Masters Project Thesis

Some Issues in Scan Based Testing

Author:

Mukesh Agrawal

Roll No: 03CS3008

Supervisors:

Prof. Dipanwita Roy Chowdhury

Prof. Indranil Sengupta

Department of Computer Science & Engineering

Indian Institute of Technology

Kharagpur

May 6, 2008

" YOGA KARMASU
KAUSALAM "

CERTIFICATE

This is to certify that this thesis entitled “**Some Issues in Scan Based Testing**” submitted by **Mr. Mukesh Agrawal** to the **Department of Computer Science & Engineering, Indian Institute of Technology Kharagpur** in partial fulfillment of the requirement for the degree of Masters of Technology during the academic year 2007-2008 is a record of authentic work carried by him under my supervision and guidance.

May 6, 2008
IIT Kharagpur

Prof. Dipanwita Roy Chowdhury
Professor
Department of CSE
IIT Kharagpur

Prof. Indranil Sengupta
Professor
Department of CSE
IIT Kharagpur

Acknowledgement

I would like to take the opportunity to express my gratitude to some people who were involved in this project. First, I owe my thanks to Professor Dipanwita Roy Chowdhury and Professor Indranil Sengupta for doing everything from the inception of the project idea to giving invaluable suggestions at every step. I would also thank all my dual degree classmates, especially Sankalp Agarwal and Tathagata Das for motivating me all the time whenever I needed them and giving me useful tips on how to use Latex. I am also grateful to all the faculty members and staff of the Department of Computer Science and Engineering for being very supportive to me.

May 6, 2008
IIT Kharagpur

Mukesh Agrawal
03CS3008
Dual Degree Student
Department of CSE
IIT Kharagpur

Contents

List of Figures

List of Tables

Abstract

Scan based design for testability (DFT) is a famous and powerful testing mechanism. While testing, power is dissipated due to transitions in the scan flip-flops. Reducing testing time is a major concern among the test engineers. Increasing test concurrency reduces the test application time at the cost of extra power dissipation. The problem becomes even more challenging if testing is extended to System-on-chip where along with power constraint, there is a pin constraint too. Power profiling is a tool for estimating the power dissipated while testing and helps in optimized test scheduling under power constraint. This work includes development of a hybrid model for power estimation which is working better than the existing models. Further, it was observed that scan based design can be used to extract information from the cryptographic hardware which is undesirable. In this work, a scan based attack on Trivium: a hardware profile in 3rd phase of e-stream is demonstrated. A recent work has been done proposing a hardware solution to avert this scan based attack. This work also presents an attacking scheme against the above work and suggests a modification in the testing hardware which proved to be resilient against the proposed attack.

Chapter 1

Introduction and Motivation

Scan chain based DFT is a powerful and popular test technique. It is embraced by almost every hardware implementation. The best thing about this technique is that it increases the controllability and observability of flip-flops in sequential modules. Thus, it increases the overall testability of the circuits. By controllability we mean, flip-flops can be set to any values that are desired for testing and by observability we mean, the content of flip-flops can be seen very easily. Sending in the test sequences and scanning out the test responses are accompanied by transitions in flip-flops which are one of the sources of major power dissipation in sequential circuits. Moreover, the admirable quality or attribute of scan chain which we talked about above, poses a great threat to the security of crypto-hardware. It becomes imperative on my part to define scan chain and other related terms of SOC before jumping into other issues.

1.1 Scan Chain

Scan chains are a technique used in Design For Testability (DFT) for adding testability features in hardware product design. The objective is to make testing easier by providing a simple way to set and observe every flip-flop in an Integrated Circuit. In full scan design, every flip flop is converted to a scan flip-flop as shown in Figure 1.1. A 2:1 multiplexer is placed before a flip-flop with select line of this multiplexer being the Test_enable signal. When this signal is asserted, every flip-flop in the design is connected into a long shift register, one input pin provides the data to this chain, and one output pin is connected to the output of the chain. Then using the chip's clock signal, an arbitrary pattern can be entered into the chain of flips flops, and/or the state of every flip flop can be read out. Generally, all sequential circuits use scan chains for testing purpose.

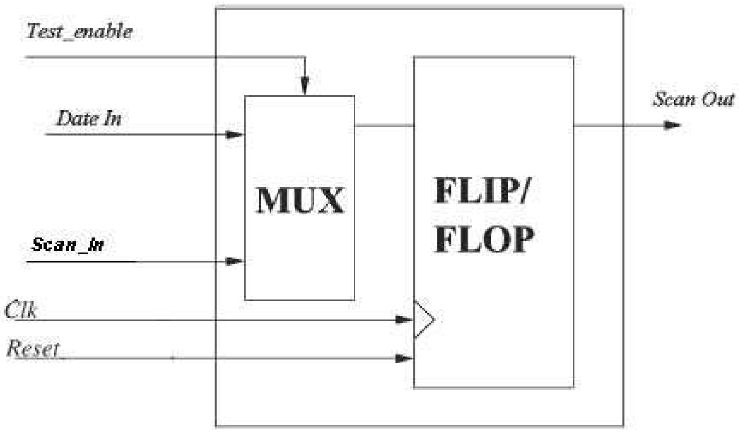


Figure 1.1: Flip flop converted to scan flip-flop

1.2 Definitions Related to SoC

1.2.1 System-on-a-Chip (SOC)

System-on-a-chip refers to integrating all components of a computer or other electronic system into a single integrated circuit (chip). It may contain digital, analog, mixed-signal and often radio-frequency functions - all in one chip. A typical application is in the area of embedded systems. A typical SoC consist of microcontrollers or DSP cores, memory blocks, timing sources, voltage regulators and other external ports.

1.2.2 IP Core

In electronic design a semiconductor intellectual property core, IP block or IP core, is a reusable unit of logic unit, cell, or chip layout design and is also the intellectual property of one party. IP cores may be licensed to another party or can also be owned and used by a single party alone. The term is derived from the licensing of the patent and source code copyright intellectual property rights that subsist in the design. In digital-logic applications, IP cores are typically offered as generic gate netlists.

Soft Cores and Hard Cores

Some vendors offer synthesizable versions of their IP cores. Synthesizable cores are delivered in a hardware description language such as Verilog or VHDL, permitting customer modification (at the functional level). Both netlist and synthesizable cores are called ’soft cores’. Digital IP-cores are sometimes offered in layout format, as well. Such cores, whether analog or digital, are called ’hard cores’, because the core’s application function cannot be meaningfully modified by the customer.

1.2.3 Test Access Mechanisms (TAMs)

Nowadays, large system-on-a-chip (SOC) designs commonly use IP cores that are deeply embedded in the system chip and direct access is often impossible. Individual cores have to be tested on a system level after manufacturing and therefore special test access mechanisms (TAMs) are required. Choosing and scheduling test solutions for SoC embedded IP cores is a very complex problem. In order to facilitate reuse of test vectors provided by the core vendor, an embedded core must be isolated from the surrounding logic, and test access must be provided from the I/O pins of the SOC. Test wrappers form the interface between TAM and core, while TAMs transport test data between SOC pins and wrappers. Refer to the figure 1.2. The arrows inside the SOC show the TAM which is responsible for test data transport. The three major components of test access architectures are:

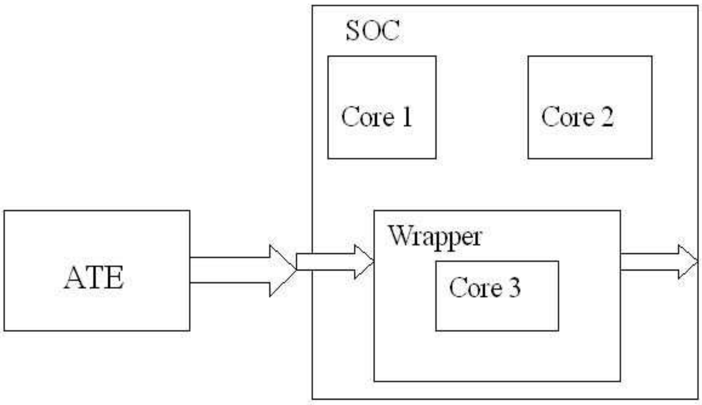


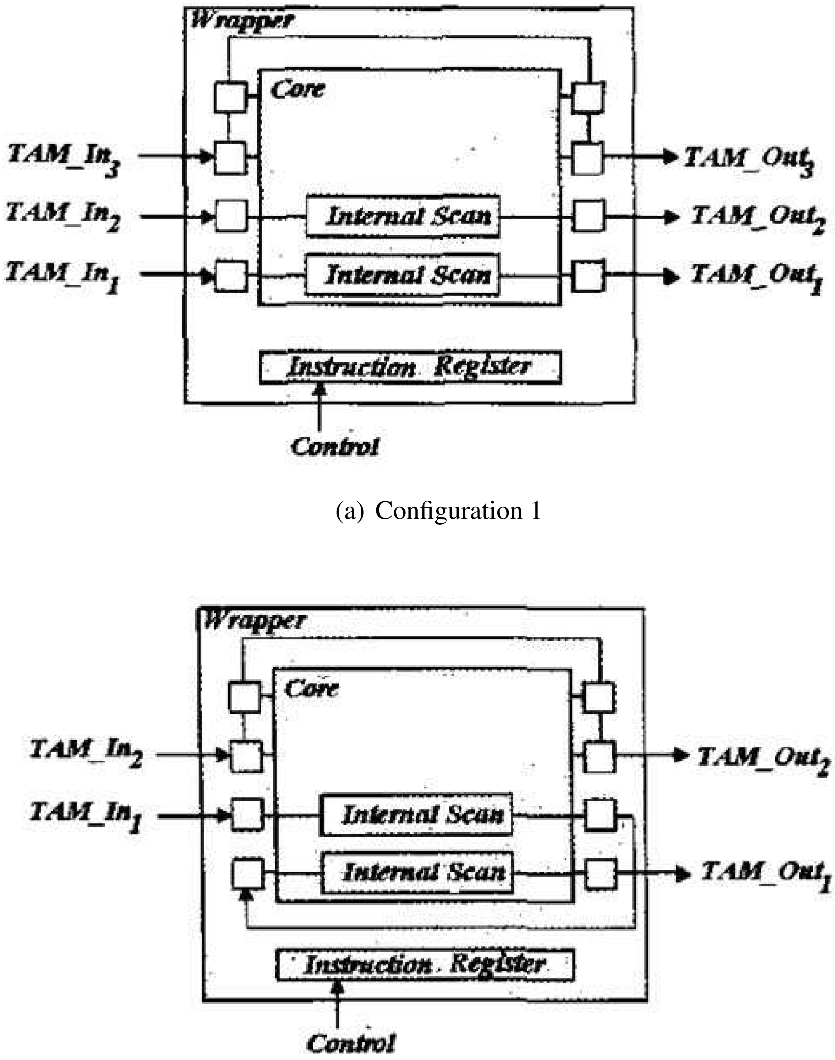
Figure 1.2: SOC showing TAM

1. test source and sink
2. TAMs
3. Test Wrappers.

