Feedback Shift Register Based Cryptographic System using LBIST Method

Jyothi.M.S\textsuperscript{1}, Latha.K\textsuperscript{2}, M N Eshwarappa\textsuperscript{3}
\textsuperscript{1}PG student (VLSI Design and Embedded systems), SIET, Tumkur, Karnataka, India
\textsuperscript{2}Assistant Prof., Dept. of E&C, SIET, Tumkur, Karnataka, India
\textsuperscript{3}HOD, Dept. of E&C, SIET, Tumkur, Karnataka, India

Abstract— Cryptographic systems are used to protect confidential information against modification done by unauthorised sources. The cryptographic systems make the systems immune to all types of random attacks and modifications done by malwares. The system is designed to respond dynamically to the fault scenario irrespective of the faults that occurs. There are several types of faults can be inserted to functional hardware, considering the hardware Trojan which is a major challenge when equipping the security for any system. Hardware Trojan is an unauthorized modification or malicious addition to the original functional circuit elements or hardware system. The traditional self testing methods are fairly detects random faults in circuit under test, but they are fail provide complete protection against unauthorized modification of a circuit known as stuck at fault type of hardware Trojans. This paper proposes a new method of LBIST for FSR- based cryptographic systems which can detect such modification done by Trojans. FSR based cryptographic systems are most efficient and fastest cryptographic systems for hardware applications.

Keywords— Logic built in self test, feedback shift register, algebraic normal form, response analyzer, test pattern generator, multiple input signature register.

I. INTRODUCTION

The major deal of cryptographic system is to make the systems immune to all possible random faults and attacks from unauthorised source. To make possible periodic fault detection during the lifetime of functional circuits, cryptographic systems often employ logic built-in self-testing method. Traditional LBIST techniques have a limited use against malicious modification of the original circuit known as hardware Trojans. The Trojan can be inserted into the LBIST itself and also Trojan can be designed not to trigger the LBIST, since LBIST usually detects only a subset of all possible faults. Hardware Trojan is an unwanted addition or unauthorised modification to the existing functional circuit elements. It can change the functionality of the circuit, leak confidential information, compromise the reliability and do malicious alterations for existing system or circuit. Applications that are likely to be hit by these malicious faults are Military applications, Aerospace applications, Civilian security-critical applications, financial applications, Transportation security. The traditional testing methods can detect random faults, but they are not able to provide complete protection against all possible malicious alterations of a circuit known as hardware Trojans. Conventional LBIST method may fail to detect even the simple case of stuck-at fault inserted by Trojans. This paper proposes a new efficient method for LBIST which can detect all possible stuck-at fault type of Trojans in hardware system. It uses a deterministic test set which covers 100% of single stuck-at faults in the circuit under test and LBIST itself that is, in TPG and in TRA and also it do not compact output responses into a MISR signature. In case of traditional testing method, detection of Trojan is based on the value of MISR signature generated. It is also possible that Trojan can produce correct MISR signature for input provided to circuit under test, detection of Trojan is impossible in this case so proposed method is designed to overcome this problem by analysing input pattern itself.

II. PROPOSED SYSTEM

The self testing method proposed in this paper for LBIST makes possible detecting stuck-at fault type of Trojans. The proposed method specifically uses FSR to test circuit under test by applying deterministic test patterns. The proposed method is shown in Fig. 1. It operates in two modes: Normal mode and test mode, operational modes are controlled by test controller. The operation is explained below.

Normal mode:

Initially the FSR based LBIST system is in state of normal mode, where the normal data is given as an input to the LBIST block; it will be the input for FSR. The data input to FSR is selected by test enable signal by test controller. In normal mode, the test enable signal is generated by applying ‘high reset’ and ‘clock’ signals to the test controller block, when reset is high and the test enable signal is low and the circuit operates in normal mode. The multiplexer selects normal data and it is shifted through FSR and analysed using TRA

Test mode:

The self testing operation takes place in the test mode, this means tester itself generate deterministic test patterns to analyse the circuit under test. When ‘reset’ goes low the ‘test enable’ signal is high. Test enable signal is given to multiplexer at the input of the FSR; it selects test data generated by the test pattern generator. Test data is then shifted and tested by setting and observing each flip-flop in a FSR.
There are various methods to generate test patterns; in this project, the pattern generator uses the feedback shift register and XOR gates in the feedback path to generate deterministic test sets. The test pattern generator circuit is implemented based on the equation called algebraic normal form (ANF). Detailed test pattern generation process is explained in the following section.

The next step is comparison of expected and computed responses; it is done using Test Response Analyser (TRA). There are many ways to analyse the response in TRA. In the proposed method, TRA block is implemented using storage elements and comparator. Storage elements are used to store the expected result, and comparator used to compare the bits of expected response and actual response. Output of TRA indicates presence or absence of fault circuit under test.

The fault can also be inserted into TPG and TRA itself. The Trojan block generates three trigger pulses to insert fault in corresponding blocks as Fig. 1. The faults inserted in these blocks are detected by LBIST system and it should indicate by three respective fail signals, this is a final output of LBIST block.

