFSR is called a primitive polynomial.

V. Output Response Analysis

On the one hand, TPGs described in Sect. VI generate test data for on-chip BIST. ORAs, on the other hand, compress responses from a CUT and make them available for the BIST to analyze. We have to have ORA for the BIST to just check a short signature of all the responses, instead of checking every response of the CUT for the test data that is applied to it. As with TPGs, ORAs must be brief in use of hardware to fit on the same chip as the rest of the BIST hardware, and of course, the CUT itself. ORAs must be efficient in compressing data so that different sets of test response vectors compress to different values, i.e., have low aliasing. In this section, we discuss various hardware structures for compressing a CUT's response to input test data.

A. *Serial LFSRs (SISR)*

A method of generating a signature from a CUT's single-bit output is to shift the serial output into an LFSR with serial input.

When used as ORA for collecting a signature from a serial data input, this structure is referred to as serial input signature register (SISR). The signature of serial data coming in this SISR via sin will be contained in the register parallel outputs. This signature can be calculated by dividing the polynomial corresponding to the input sequence by the LFSR polynomial. The register output will go to the BIST compare component, where it will be compared with the predetermined expected signature for the CUT's golden model.

B. *Parallel Signature Analysis*

Another ORA that is inherently parallel and is a better fit for calculating signatures from bus outputs of CUTs is a multiple input signature register (MISR) [4].

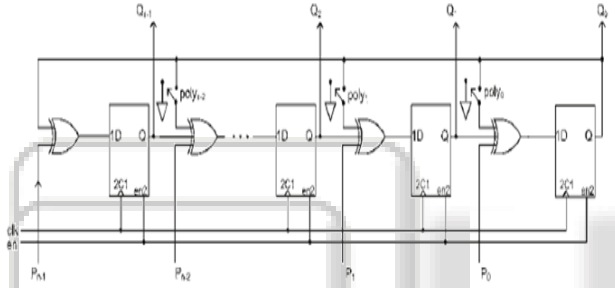


Fig.10: An n bit MISR

Figure 10 shows an *n*-stage MISR.Parallel data inputs to be compressed (*P[n-1:0]*) are inputs of the XOR gates at the flip-flop inputs. Switches at the XOR inputs determine whether the feedback exists or not, i.e., removes the corresponding polynomial term if set to 0. Therefore, the XOR gates are always there, some have three inputs for feedback, and others are just 2-input gates when feedback is off. For control by the BIST controller, the MISR shown has an enable input that tells it when to participate the incoming parallel data in the formation of the signature. The signature appears at the *Q* outputs.

VI. Self-Testing Using MISR and Parallel Shift Register Sequence Generator (STUMP)

The most widely used BIST in industry today is self-testing using MISR and PRPG (SRSG) (STUMPS) This BIST architecture solves the problem of long internal scan chain of a CUT (as in RTS and LOCST) by splitting the internal scan into several individually accessible scan chains. Preferably, the scan chains should be of equal or close lengths, but there is no penalty if this is not done.

A. *STUMPS Structure*

Figure 11 [4] shows a generic form for the application of STUMPS BIST method to the Huffman model of our CUT. As shown three separate scan chains are used here. The serial input of each scan chain is driven by a bit of the parallel output of a PRPG. This result in pseudo-random serial test data shifted into the scan chain. The serial outputs of the internal CUT scan chains drive the bits of the parallel input of an MISR.

The block diagram shown in Fig. only deals with pseudo inputs and pseudo outputs, and it does not specify how primary input test data are applied and how primary output test response values are read. STUMPS leave these issues to be decided by the individual implementations.

However, the boundary scan here is to be used by an ATE to scan in test data into primary input cells and scan out contents of primary output cells. Extending this option one step further, the ATE can also provide LFSR configuration data and seeds for the STUMPS PRPG and MISR.

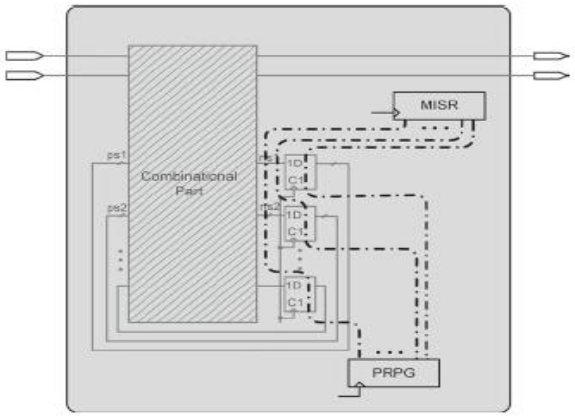


Fig.11**:** General structure of STUMPS

The architecture of the self-testing using MISR and parallel SRSG (STUMP) is shown in Figure. Instead of using only one scan chain, it uses multiple scan chains to minimize the test time. Since the scan chains may have different lengths, the LFSR runs for N cycles (the length of the longest scan chain) to load all the chains. For such a design, the internal type LFSR is preferred. If the external type is used, the difference between two LFSR output bits is only the time shift. Hence, the correlation between two scan chains can be very high.

B. *STUMPS Test Process*

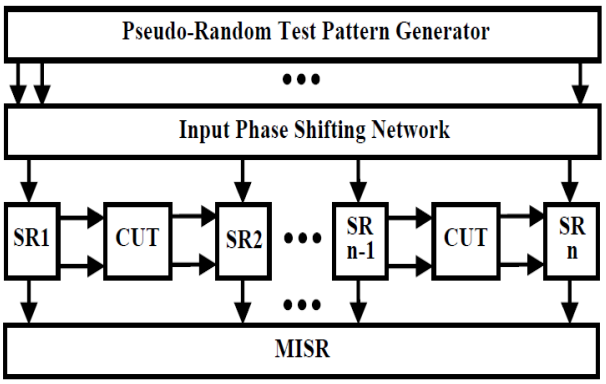


Fig. 12: STUMPS Architecture

Depending on implementation [1], STUMPS test process is similar to RTS or LOCST. The only difference is in the number of clocks that is needed in each test cycle to shift pseudo random serial data into the CUT's internal registers.

Obviously, fewer clock cycles are needed because several registers are receiving test data in parallel

1) Scan in patterns from LFSR to all scan chain.
2) Switch to normal function mode and apply one clock.
3) Scan out chains into MISR.
4) Overlap steps 1 and 3.

C. STUMPS Features

In board level testing, various STUMPS scan registers correspond to internal scan of individual chips. At the chip level, STUMPS registers are segments of the chip's scan register. The biggest advantage of STUMPS is that it uses significantly less clock cycles per test cycle.

VII. CONCLUSION

The process of testing involves generating pseudo-random patterns on the chip using LFSR, passing the patterns through the DUT, compact the responses from the DUT into signatures, passing it through the signature analyzer for analyzing these signatures to match with the golden values and finally giving out the pass or fail signal.. For a particular IC, the size is fixed and hence with just the clock signal and the test mode signals the particular IC can be tested. The output is also simplified with just the single signal indicating the pass or fail status after the signature analysis.

REFERENCES

[1] Bushnell and V.D. Agarwal, "Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits "Kluwer Academic Publishers, 2000.
[2] L.T. Wang, Cheng- Wen Wu and Xiaoqing Wen, " VLSI Test Principles & Architectures Design for testability"
[3] MIRON ABRAMOVICI, AT&T bell Laboratories IEEE Revise printing "Digital System Testing And Testable Design" 1990 by AT&T.