The general problem of SoC test integration includes the design of Test Architectures, optimization of core wrappers, test scheduling wrapper pin assignments.

1.2.4 Wrapper Width

A set of wrapper configurations for each core are possible. The number of SOC pins needed to access the core through its wrapper under one configuration is called the wrapper width of the core under this configuration. Refer to the Figure 1.3. In 1.3(a), two internal scan chains are separately accessed from SOC boundary through TAM,



(a) Configuration 1

(b) Configuration 2

Figure 1.3: Wrapper Configurations

and all the functional core terminals are accessed serially. In this scenario, 3 wrapper scan chains (6 SOC pins) are needed to access this core. In 1.3(b), the internal scan chains are concatenated and accessed via a single wrapper chain, and all the functional core terminals are accessed serially through another wrapper chain. Thus, 4 SOC pins are needed. Another important thing to be noted is the test application time in the two configurations. In configuration 1, time taken will be less as compared to the time time taken in configuration 2. It is evident that as we try to decrease the number of SOC external pins to be used for testing, testing time increases. So, testing time of an IP core is a function of wrapper width.

1.3 Motivation

For each core of the SOC, there are different wrapper configurations and for each wrapper configuration is associated a testing time. So, each core can be represented as a set of rectangles with one side as the wrapper width and other side being the width dependent testing time. For shortening test time, concurrency in test scheduling is desirable. This concurrency in test scheduling leads to a high power consumption. So, power dimension is added to this rectangle to get a 3D rectangular block. Thus, the entire problem is modeled as a restricted 3D bin packing problem with constraints being power and SOC pins. Existing power approximation models either include a huge fraction of false power or are too complex to implement. First, we tried to address this issue with a hope to get a less complex model with a better opportunity to fit the core specific bins in an effective way. Secondly, on finding that the scan chains whose transitions were being considered to estimate the power consumption of a core are itself a threat to the security of cryptographic hardware, our major focus became the improvement in the hardware so that we could overcome the scan based attack on these hardware.

1.4 Contribution of this work

1.4.1 Power Profile Modeling

Power is dissipated while testing of SOC cores. But, estimation of power consumption of a core while testing is another good problem and it is usually estimated using different power profile models. A Power Profile Model estimates the power dissipated in a core while testing and it often includes false power which are not dissipated actually. Different models differ from each other by the amount of false power considered.

1.4.2 Scan Based Attack

The scan chain can be used to decipher the cryptogram. This is made possible by the inherent capability of scan chain to shift out the internal state of flip-flops in test mode. Scan based attacking schemes are becoming increasingly popular where an attacker gets to know the internal state of the encrypting hardware and then he uses the knowledge of encryption algorithm to decrypt the message or to get the secret key. This type of attack is another sort of side channel attack where in attack is based on the information gained from the physical implementation of the cryptosystem, rather than the theoretical weaknesses in the algorithms. Similarly, in scan based attack, weakness lies in the scan chain architecture.

1.5 Thesis Layout

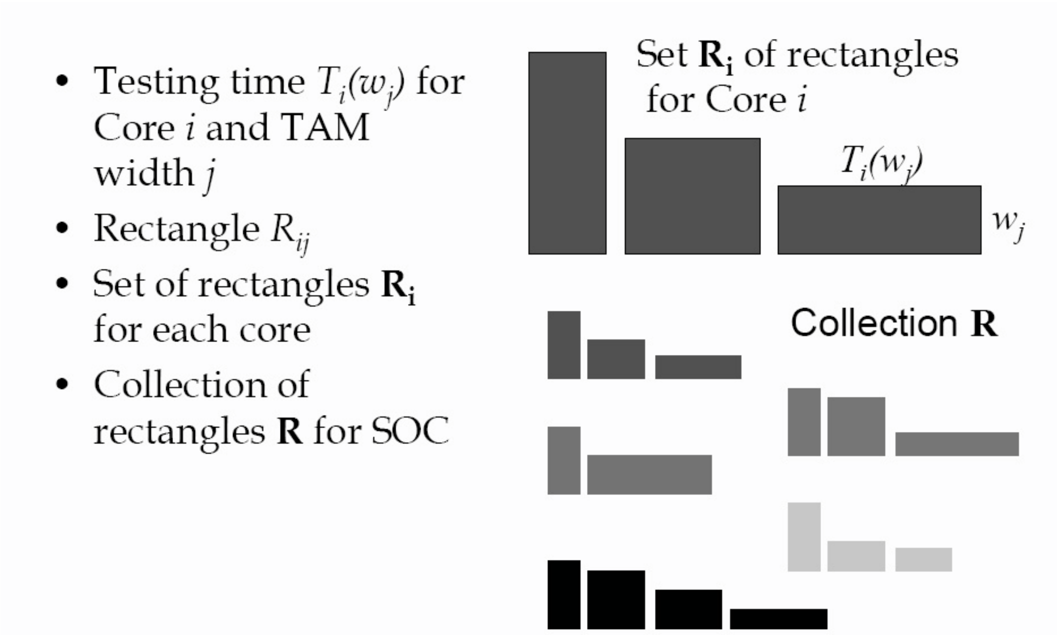
There are 6 chapters in this thesis. In chapter 1, introduction to basic terms and motivation behind this thesis is given. Chapter 2 is Literature survey which superficially reviews the papers referred. Chapter 3 concerns with the development of a new power model whose complexity is not very high. Moreover, it tries to minimize the false power considered. In Chapter 4, scan based attack on Trivium: a stream cipher, is demonstrated. In chapter 5, an attack on flipped scan chain architecture: a solution to overcome the scan based attack, is proposed. Modification in this architecture is also dealt with in the same chapter. In chapter 6, we have tried to outline some of the possible future work in this area.

Chapter 2

Literature Review

2.1 2D-bin packing problem

In [6], SOC test scheduling problem was formulated as a 2-dimensional bin-packing. This work is the first to lay the groundwork to achieve optimum test scheduling. If we consider the testing time of a core, it certainly depends on the wrapper width or the number of SOC pins allocated to this core. Thus, we can say that testing time T is a function of Wrapper width W. Each core can be represented as a set of rectangles as shown in figure 2.1. For each core, there are different wrapper configurations and with



- Testing time $T_i(w_j)$ for Core i and TAM width j
- Rectangle R_{ij}
- Set of rectangles $\mathbf{R_i}$ for each core
- Collection of rectangles $\mathbf{R}$ for SOC

Figure 2.1: Rectangular representation of cores

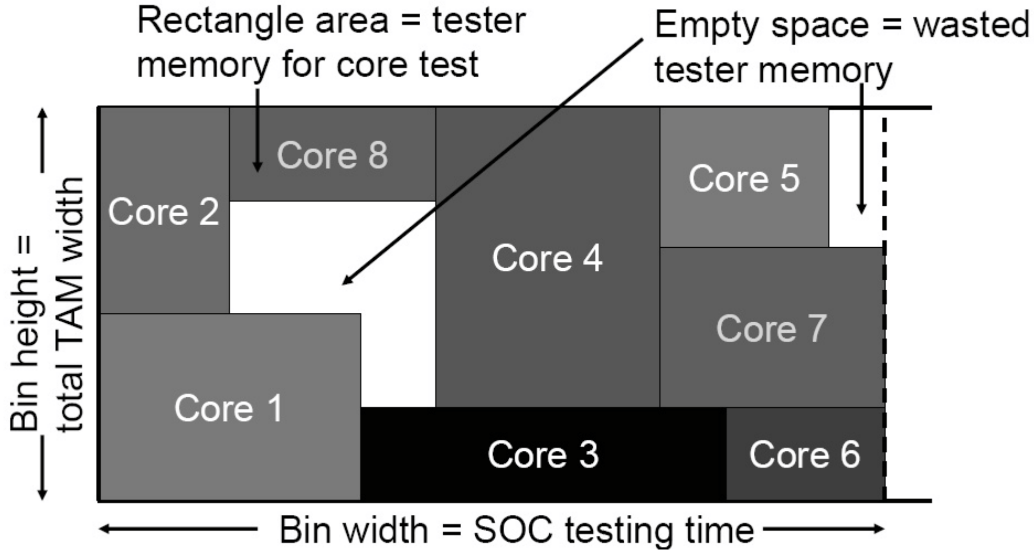

Rectangle area = tester memory for core test
Empty space = wasted tester memory
Bin height = total TAM width
Core 2
Core 8
Core 4
Core 5
Core 7
Core 1
Core 3
Core 6
Bin width = SOC testing time

Figure 2.2: A test schedule under pin constraints

each wrapper configuration is associated a testing time. The 2D bin packing problem states that given a collection R of rectangle sets for the SOC cores, select one rectangle R_{ij} for each core i and pack the selected rectangles into a bin of fixed height such that the bin width is minimized.

In the Figure 2.2, 8 cores have been used and one of the many possible ways of scheduling is shown. Space that remained unfilled is the wasted tester memory that could have been utilized otherwise.