III. DESCRIPTION FOR INTERNAL BLOCKS

A. Test Pattern Generator

Pattern generator is the algebraic normal form. This ANF generates a pattern depending on the equation which is generated based on number of flip-flops used to generate pattern. According to the logic of FSR, patterns are generated using a cryptography concept which is algebraic normal form. ANF is a common method of representation for Boolean functions used in cryptographic systems. Combinational logic circuit is implemented using this n-variable Boolean function represented in ANF form. The test sets generated by this logic is tested for all single stuck-at faults. To be hardware efficient, cryptographic systems typically use ANF of a small size.

The Boolean function representing combinational logic has a unique Algebraic Normal Form (ANF) which is a representation of type,

\[ f(x_1, \ldots, x_n) = \sum_{i=0}^{2^n-1} c_i \cdot x_1^{i_1} \cdot x_2^{i_2} \cdot \ldots \cdot x_n^{i_n} \]  

(1)

Where \( c_i \) is constants, \( \cdot \) stands for the Boolean AND operation, \( \sum \) stands for the Boolean XOR. The term \( x_i \) denotes the variables.

For example, consider 8 bit Feedback Shift Register (FSR) ANF

\[ f_6 = x_3 \oplus x_7 \]  

(2)

Where \( \oplus \) is the Boolean XOR. The function \( f_6 \) can be implemented by a circuit shown in Fig. 4.

![Fig. 1 Block diagram of proposed system with input and output](image1)

![Fig. 2 Modifications of FSR flip-flops to support test mode](image2)

The initial value of the LFSR is chosen by the designer and the operation of producing pattern is deterministic, the stream of bits called test sets produced by the register is determined by the current or previous state of register. However, by choosing a proper initial state an LFSR can produce small deterministic test sets.

B. Linear Feedback Shift Register

The series of shift register is used to implement an LFSR whose input bit is depends on previous state of the flip-flop. An n-bit LFSR consists of single bit storage elements, called cells or stages as shown in Fig. 5. To support the test mode, multiplexer is connected at the input of FSR. Modifications of FSR flip-flops to support test mode is shown in Fig. 6. When the test mode is selected, the flip-flops with multiplexed inputs become inputs to the FSR. Multiplexer selects test input as input to FSR depending on the value of test enable signal generated by test controller. When test enable signal is high FSR shifts the pattern generated by test pattern generator otherwise it shifts the 8-bit input given by user to LBIST block. Since FSR is operating in test mode logic of the FSR is tested with test pattern generated by TPG and patterns are analysed by TRA to make sure there are no faults in FSR which can be inserted by the Trojan.

C. Test Controller

This generates a test signal test enable which indicates whether circuit is in normal mode or test mode. Operations starts by receiving enable signal from test controller. All blocks in LBIST system operates in test mode.

D. Test Response Analyser

The comparison of expected and actual responses can be done using TRA shown in Fig. 8 and Fig. 9. For each observable cell the value at the test output is compared to the expected value using a comparator. Output of the comparator indicates the presence/absence of a fault in any functional block. There are different fail signals generated by the LBIST to indicate fault in certain block. The fault can be inserted into TRA itself; in this case the fault signal from Trojan circuit triggers LBIST system to analyse the TRA for fault detection.

This compares Fsr_out with expected output to see that no fault is occurred in the functional block. Expected output is pre determined and stored in TRA at the same time when TPG generates pattern. If the both data are not equal then it raises a fail signal, if they are equal it raises a pass signal.

E. Trojan Circuit

Trojans is a program that appears to do a favourable task but in reality it performs undisclosed malicious functions that allow the attacker to gain unauthorized access to the functional circuit or cause damage to the circuits. Here the main concept is to detect unauthorized modification or alteration caused by the Trojans in functional circuits such as test pattern generator, feedback shift register and test response analyser.

In proposed method Trojan is designed to affect three functional blocks. To affect these blocks Trojan circuit generates three triggering signals as shown in Fig. 10.

There are three trigger pulses applied to functional blocks which alter the functionality of that block. Trigger1 pulse is applied to test pattern generator block and it leads to generate

---

Fig. 5 General structure of Feedback Shift Register

Fig. 6 Modifications of FSR flip-flops to support test mode

Fig. 7 Test Controller

Fig. 8 Test Response Analyser

Fig. 9 Internal Block Diagram of Test Response Analyser

Fig. 10 Trojan circuit
wrong pattern which have to be detected by the test response analyzer. Trigger2 pulse is applied to FSR block and it make FSR test wrong pattern even in the case where test pattern generator generates correct pattern. In this case TRA detects fault in FSR by comparing FSR output with pre determined pattern which is already stored in TRA. In this case Trojan makes any of the FSR bit to stuck at ‘1’. Trigger3 pulse applied to test response analyzer. Trojan alters the TRA and makes it to produce wrong output where TRA indicate fault signal even in case where no fault is present in functional circuits.