2.2 Restricted 3D-bin packing problem

As already discussed previously, for shortening test time, concurrency in test scheduling is desirable. This concurrency leads to a high power consumption. High power consumption can damage the CUT (circuit under test). It becomes imperative on the part of test engineers to keep this test power under control. For core based SoCs, it is possible to arrange the testing of each module such that test time is minimized while power constraints are not violated. In [7], the researchers added a third dimension to the earlier problem of 2D bin packing i.e. along with TAM width constraints power constraint was added too. So, what we have is a cube in place of a rectangle now (Figure 2.3). Each wrapper configuration of a core is represented by a triple (W, T(W),

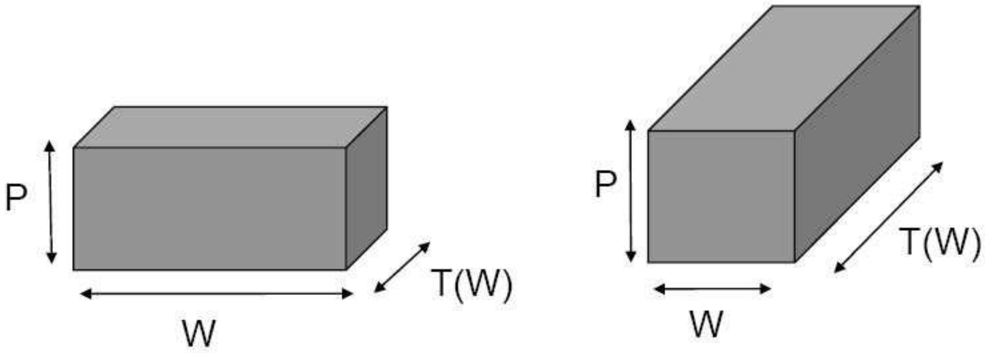


Figure 2.3: Cubical representation of a core

P). W is the number of wrapper chains. T(W) is the core's testing time at W wrapper chains. P is a constant power consumption at any W. However, the problem at hand is a restricted 3D bin packing problem. For example, if two cores are tested concurrently, they overlap in the time dimension and hence cannot have any overlap in the other two dimension since both pins and powers cannot be shared between the two cores that are tested concurrently (Figure 2.4). This is a very important difference from the normal 3-D bin packing problem.

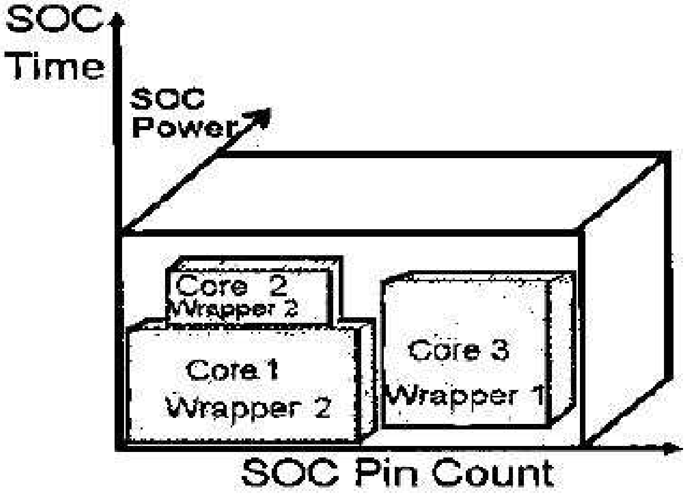


Figure 2.4: A test schedule under pin and power constraints

2.3 Power Approximation Models

Chou et al. [4] approximate the test power consumption for each core to a single fixed value, the peak power consumption. Rosinger et al. [13] referred to this as the global peak power (approximation) model. The false power is the mismatch between the actual power consumption and the modeled power consumption. The single-value power model is rather pessimistic, but it guarantees that the maximum power consumption will not be violated, and is very simple to be handled by a test scheduling algorithm.

Rosinger et al. [13] proposed a double-value test power model (one value representing a constant low power consumption, and the other value representing a constant high power consumption for a test). On top of this double-value test power model, they used test pattern reordering to reshape the power profiles. Further, they also considered test sequence expansion for lowering peak power values, that is, they inserted a new test pattern between two test patterns that generate high peak power values. This new test pattern would not increase the test coverage, but it was merely used for the purpose of lowering the power consumption in different time intervals. Since this resulted in a longer testing time, the test sequence expansion technique is clearly a trade-off between power consumption and testing time.

In [14], the researchers proposed cycle accurate power models which considered power consumed at each clock cycle which resulted in zero false error. But, the disadvantage of this model is that it very complex. In chapter 3, we will be discussing these models in detail.

Transitions in scan chain are considered for computing the power dissipated between application of two successive test patterns. But, it was observed that the structure of scan chain itself makes cryptographic hardware vulnerable to attack. So, we plunged into a new area of scan based attack on crypto-chips.

2.4 Scan Based Attack

There is a raised concern over the security of the cryptographic devices with the increase in its applications and complexity. Scan chains are the most popular testing technique owing to their simplicity, least hardware overhead and appreciable fault coverage. However, this technique poses some security problem as side channel attack is possible on them [17, 8]. In [11, 16], researchers have shown how to decode the cryptogram generated by a stream cipher by exploiting the scan chain. In [17], the authors have shown how to mount a scan based attack on a block cipher(DES). Chapter 4 is devoted entirely in demonstrating such an attack on Trivium: a hardware stream cipher. Intermediate values stored in the flip flops are made available accessing the scan chain, thereby, determining the key. To overcome this threat, researchers have done some good relevant work.

2.5 Related Works against Scan Based Attack

In [5], a scan-chain design based on scrambling was proposed which dynamically reorders the flip-flops in a scan chain. However, statistical analysis of the information scanned out from chips can still determine the scan-chain structure and the secret information [18]. Furthermore, the area overhead and wiring complexity is high. The scrambler uses a control circuit, which requires flip-flops for their implementation. The control circuit uses a separate test key in order to program the interconnections. Thus, if one uses scan chains to test the scrambler circuit, the attack proposed in the study in [17] can be used to decode the test key and, hence, break the scheme of reordering.

In [18], a secure scan-chain architecture with mirror key register having two modes of operation, namely, secure and insecure mode was proposed. A crypto chip can be switched from normal mode to test mode and vice versa when in insecure mode, similar to the normal design for Testability (DFT). However, a chip can only remain in normal mode, when in secure mode. The switching between secure and insecure mode is facilitated by the use of power OFF reset. But the method has the shortcoming that the security is derived from fact that switching off power destroys the data in registers. In addition, at-speed or online testing is not possible with this scheme. Moreover, resetting the device consumes power.

In [9], a lock-and-key technique was proposed, which uses a test security controller (TSC). When a key is successfully entered, the finite-state machine (FSM) of the TSC switches the chip to a secured mode, allowing normal scan-based testing. Otherwise, the device goes to an insecure mode and remains stuck until an additional test-control pin is reset. The design suffers from the problem of large overhead due to the design of the TSC. The TSC itself uses a large number of flipflops (for linear feedback shift registers (LFSRs) and FSMs), which requires built-in self test for testing leading to an inefficient design. Furthermore, the design uses an additional key (known as test key) for security. If the cipher uses an n-bit key for its operation, a brute-force attack would require 2n operations. If the design uses additional t key bits for security, then, with a total of n + t bits of key, the design provides security equivalent to that of min(n, t) bits, which is not desirable.

In [11], the design based on the scan-tree architecture with aliasing-free compactor was proposed. However, the design has the weakness of a large-area overhead due to the design of compactor and its testing circuit.

In [16], a novel architecture was proposed which included the insertion of inverters at certain positions in the scan chain which is unknown to the attacker but, known to the designer and user. Its security is derived from the fact that the attacker does not know the positions of these inverters and thus, he remains unable to analyze the scanned out pattern.

Chapter 3

Hybrid Power Approximation Model

3.1 Chapter Overview

Power is dissipated while testing and under normal operation. Researchers have always been fascinated by the ways to reduce test power. This becomes even more interesting when power becomes a constraint. As a result, we have to make compromises with the test application time. This chapter starts with existing power profile models which is followed by the proposed hybrid power model. This model tries to combine the advantages and disadvantages of existing power models. This is followed by a result section where we will get to see its performance comparison over the performance of other models.

3.2 Existing Models

3.2.1 Global Peak PAM

As shown in the Figure 3.1, the Global Peak Power Approximation Model (GP-PAM) basically flattens the power profile of a core to the worst case instantaneous power dissipation value, i.e., its peak value [4]. According to this model, the power profile of a block is described by the rectangle (width = test sequence length(L), height = global peak value of the power profile P_{hi}). Thus, this model is very simple both in terms of its reliability and complexity which are the basic requirements of a good power approximation model. However, this low complexity of GP-PAM is achieved at a high cost of approximation error, as explicitly shown in Figure 3.1 by the large *false power* region. But, the low complexity cannot be justified on the grounds of high approximation error.

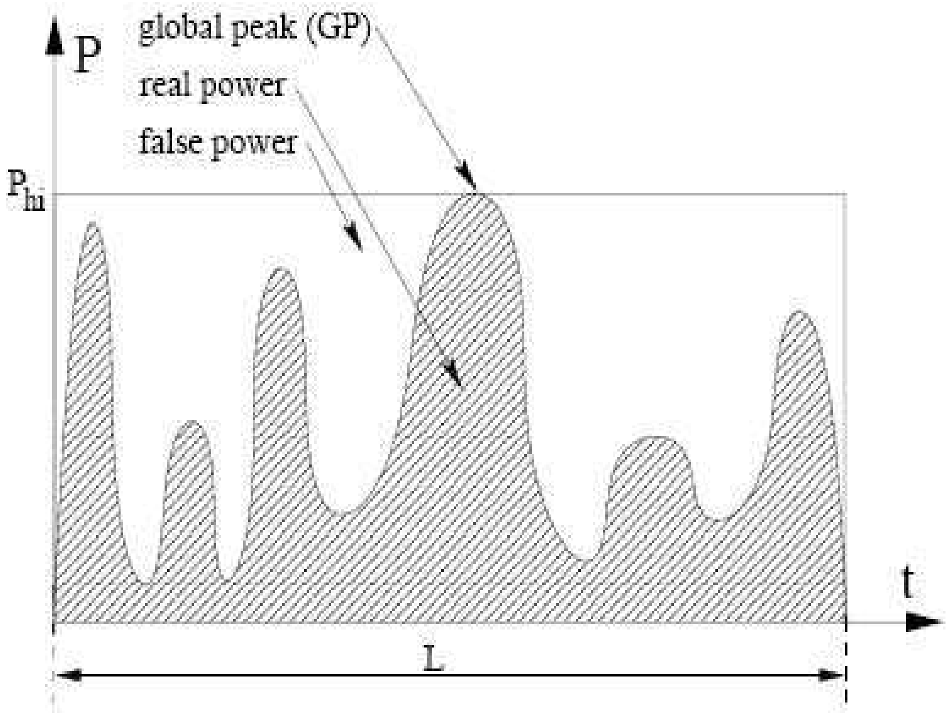


Figure 3.1: Global Peak Power Approximation Model

3.2.2 Cycle Accurate Power Model

In this model, power is computed on a per clock basis and no approximation is done [14]. On how this power is computed is considered in the next section. This is the same method used in the proposed hybrid model, so we thought to deal with it in a separate section of this thesis. The disadvantage of this power model is its huge complexity for storing the power generated in each cycle.

3.3 Computation of Power in sequential blocks

The power consumption is the sum of a static part and a dynamic part. For most current CMOS technologies, the static part is constant and dominated by the dynamic part. Usually, the dynamic part is proportional to the switching activity α (the number of zero-to-one and one-to-zero transitions) in the circuit. Hence, the researchers concentrate on how this α is computed on the basis of the given test stimuli and the given expected test responses for a core. A sequential block has a combinational logical part and other part comprising of flip-flops. For testing purpose, these flip-flops are modified into scan flip-flops which in turn, form scan chains. Sankaralingam et al. [15] empirically showed that the number of transitions in the logic of a core when applying a test, α_{logic}, is approximately linear to the transitions in the cores flip-flops, α_{ff}. A

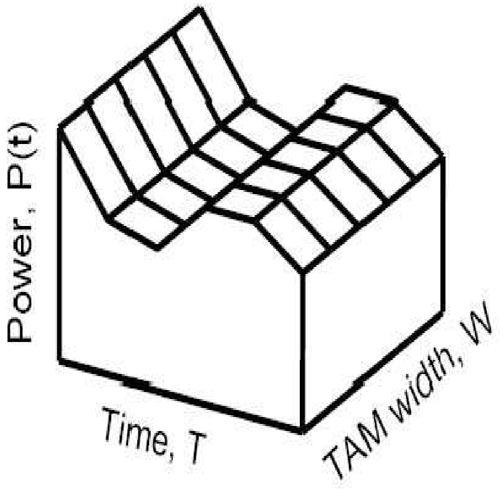


Figure 3.2: A core with Cycle Accurate Power Model

representative plot for one of the benchmark circuits namely s9234 is given in the figure 3.3. Hence, α = α_{ff}+ α_{logic} = α_{ff}+kα_{ff}+l, where k and l are constants. So, only α_{ff} is needed to be computed.

Computing α_{ff}

Power is consumed whenever a flip flop undergoes transition 0→1 or 1→0. When some pattern is fed into a scan chain, a pattern already stored in the scan chain is scanned out simultaneously. Total transition count while scanning in or scanning out is called *Weighted Transition Count* or WTC. Since they occur simultaneously, a general term *Weight* is used for referring total transition count in scanning out a pattern and scanning in another pattern concurrently. Suppose, we want to compute the power dissipated between the shifting out the test response V_j and shifting in the input vector V_i in a scan chain of length m. The WTC values corresponding to V_i are:

$$WTC_{scanin}(V_i) = \sum_{j=1}^{m-1} (V_i(j) \oplus V_i(j+1))(m-j))$$

$$WTC_{scanout}(V_i) = \sum_{j=1}^{m-1} (V_i(j) \oplus V_i(j+1))j)$$

In the above two equations, $V_i(j)$ represents j^{th} bit of the vector V_i. Now, power dissipated in scanning out V_i and scanning in V_j can be written as:

$$Weight(V_i, V_j) = WTC_{scanout}(V(i)) + WTC_{scanin}(V(j))$$

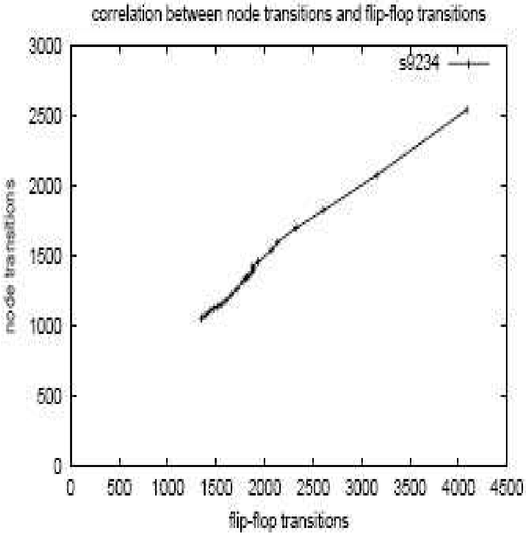


Figure 3.3: Correlation between Node Transition Count and flip-flop transition count

If we have a test set **V** = $\{V_1, V_2, ...V_n\}$ and the corresponding test response set **R** = $\{R_1, R_2, ..., R_n\}$, and it is fed in the same sequence as in the test set above, α_{ff} can be computed as:

$$\alpha_{ff} = WTC_{scanin}(V_1) + \sum_{i=1}^{n-1} Weight(R_i, V_{i+1}) + WTC_{scanout}(R_n)$$

3.4 Hybrid Model

This model is proposed to reduce the complexity of cycle accurate power model and the approximation error of GP-PAM. The construction of this model is divided in three basic steps which are outlined below:

STEP 1: The entire power profile is split into a number of block. The length of each block is fixed depending on the number of test patterns. For example, say dividing the power profile into blocks of 50 or 100 test patterns each.

STEP 2: Within each block so formed, test vectors are rearranged in such a way that the initial sequence of test vectors leads to a comparatively low power consumption followed by a sequence of vectors leading to a relatively high power consumption. The test vectors are reordered using a Greedy approach. A complete graph is formed with

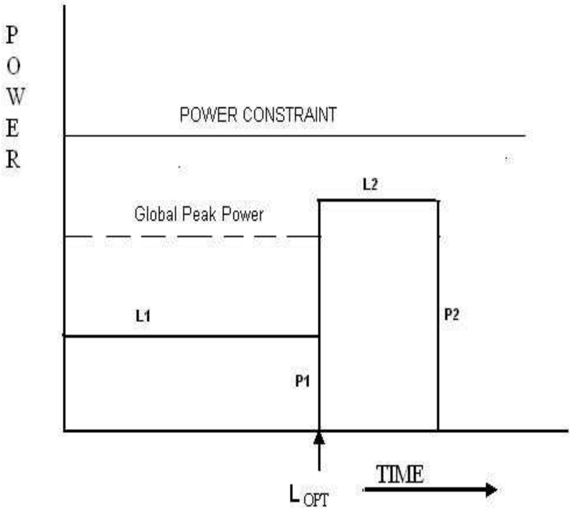


Figure 3.4: Hybrid Model

vertices as test patterns. A vertex i represents test pattern i. An edge from vertex i to vertex j means pattern V_i is scanned out and pattern V_j is scanned in. The weight assigned to an edge between vertices i and j is Weight(V_i, V_j). As a result, a bidirectional clique is formed. Now, the problem at hand is reduced to finding a Hamiltonian tour of low cost. Any vertex is selected as the starting node and then next best node is sought. The next best node is defined as a node which minimizes the power dissipation in terms of node transition count from the current node being considered.

STEP 3: A value $t = L_{OPT}$ is to be found out which minimizes the quantity $P_1 \times L_1 + P_2 \times L_2$ where L_1 is the width of the region on the horizontal axis before $t = L_{OPT}$ and L_2 is the region following it as shown in Figure 3.4. P_1 is the maximum instantaneous power in the region L_1 and P_2 is the sam in region L_2. Obviously, $P2 > P_1$ because test vectors are arranged in such a manner that instantaneous power at each step is minimum among the rest of the vectors.

3.5 Implementation and Results

Implementation of hybrid model requires the knowledge of cores. Test vectors are required for generating the power profile and manipulating them. But, the problem with this is that ITC'02 benchmark SOC circuits does not provide the knowledge of test vectors or the internal circuits design. Another alternative is using already tested ISCAS'89 circuits whose internal design is known. For generating the power profile test vectors were generated using the Synopsys tetramax tool. Test vectors are then parsed from the pattern file using the Java ANTLR tool. Table 3.1 shows the improvement

Circuit	Proposed	GP-PAM	Difference	improvement(%)
s344	3204	3861	657	17.02
s420	9361	13932	4571	32.81
s444	11140	13676	2537	18.55
s1423	198686	225638	26952	11.95
s5378	4360449	4700752	340303	7.24
s9234	9369731	9995040	625309	6.26
s13207	108851211	112105128	3253917	2.91
s15850	65095828	70414911	5319083	7.56
s35932	101927819	105108126	3180307	3.03
s38584	758351027	806073609	47722582	5.93

Table 3.1: Comparison of Hybrid Model with GP-PAM.

Block Size	Proposed	Difference	Improvement(%)
30	65095828	5319083	7.56
40	65353153	5061758	7.12
50	65349776	5065135	7.20
60	65602782	4812129	6.84
70	65606352	4808559	6.83
80	65866707	4548204	6.46

Table 3.2: Effect of block size on percentage improvement

in power approximation error of the proposed hybrid power approximation model over the GP-PAM.

Important thing to observe is the effect of block-size (i.e. the number of test vectors each block should have) on the percentage improvement over GP-PAM. It is noted that by increasing the block size, the overall improvement gradually reduces. When considered the work in [13], our solution reduces to their solution when block size is considered to be the entire length of the test set. So, our proposed method is working better in performance consideration. But then, we cannot go on decreasing the number of vectors per block for gaining improvement. As such, the complexity will increase tremendously. So, this value should be chosen in such a way that complexity as well as false power is reduced. Table 3.2 shows the same effect on s15850. The improvement gradually decreases with the increase in number of test patterns per block.

Chapter 4

Scan Based Attack on Trivium

4.1 Chapter Overview

As stated earlier, scan based attack on encryption hardware is possible by getting the knowledge of internal structure of scan chain and then decrypting the cryptogram. Such attacks have been reported in different works as mentioned in chapter 3. This chapter is dedicated to such an attack on Trivium. The following section introduces Trivium and then we move on to the proposed attacking mechanism.

4.2 Trivium Specifications

Trivium is one of the hardware candidates in phase 3 of *estream*[1]. Trivium is a synchronous stream cipher designed to provide a flexible trade-off between speed and gate count in hardware, and reasonably efficient software implementation [3]. It generates up to 2^{64} bits of output from an 80-bit key and an 80-bit *Initialization Vector* (IV). As for most stream ciphers, this process consists of two phases: First the internal state of the cipher is initialized using the key and the IV, then the state is repeatedly updated and used to generate key stream bits. Here, second phase of key stream generation is considered first in order to save some space.