IV. FAULT DETECTION

This section shows that it is possible to detect all single stuck at faults in TPG, FSR and TRA of LBIST system, the RTL schematic of top LBIST block is shown in Fig. 11. For each functional block different trigger pulse is applied and different fault signal indicating the fault in particular block. The procedure for detecting faults in functional circuits is explained below.

A. Detecting fault in test pattern generator

Consider the case when a fault occurs at the output of TPG, which change the functionality of TPG by triggering signal from Trojan circuit. Trojan makes the output of the TPG is equal to inverse of the predetermined test pattern. The wrong pattern should be tested identified by TRA. Fault in TPG will be detected by procedure 1.

Procedure 1:
1) Assert test enable signal high to connect test input to FSR.
2) Set enable signal to high and reset signal to low to make TPG functional.
3) Apply clock to all flip flops of TPG and FSR in parallel.
4) Apply fault trigger pulse “hi” to TPG and TPG will generate wrong pattern at its output which is inverse of the expected pattern as shown in Fig. 12.
5) Pattern at the output of the TPG will be compared with expected output of TPG in TRA
6) If two patterns are equal top LBIST block generates “Pass” signal otherwise it outputs “Fail” as shown in Fig. 13.

B. Detecting fault in FSR

Consider the case when a fault occurs at the output of FSR, Trojan circuit triggers the signal to insert fault in FSR which makes any of the bit in FSR stuck at ‘1’. This fault is detected in TRA by comparison. TRA not only detect single stuck, it can detect stuck in multiple bits by using multiple XOR operation. Each FSR output bit is XOR with the bit of predetermined pattern. Detection of fault in FSR is done by procedure 2 shown below.

Procedure 2:
1) Assert test enable signal high to connect test input to FSR.
2) Set enable signal to high and reset signal to low to make TPG functional.
3) Apply clock to all flip flops of FSR in parallel.
4) Apply fault trigger pulse “high” to FSR and FSR shift the bit and output is given to TRA.
5) Output of the FSR will be the stuck at fault when Trojan is triggered as shown in Fig.14, this is compared with input of the FSR in TRA.
6) If two patterns are equal TRA generates “Pass” signal otherwise it outputs “Fail1” as shown in Fig.15.
C. Detecting fault in test response analyzer

Consider the case when a fault occurs at the output of TRA, when fault is inserted by triggering signal from Trojan circuit, the functionality of response analyzer changed as the output of TRA will be inverse of actual output. This makes the TRA to indicate fault even when no fault is present and vise versa. Detection of fault in TRA is done by procedure 3 shown below.

Procedure 3:
1) Assert test enable signal high to connect test input to FSR.
2) Set enable signal to high and reset signal to low to make TPG functional.
3) Apply clock to all the functional blocks of LBIST in parallel.
4) Apply fault trigger pulse “high_comp” to TRA block, then comparator generates fault comparison output as shown in Fig. 16.
5) Then top LBIST block outputs “Fail2” signal indicates fault in TRA as shown in Fig. 17.

V. CONCLUSIONS AND OPEN PROBLEMS

The proposed algorithm is converted into an optimized digital hardware circuit. The hardware is implemented on the Xilinx Spartan3 FPGA using Verilog HDL. Feedback Shift Register (FSR) based cryptographic systems are the fastest and the most power-efficient cryptographic systems since it is designed by using simple combinational and sequential components and most of the circuit components will operate in single clock cycle hence the delay will be small. Proposed method for LBIST makes possible detecting stuck-at fault type of Trojans by using small deterministic test sets generated by test pattern generator with appropriate seed selection.

The presented method has several advantages compared to the traditional pseudo-random pattern-based LBIST using scan,
1) It causes no performance degradation. Delay of testing hardware will not affect the delay of the chip.
2) The set of test patterns required is small and easy to handle by test hardware.
3) It requires a small set of deterministic tests to cover 100% of single stuck-at faults.
4) The test execution time is much shorter that is at least two orders of magnitude of conventional method.
5) It has a higher resistance against stuck-at fault type of hardware Trojans Adversary attacks. That is no stuck at fault in the hardware circuit can go undetected.
6) Using this method exact fault location can be determined, which will help in redesigning of faulty hardware
7) Time and cost to redesign can be reduced effectively
No single method can protect against all possible types of adversary attacks. The proposed method may fail to detect some types of Trojans, e.g. parametric Trojans which change the parameters of a circuit rather than its functionality. Finding adequate protective mechanisms against parametric Trojans remains a topic of future work. The speed and the power are two crucial factors for future cryptographic systems,
since they are expected to support very high data rates in 5G
ultra-low power products and applications.

ACKNOWLEDGMENT

I would like to thank Latha.K, Assistant Professor, department of Electronics and Communication Engineering, for accepting to be my internal guide and for her constant and valuable guidance in preparing this paper and I express my deep sense of gratitude to Dr. M.N. Eshwarappa, Professor and Head, department of Electronics and Communication Engineering for providing me the support required for preparing this paper. I take this opportunity to extend my whole heartfelt thanks to all teaching staff of Electronics and Communication Engineering department.

REFERENCES