4.2.1 Key Stream Generation

Trivium's 288-bit internal state consists of three shift registers of different lengths as is shown in Figure. 4.1. The key stream generation consists of an iterative process which extracts the values of 15 specific state bits and uses them both to update 3 bits of the state and to compute 1 bit of key stream z_i [3]. The state bits are then rotated

[1] `http://www.ecrypt.eu.org/stream/triviump3.html`

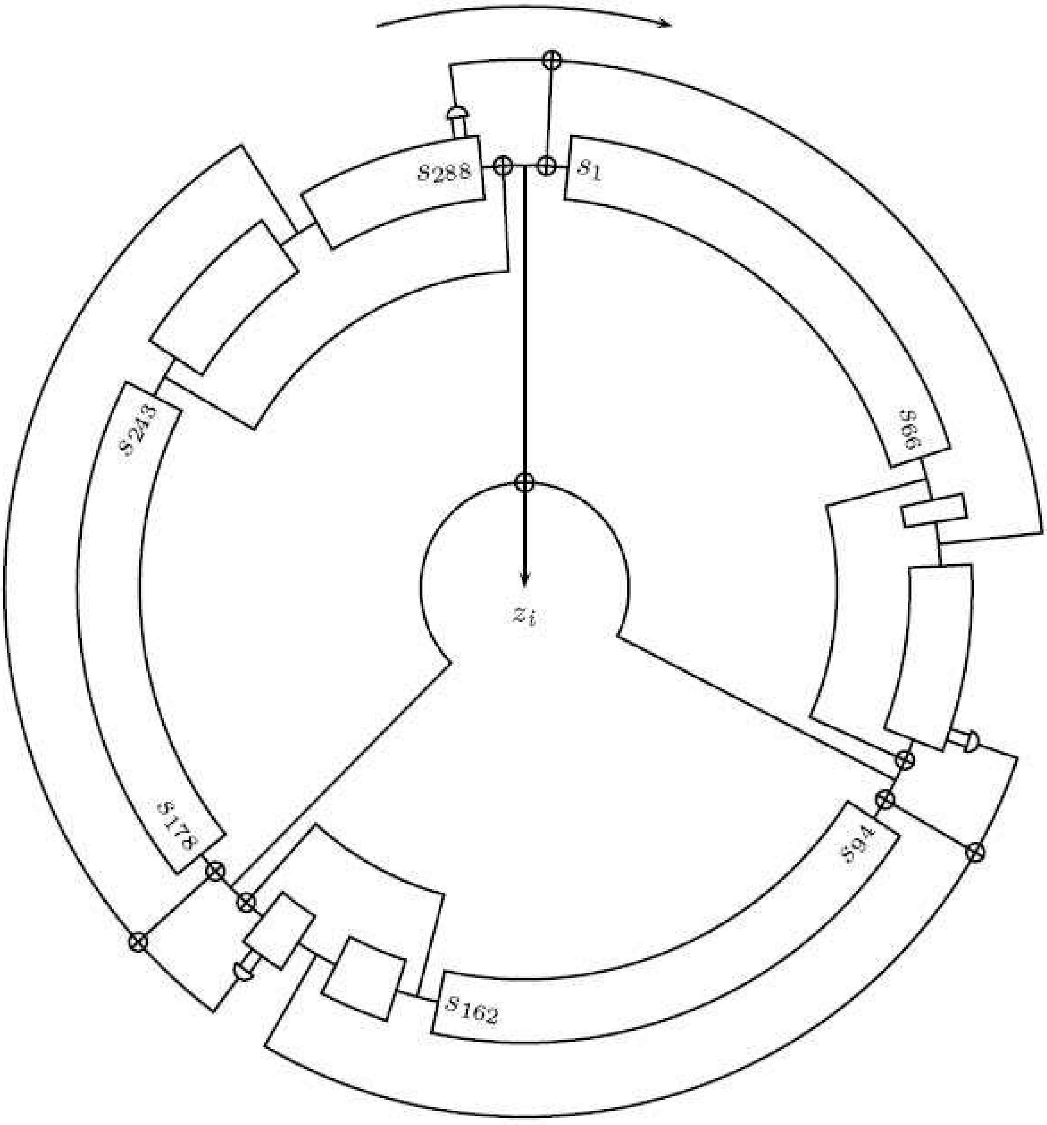


Figure 4.1: Trivium

and the process repeats itself until the requested N $\leq$ 264 bits of key stream have been generated. This can be summarized as:

```
for $i$ = 1 to N do
    $t_1 = s_{66} + s_{93}$
    $t_2 = s_{162} + s_{177}$
    $t_3 = s_{243} + s_{288}$
    $z_i = t_1 + t_2 + t_3$
    $t_1 = t_1 + s_{91}.s_{92} + s_{171}$
    $t_2 = t_2 + s_{175}.s_{176} + s_{264}$
    $t_3 = t_3 + s_{286}.s_{287} + s_{69}$
    $(s_1, s_2, ..., s_{93}) \leftarrow (t_3, s_1, ..., s_{92})$
    $(s_{94}, s_{95}, ..., s_{177}) \leftarrow (t_1, s_{94}, ..., s_{176})$
    $(s_{178}, s_{179}, ..., s_{288}) \leftarrow (t_2, s_{178}, ..., s_{287})$
end for
```

Note: ‘+’ and ‘.’ symbols are used for showing XOR and AND operation respectively. This completes the key stream generation phase of the encryption algorithm. The initialization is dealt with in the next subsection.

4.2.2 Key and IV Setup

To initialize the cipher, the key and IV are written into two of the shift registers, with the remaining bits starting in a fixed pattern. The cipher state is then updated $4 \times 288 = 1152$ times, using the same algorithm as above but without producing the key stream bit z_i. This can be summarized as:

```
$(s_1, s_2, ..., s_{93}) \leftarrow (K_1, K_2, ..., K_{80}, 0, .., 0)$
$(s_{94}, s_{95}, ..., s_{177}) \leftarrow (IV_1, IV_2, ..., IV_{80}, 0, 0, 0, 0)$
$(s_{178}, s_{179}, ..., s_{288}) \leftarrow (0, ..., 0, 1, 1, 1)$

for $i$ = 1 to 4.288 do
    /*Same algorithm as in phase 1 but without
    generating the key stream bit $z_i$*/
end for
```

4.3 Attack on Trivium

The objective of the attacker is to obtain the message from the stream of ciphertexts[11]. He observes the cryptograms $c_1, c_2, ..., c_l$. He then gets possession of the device and his intention is to obtain the plaintext $m_1, m_2, ..., m_l$ using scan based side channel attack. As illustrated in [11, 16], this attack works in two phases. The first phase aims at ascertaining the position of different registers in the scan chain. In the second phase, this information is used to obtain the sequence of plaintext. In the context of Trivium, the first phase involves determining the positions of s-bits namely $s_1, ..., s_{288}$ in the scan chain. This phase of the attack is is described below in the next section.

4.3.1 Ascertaining the location of s-bits

First, when the attacker takes control of the chip, he scans out the pattern which the internal state of the trivium registers have. But he does not know the exact positions of these s-bits which he decides in the following steps.

1. The attacker sets the K and IV input lines to be zero. Therefor value of K_i and IV_i is zero. One clock is then given in normal mode which sets s_1 to s_{285} to 0 and s_{286} to s_{288} to 1. This is easily followed from the encryption algorithm given in the previous section. The pattern is scanned out in test mode to know the positions of s_{286} to s_{288} collectively (in a group). To know the exact position of s_{286}, the same procedure is repeated and one extra clock is given in the normal mode. This makes s_{286} zero (due to right shifting operation in the trivium registers) and thus, when the pattern is scanned out, position of s_{286} is known. Similarly, the respective positions of s_{287} and s_{288} are also known by giving one more extra clock in normal mode. This can be easily seen because when we give three clocks after setting IV and K lines to be zero, other location such as s_1 becomes 1, but since we are only interested in three specific locations whose collective positions are already known, other such positions can be easily ignored. Total time taken in this procedure is $O(3 \times 288)$ clocks.

2. Set $K_1 = 1$ and all other input lines to be zero. One clock in normal mode lead us to the position of s_1. This is known by the scanning out the pattern. Similarly, s_2 to s_{80} can be known by setting the corresponding K_i line to be one and applying one clock. Also, s_{94} to s_{173} is located by setting IV lines to 1 individually one-by-one and repeating the same procedure as stated above. Time complexity of this step is $O(160 \times 288)$ clocks.

3. s_{81} to s_{93} can be located by setting $K_{80} = 1$ and all other input lines to be 0 and applying repeated clocks and scanning out in an iterative manner. Locating s_{81} will require 2 clocks , s_{82} will require 3 clocks and so on. These are the number of clocks other than the clocks required for scanning out the entire pattern. Similarly, by setting IV_{80} to be 1 and all other input lines to be 0 positions of s_{174} to s_{177} can be known in an iterative fashion. Total time taken in this step is again $O(c \times 288)$ clocks where c is a small constant.

4. Since we know the position of s_{177} we can set this bit to 1 either by scanning in the required pattern in test mode or by setting $IV_{80} = 1$ and giving 5 clock cycles in normal mode. After doing this, one more clock is given in normal mode which sets $s_{178} = 1$. Its position is known by scanning out the pattern. Remaining s-bits namely $s_{179}\ to\ s_{285}$ can be located similarly in an iterative procedure.

This completes the first phase of the attack. It should be noted that the total number of clocks required for knowing the position of each bit is O(n) where n is the total number of flip-flops. Overall the time taken is $O(n^2)$. Now we show how can we decipher the cryptogram.

Present State	**Previous State**
$(s_1, s_2, ..., s_{93})$	$(s_2, s_3, ..., s_{93}, a)$
$(s_{94}, s_{95}, ..., s_{177})$	$(s_{95}, s_{96}, ..., s_{177}, b)$
$(s_{178}, s_{179}, ..., s_{288})$	$(s_{179}, s_{180}, ..., s_{288}, c)$

Table 4.1: Internal states of Trivium

4.3.2 Deciphering the cryptogram

In a stream cipher, we XOR the plain text bit to the key stream bit to get the cipher text bit. So, if we have key stream bit K_i and cipher text bit C_i, plain text bit P_i can be obtained as

$$P_i \; = K_i \; \oplus \; C_i$$

So our basic motive is to get all the key stream bits. The attacker had scanned out the internal state of Trivium after getting hold of the device. Now, he has ascertained the positions of s-bits in the scan chain. This information will be used to decipher the cryptogram and to obtain the plain text. We proceed by knowing the previous state from the current state. Refer to the table 4.1. As is clear from the encryption algorithm, current state is a right shift of previous state with first bit being the non-linear function of some other bits. So, our task remains to calculate ‘a’, ‘b’ and ‘c’. Observe the following equations:

$$t_1 = s_{66} + s_{93} \tag{4.1}$$

$$t_1 = t_1 + s_{91}.s_{92} + s_{171} \tag{4.2}$$

$$(s_{94}, s_{95}, ..., s_{177}) \leftarrow (t_1, s_{94}, ..., s_{176}) \tag{4.3}$$

Equations 5.1 and 5.2 can be combined to get

$$t_1 = s_{66} + s_{93} + s_{91}.s_{92} + s_{171} \tag{4.4}$$

This should be noted that if we give a clock at this configuration of trivium registers s_{94} gets loaded with t_1 and other bits are shifted to their right. So, we can say that what is s_{67} now, must be s_{66} in the previous state and what is s_{93} now, must be s_{92} in the previous state and so on. Hence, from equations 5.3 and 5.4 and by referring to the table 4.1 we have the following equation:

$$s_{94} = s_{67} + s_{92}.s_{93} + a + s_{172}$$

$$\Rightarrow a = s_{94} + s_{67} + s_{92}.s_{93} + s_{172}$$

Similarly, ‘b’ and ‘c’ can be deduced by the following set of equations:

$$s_{178} = b + s_{163} + s_{176}.s_{177} + s_{265}$$

$$\Rightarrow b = s_{178} + s_{163} + s_{176}.s_{177} + s_{265}$$

And,

$$s_1 = c + s_{244} + s_{287}.s_{288} + s_{70}$$

$$\Rightarrow c = s_1 + s_{244} + s_{287}.s_{288} + s_{70}$$

In this way, we can compute the previous state from a given current state. That means, we can compute all the previous states given a single current state of the internal registers. Once obtained a state, it is loaded on to hardware by scanning in the required pattern. Applying one clock in normal mode then gives us the key stream bit which when xored with ciphertext bit of that state produces corresponding plaintext bit.

Chapter 5

Attack on Flipped Scan chain Architecture

5.1 Chapter Overview

This chapter deals with the attack on Flipped Scan Chain (FSC) Architecture [16]. Details of FSC is given in the next section with three different subsections on its architecture details, working mechanism and security analysis. That section is followed by another section dealing with an attack on this FSC Architecture which we have proposed. Furthermore, a solution is proposed by modifying the FSC architecture and its performance is evaluated against other existing architectures.

5.2 Flipped Scan Chain (FSC) Architecture

5.2.1 Architecture Details

In [16], inverters or NOT gates were introduced (Fig. 5.1) at the input of the scan in pin of the scan D-flip flop (SDFF) and these were termed as flipped scan DFF or FSDFF. In the architecture, scan chain consisted both of SDFFs and FSDFFs. The presence of inverters aimed at preventing the scan data from being analyzed in order to determine the intermediate values stored in the flip flops. Moreover, this design does not pose any hindrance to the normal functionality of the device. As illustrated from the figure 5.2, value of each a_i tells the presence or absence of inverters at the i^{th} position of the chain. If a_i = 1, inverter is present, otherwise absent.

5.2.2 Working Mechanism

With the above mentioned structure of scan chain, one who doesn't know the location of these inverters will not be able to know the internal value of flip-flops even he performs

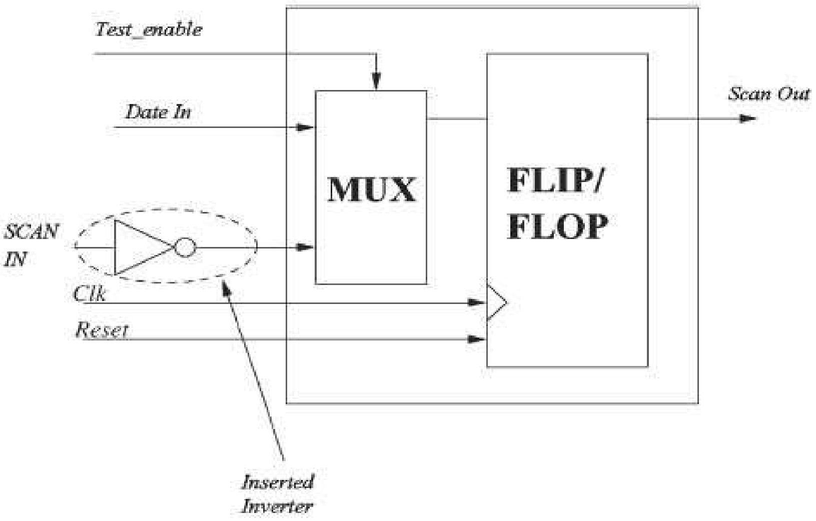


Figure 5.1: Flipped Scan DFF (FSDFF)

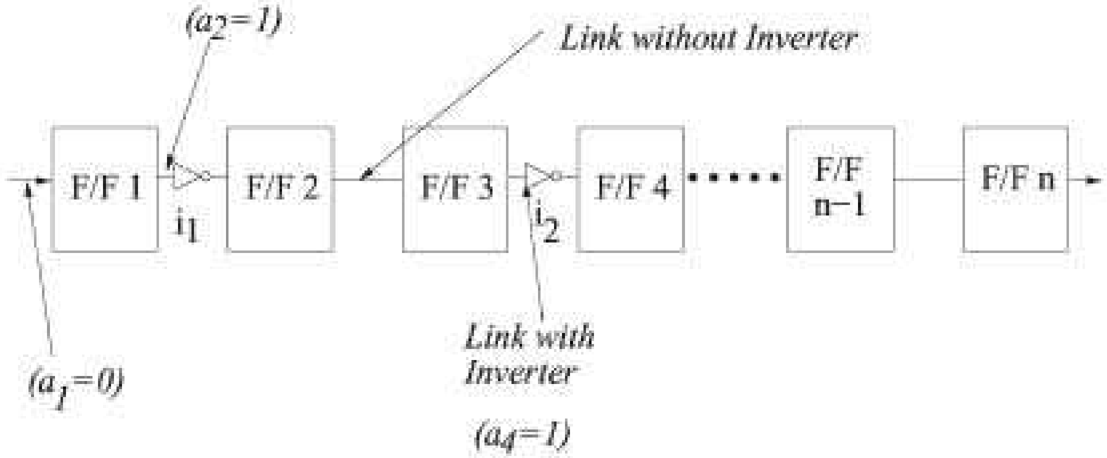


Figure 5.2: FSC Architecture

a scan out operation in test mode. This is a explained as follows:
Suppose a sequence $X = \{X_1, ..., X_n\}$ is fed in the scan chain, then the actual sequence reaching to the corresponding flip flops would be $\overline{X} = \{\overline{X}_1, ..., \overline{X}_n\}$ where

$$\overline{X}_1 = X_1 \oplus a_1$$

$$\overline{X}_2 = X_2 \oplus a_1 \oplus a_2$$

$$\vdots$$

$$\overline{X}_i = X_i \oplus a_1 \oplus a_2 \oplus ... \oplus a_i$$

And, the scanned out sequence would be $X' = \{X'_1, ..., X'_n\}$ where

$$X'_i = X_i \oplus a_1 \oplus a_2 \oplus ... \oplus a_{n+1}$$

Moreover, if at certain point of time, the internal state of flips-flops is $I = \{I_1, I_2...., I_n\}$ and somebody scans out the pattern in test mode, the scan output would be a pattern $I' = \{I'_1, I_2..., I'_n\}$ where

$$I'_1 = I_1 \oplus a_2 \oplus a_3 \oplus \ldots \oplus a_{n+1}$$

$$I'_2 = I_2 \oplus a_3 \oplus a_4 \oplus \ldots \oplus a_{n+1}$$

$$\vdots$$

$$I'_n = I_n \oplus a_{n+1}$$

Suppose the attacker sends in a pattern through the scan-input pin and observes the pattern coming out of the scan-output pin when the design is in test mode. From the polarity of the input and output pattern, the attacker is able to know whether an even or odd number of inverters are present in the scan chain. Hence, he knows the value of $a_1 \oplus a_2 \oplus \ldots \oplus a_{n+1}$, as for n flip-flops, there are (n + 1) links to place the inverters. However, using the scan chains, the attacker is unable to ascertain the number of inverters between the scan-input pin and a flip flop, between any two flip-flops, or between a flip-flop and the scan-output pin. Hence, the attacker cannot exploit the scan chain to ascertain the values of $a_1, a_1 \oplus a_2, ..., a_1 \oplus a_2 \oplus \ldots \oplus a_{n+1}$. Thus, as may be observed from the above relations, the attacker cannot obtain the values of the pattern stored in flip=flops any better than guessing the values of n+1 binary variables $a_i (1 \leq i \leq n+1)$.

5.2.3 Security Analysis

We have seen how the FSC architecture works. Its security lies in the fact that the attacker does not know the specific locations in the scan chain where inverters are placed. So, if an attacker somehow gets to know these locations, he can easily analyze the scanned out data and thus cryptographic hardware remains no more secure. Its security analysis was based on probability theory[16]. It aimed at showing that the task of determining the position of inverters is infeasible. Suppose, designer has placed inverters at k location out of n+1 possible locations. Then probability of guessing these positions is $1/\ {}^{n+1}C_k$. Further, it was shown that this approximates to $1/2^n$ for large values of n, if $k = \lfloor (n+1)/2 \rfloor$.

In the next section, an attack on FSC is shown.

5.3 Attack on FSC Architecture

A very simple attack is designed to know the positions of this architecture. The attack works in the following way:

There is a RESET signal going to each SDFF which resets each of them to zero. So what an attacker can do is use this reset signal and obtain the scan out pattern by operating the crypto chip in test mode. The scan out pattern would look like pattern of zeroes interleaved with patterns of ones i.e. patterns of zeroes and ones would appear alternatively. The places where polarity is reversed are the locations where an inverter has been inserted. This will become more clear with the following example.

Example:

Suppose there is a chain of 10 SDFFs and inverters are placed at position number 1, 3, 6, 7, and 11 out of 11 valid positions as is shown in Figure. 5.3. We apply a reset signal and shift out the pattern. We will get a pattern $X = \{X_1, X_2, ..., X_{10}\}$ where $X_i = a_{i+1} \oplus a_{i+2} \oplus ... \oplus a_{11}$. We will get this pattern in the order $X_{10}, X_9, ..., X_1$.In this example,

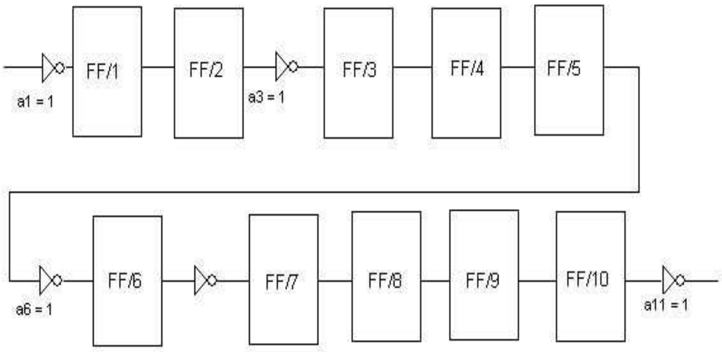


Figure 5.3: Example demonstrating attack on FSC

$$X_{10} = a_{11}$$

$$X_9 = a_{10} \oplus a_{11}$$

$$X_8 = a_9 \oplus a_{10} \oplus a_{11}$$

$$\vdots$$

$$X_1 = a_2 \oplus a_3 \oplus \ldots \oplus a_{11}$$

The above set of equations is easily followed from the equations in the subsection on Working Mechanism of FSC Architecture and from the fact that initial state of the flip-flops have been all reset to zero before scanning out. So, we will get a sequence $X = \{0,0,1,1,1,0,1,1,1,1\}$. As previously stated, inverters have been inserted at those positions where polarity is reversed. Here, polarity is reversed at positions 3, 6 and 7. Hence, the positions where inverters were placed are 3, 6 and 7 and 11. Presence of an inverter at the last position i.e. at the 11^{th} position was detected using the fact that first bit obtained while scanning out is 1. Again, to know whether there is an inverter in the first position or not, any vector is applied. Since, we know the total number of inverters placed at other positions, by observing the polarity of the scan out pattern, this problem can be easily solved. Thus, inverters are placed at position 1,3,6,7 and 11.

By the above procedure, one can always ascertain the location of inverters. Therefore, the design in [16] is no more secure.

5.4 ScanSeal Architecture: A Solution

In the attack shown above, Reset signal is creating trouble and making the entire architecture vulnerable to external attack. So, had there been a design which could have subdued the effect of Reset signal, then our problem would have been solved. This section concentrates on such a design which effectively replaces each inverter with a digital circuit. This circuit is described in details in the next subsection.

5.4.1 Architecture Details

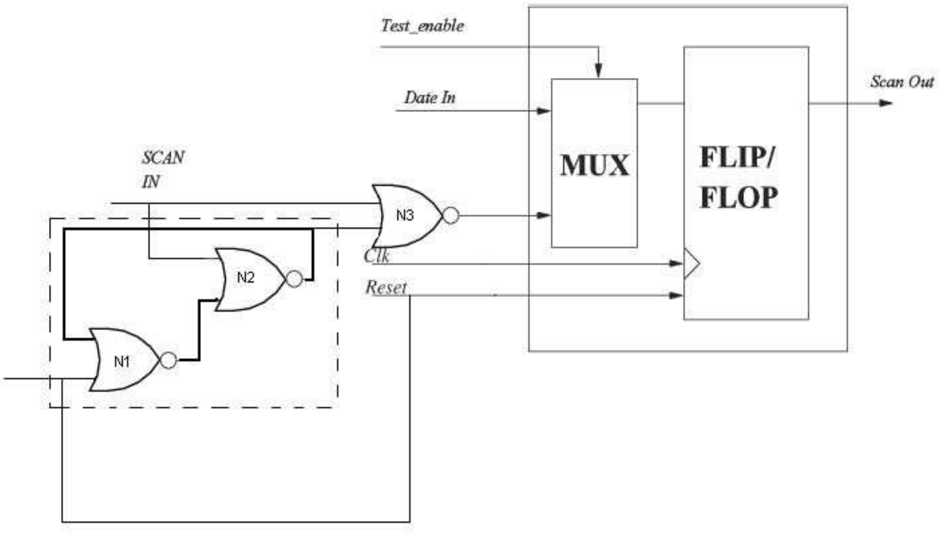


Figure 5.4: ScanSeal: Inverters replaced with 3 NOR gates

In this design, replacement of each inverter in the FSC by a digital circuit is proposed. Other than overcoming the effect of Reset signal, this design aims at restoring the same sense of security that the inverters had incorporated. We will call this digital circuit *ScanSeal* from here onwards. It consists of three NOR gates interconnected among each other as shown in the Figure 5.4. The two NOR gates, namely N1 and N2, within the broken rectangle can be thought as a SR latch as shown in the Figure 5.5. The Q output of this SR latch is fed into a 2 input NOR gate namely N3. The other input of this NOR gate is driven by the scan-in line. This scan-in line also goes to the R input of the SR latch. The reset signal which resets every flip flop in the scan chain is also wired to the S input of the SR latch. The working mechanism of ScanSeal is addressed in the next subsection.

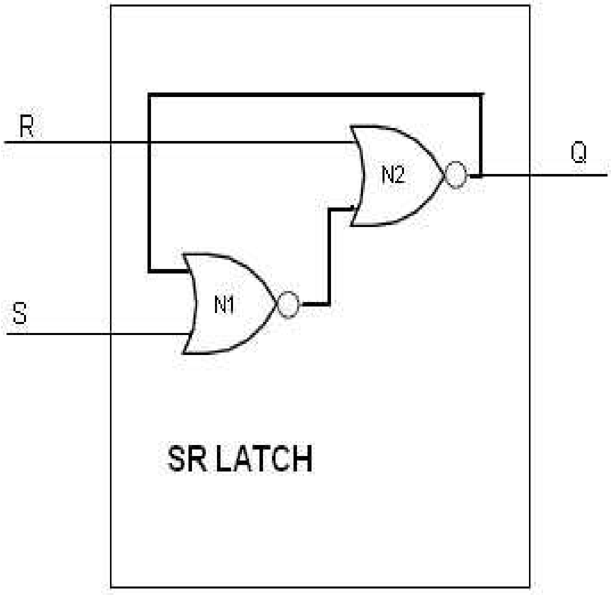


Figure 5.5: SR latch

5.4.2 Working mechanism

The principle of its operation depends on the state of SR latch. Table 5.1 shows different states in which Q jumps to, when S or R is changed. The S signal sets the SR latch i.e. Q goes to 1 when S is given a high value and R signal resets the latch i.e. Q jumps to 0 when R is given a high value. When both inputs are zero, nothing happens and Q continues to be in the same state in which it had been before. It is to be noted that both S and R cannot hold high values simultaneously.

We call a NOR gate to be deactivated if one of its input value is 1. This is so because, the output value of NOR gate then goes to low and remains in that sate unless

S	R	Action
0	0	Keep state
0	1	0
1	0	1
1	1	Invalid

Table 5.1: SR latch operation

both of the input lines go to zero. In the context of the proposed design of ScanSeal, N3 is said to be deactivated when its input driven by the output of N2 goes high (This is same as Q output of SR latch which we were talking about). And, we will call it activated otherwise. When Reset signal is applied, since it is wired to S input of SR latch, N3 gets deactivated. Thus, a value of zero goes to the scan flip-flop irrespective of the value of the scan-in line. It continues to be in this way unless N3 gets activated. For its activation Q must go to a high state. This in turn requires, scan-in line going to R input of SR latch to go high. Once activated, N3 starts behaving like a simple NOT gate. It is to be noted that forbidden state of SR latch is not reached i.e. both S and R inputs cannot become 1 simultaneously. Because, as soon as the Reset signal is applied, all flip-flops return to zero state. With this all scan in line which is scan out of previous flip flop in the scan chain becomes 0. Hence, R line becomes zero.

It should have become clear by now that the kind of attack discussed in the section 5.2 won't work here because on getting a reset signal every flip-flop goes to zero and scanning out would give me all zeroes. What ScanSeal does is basically it conceals the data in scan-in line and prevents it from passing down the chain. With this design, the attacker cannot do anything but guessing to determine the positions of these ScanSeal circuits.

5.5 A Problem with ScanSeal

5.5.1 Issue

There is a major issue in the above design of ScanSeal. Here, a problem arises when the Reset signal has been applied and the user wants to use the device in the test mode thereafter. There is an associated difficulty because all the NOR gates of type N3 (Refer to Figure 5.4) are deactivated and there is no mechanism to activate them so that the user could use it for analyzing the scan out data. In this scenario, a pattern is introduced termed as *activation sequence* which is send through the scan chain.

5.5.2 Activation Sequence

Activation sequence is named so after the activity it performs when it is send in the scan chain with all deactivated NOR gates. It goes on activating all the NOR gates in its path. We will see shortly what this activation sequence is, and how it works. If the total number of ScanSeal circuits being used is k and the total number of flip flops is n, then my *activation sequence* would look like:

$$\overbrace{1010....}^{k\ bits}\underbrace{00...}_{n\ bits}$$

The way this sequence works is very simple. The first bit which is 1 is undoubtedly going to reach the first ScanSeal circuit thereby making its corresponding inactive NOR gate active. Now, this 1 gets inverted and similarly, all the following bits of the activation sequence get inverted on successive clocks. So, the second bit of the sequence now becomes 1 which activates the next inactive NOR gate in the chain. This bit and the following bits get inverted due to this effect. By then, third bit of the sequence becomes 1 which activates the next NOR gate. This continues until every NOR gate gets activated. This procedure takes a total of n+k clock cycles. In this way, the user can work with the hardware in test mode with all the NOR gates activated. This *activation sequence* does not depend on the position of the ScanSeal circuits placed in the scan chain but somewhat depends on their total number. Also, this sequence need not be unique. For example, the trailing n bits need not be all zeroes. It can be anything. And, the leading sequence of ones and zeroes need not be that too. Intuitively, one can see that the activation sequence is a concatenation of two patterns. The first pattern should have a change in polarity at least k times and the second pattern can be anything of size n bits. Hence, there is no way the attacker can ascertain the total number or the positions of ScanSeal circuits using the knowledge of activation sequence.

5.5.3 Demonstration of an Activation Sequence

Consider the network of scan flip-flops in a scan chain as in Figure 5.3, only difference being is that instead of inverters, we have ScanSeal circuits in place of them. Assume that Reset signal has been applied and we are giving in the activation sequence *101010000000000*. The states of flip-flops at each clock is shown in Table 5.2. SO is used to refer scan_out. The designed circuits are placed at positions 1, 3, 6, 7 and 11. NOR gate at position 1 gets activated in 1^{st} clock, that at position 3 gets activated in 3^{rd} clock, at position 6 in 7^{th} clock, at position 7 in 9^{th} clock and finally at position 11 in 14^{th} clock. The final state is indeed dependent on the activation sequence and positions of circuits placed, but again, an attacker doesn't know these positions and hence,

ffs →	F/F1	F/F2	F/F3	F/F4	F/F5	F/F6	F/F7	F/F8	F/F9	F/F10	SO
clks ↓	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0
5	0	1	1	0	0	0	0	0	0	0	0
6	1	0	0	1	0	0	0	0	0	0	0
7	1	1	1	0	1	0	0	0	0	0	0
8	1	1	0	1	0	0	0	0	0	0	0
9	1	1	0	0	1	1	0	0	0	0	0
10	1	1	0	0	0	0	0	0	0	0	0
11	1	1	0	0	0	1	1	0	0	0	0
12	1	1	0	0	0	1	0	1	0	0	0
13	1	1	0	0	0	1	0	0	1	0	0
14	1	1	0	0	0	1	0	0	0	1	0
15	1	1	0	0	0	1	0	0	0	0	0

Table 5.2: Activation Sequence Propagation

is unable to locate them.

5.5.4 When and how many times should it be applied?

The user should be careful about application of this sequence after every use of the reset signal. This is so because, if the reset is applied and the user wants to operate it under the test mode after operating it in normal mode, then, the data captured by the device after normal mode will be all destroyed even with the application of a single clock in test mode. So, the user should make it sure to apply an activation sequence after every application of reset signal or to be on the safe side, even after the device is switched on.

In the following section, we will see the security induced by this modified architecture in the Trivium hardware and then we will move on to the section devoted to its performance evaluation.

5.6 Security induced in Trivium

Using the design of the proposed architecture, the attacker cannot control or observe the values of the internal state registers of the trivium hardware through the scan inputs and outputs due to unknown position of the ScanSeal circuits. Hence, the steps of the attack described in the section 4.3 are not successful in breaking this hardware stream cipher. Trivium uses 288 flip flops, so if we place 288/2 = 144 ScanSeal circuits at certain locations in the scan chain, as per [16], the probability to guess the correct structure is approximately $1/2^{288}$, which is too less. Thus, we can avoid the scan based attack.

ISCAS'89 circuit	Total flip-flops	Logic gates in the ckt.	Total gate count	ScanSeal circuits used	Gate overhead	Overhead as in [11]
s298	14	119	287	5	5.23%	21%
s344	15	160	340	5	4.42%	18%
s382	21	158	410	7	5.13%	19%
s400	21	162	414	7	5.08%	19.4%
s5378	179	2779	4927	60	3.66%	17%
s9234	228	5597	8333	76	2.74%	17.7%
s13207	669	7951	15979	223	4.19%	16.4%
s15850	597	9772	1936	199	3.53%	17%
s35932	1728	16065	36801	576	4.7%	15.8%
s38417	1636	22179	41811	546	3.92%	16.4%

Table 5.3: Hardware Overhead

We may place different number of NOR circuits. For example, placing 288/3 = 96 ScanSeals would give us a security margin of $1/\,{}^{289}C_{96}$ which is extremely good. But placing half the number of total flip-flops provides the maximum security. This security analysis is based on probability theory as described in the section 5.2.3.

5.7 Performance Evaluation

In this section, area overhead is evaluated for the ScanSeal Architecture.

First, the overhead that would be incurred while implementing the design given in this work is compared with the results obtained in [11]. In [11], the authors have considered ISCAS'89 benchmark circuits [2]. They generated scan tree after finding compatible scan cells from the test set, as in [1]. The output from this scan tree was then fed into aliasing free compactor to match the output with the expected output. This compactor is supposed to be designed by the design engineer given he knows the test patterns and test responses. Area overhead for this design was computed using Synopsys Tetramax and Design Compiler tool. Overhead over the same set of circuits has been evaluated because these are already tested benchmark circuits. Refer to the Table 5.3. Since, area overhead and gate overhead are comparable figures, we have computed gate overhead in this work. Using the fact that 12 NAND gates are needed to synthesize a D-flip-flop, we have computed total number of gates required for the synthesis of ISCAS'89 circuits. Also, total number of NOR gates used in our design is 3 for each of the ScanSeal circuits. And, we have taken the count of total ScanSeal circuits to be one-third of total number of flip flops as these many are sufficient from security perspective. As can be seen, we have a fair improvement from the results in [11]. In the next paragraph, overhead analysis for AES cryptosystem is discussed.

In [12], end to end design of AES is done using 0.18-μm CMOS technology. The

AES Architecture	scheme in [18]			scheme in [16]			our scheme		
	Gates	Overhead		Gates	Overhead		Gates	Overhead	
		Gate	%		Gate	%		Gate	%
with KS	273,183	412	0.15	184,209	80+159	0.12	184,209	240	0.13
without KS	282,120	4620	1.64	57,017	80+159	0.41	57,017	240	0.42

Table 5.4: Overhead analysis in AES implementation

work in [16] is based on this design. On the other hand, the work in [18] is based on the AES hardware implementation given in [10]. We have considered the same implementation as in [16] and got our results which is shown in table 5.4. Overhead analysis is done under two conditions: with key scheduling (number of flip-flops 6336) and without key scheduling (number of flip-flops 4048). 80 inverters were used in [16]- 10 in each of the 8 scan chains each containing equal number of flip-flops. What we have done is replaced these inverters with ScanSeal circuits - one for one - and computed the overhead. Obviously, our overhead compared to what is in [16] will be slightly greater (but our scheme is more secure as it is resilient to the attack explained in section 5.3) but, the results are fairly good when compared to the percentage overhead in [18](Table 5.4)

Chapter 6

Conclusion and Future Work

6.1 Summary

We have seen that hybrid model of power approximation works better than existing models and has an added advantage of being less complex in the sense that it is easy to implement. The basic idea of this model was to divide a cubical representation of core into multiple sub-cubes. Further, a scan based attack on Trivium was demonstrated Then, an attack on FSC architecture which claimed to be secure against scan based attack was also designed. We modified this architecture to get a new architecture, which we named as ScanSeal Architecture, and we found it robust enough to withstand the aforementioned attack. Moreover, its design overhead was less than other existing designs.

6.2 Future Path

The future work of this project can go in the following directions:

- The hybrid model of power approximation can be implemented and used in any of the heuristics for solving constrained 3D-bin packing problem with slight modification in the original heuristic.

- The designed hardware resilient to scan based attack can be incorporated in any actual hardware design like AES using CAD tools and compare the actual area overhead with the theoretical overhead shown in this work.

- A distinct class of stream ciphers can be identified which are vulnerable to this kind of scan based attack. This can be extended to block ciphers too.

Bibliography

[1] Y. Bonhomme, T. Yoneda, H. Fujiwara, and P. Girard. An efficient scan tree design for test time reduction. In *Proc. 9th IEEE ETS*, pages 6–11, November.

[2] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE Int. Symp. on Circuits and Sys.*, pages 1929–1934, May 1989.

[3] C. D. Caniere and B. Preneel. Trivium specifications. eSTREAM submitted papers.

[4] R. Chou, K. Saluja, and V. Agrawal. Scheduling tests for vlsi systems under power constraints. *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, 5(2):175–185, 1997.

[5] D. Hely, M. Flottes, F. Bancel, B. Rouzeyre, N. Berard, and M. Renovell. Scan design and secure chip. In *Proc. 10th IEEE IOLTS*, pages 219–226, July 2004.

[6] Y. Huang, W. Cheng, C. Tsai, N.Mukherjee, O. Samman, Y. Zaidan, and S. Reddy. Resource allocation and test scheduling for concurrent test of core bosed soc design. In *Proc. IEEE Asian Test Symposium*, pages 348–353, 2005.

[7] Y. Huang, W. Cheng, C. Tsai, N.Mukherjee, O. Samman, Y. Zaidan, S. Reddy, and P. Reuter. Optimal core wrapper width selection and soc test scheduling based on 3-d bin packing algorithm. In *Proc. IEEE Internatioanl Test Conference (ITC)*, pages 74–82, 2002.

[8] R. Kapoor. Security vs. test quality: Are they mutually exclusive? In *Proc. ITC*, page 1414, October 2004.

[9] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic. Securing scan design using lock and key technique. In *Proc. 20th IEEE Int. Symp. DFT VLSI Syst.*, pages 51–62, 2005.

[10] S. Mangard, M. Aigner, and S. Dominikus. A highly regular and scalable aes hardware architecture. *IEEE Transactions of Computers*, 52(1):483–491, April 2004.

[11] D. Mukhopadhyay, S. banerjee, D. RoyChowdhury, and B. Bhattacharya. Cryptoscan: Secured scan chain architecture. In *Proc. 14th IEEE ATS*, pages 265–270, 2001.

[12] D. Mukhopadhyay and D. Roychowdhury. An efficient end to end design of rijndael cryptosystem in 0.18 μ cmos. In *Proc. 18th Int. Conf. VLSID*, pages 405–410, January 2005.

[13] P. Rosinger, B. Al-Hashimi, and N. Nicolici. Power profile manipulation: a new approach for reducing test application time under power constraints. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 21(10):1217–1225, October 2002.

[14] S. Samii, E. Larsson, K. Chakrabarty, and Z. Peng. Cycle-accurate test power modeling and its application to soc test scheduling. In *Proc. of IEEE International Test Conference(ITC)*, October 2006.

[15] R. Sankaralingam, R. Oruganti, and N. Touba. Static compaction techniques to control scan vector power dissipation,. In *Proc. of IEEE VLSI Test Symposium*, pages 35–40.

[16] G. Sengar, D. Mukhopadhyaya, and D. RoyChowdhury. Secured flipped scan chain model for crypto-architecture. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 26(11):2080–2084, November 2007.

[17] B. Yang, K. Wu, and R. Karri. Scan based channel attack on dedicated hardware implementation of data encryption standard. In *Proc. ITC*, pages 334–344, October 2004.

[18] B. Yang, K. Wu, and R. Karri. Secure scan: A design for test architecture for crypto chips. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 25(10):2287–2293, October 2006.