ABSTRACT

Title of Dissertation: SCAN CHAIN BASED HARDWARE SECURITY

Xi Chen, Doctor of Philosophy, 2018

Dissertation directed by: Professor Gang Qu
Department of Electrical and Computer Engineering

Hardware has become a popular target for attackers to hack into any computing and communication system. Starting from the legendary power analysis attacks discovered 20 years ago to the recent Intel Spectre and Meltdown attacks, security vulnerabilities in hardware design have been exploited for malicious purposes. With the emerging Internet of Things (IoT) applications, where the IoT devices are extremely resource constrained, many proven secure but computational expensive cryptography protocols cannot be applied on such devices. Thus there is an urgent need to understand the hardware vulnerabilities and develop cost effective mitigation methods.

One established field in the semiconductor and integrated circuit (IC) industry, known as IC test, has the goal of ensuring that fabricated ICs are free of manufacturing defects and perform the required functionalities. Testing is essential to isolate faulty chips from good ones. The concept of design for test (DFT) has been

integrated in the commercial IC design and fabrication process for several decades. Scan chain, which provides test engineer access to all the flip flops in the chip through the scan in (SI) and scan out (SO) ports, is the backbone of industrial testing methods and can be found in almost all the modern designs. In addition to IC testing, scan chain has found applications in intellectual property (IP) protection and IC identification. However, attackers can also leverage the controllability and observability of scan chain as a side channel to break systems such as cryptographic chips. This dissertation addresses these two important security problems by proposing (1) a practical scan chain based security primitive for IP protection and (2) a partial scan chain framework that can mitigate all the existing scan based attacks.

First, we observe the fact that each D-flip-flop has two output ports, Q and Q', designed to simplify the logic and has been used to reduce the power consumption for IC test. The availability of both Q and Q' ports provide the opportunity for IP protection. More specifically, we can generate a digital fingerprint by selecting different connection styles between adjacent scan cells during the design of scan chain. This method has two major advantages: fingerprints are created as a post-silicon procedure and therefore there will be little fabrication overhead; altering the connection style requires the modification of test vectors for each fingerprinted IP and thus enables a non-intrusive fingerprint verification method. This addresses the overhead and detectability problems, two of the most challenging problems of designing practical IP fingerprinting techniques in the past two decades. Combined with the recently developed reconfigurable scan networks (RSNs) that are popular for embedded and IoT devices, we design an IC identification (ID) scheme utilizing the different connection styles. We perform experiments on standard benchmarks to demonstrate that our approach has low design overhead. We also conduct security analysis to show that such fingerprints and IC IDs are robust against various attacks.

In the second part of this dissertation, we consider the scan chain side channel attack, which has been reported as one of the most severe side channel attacks to modern secure systems. We argue that the current countermeasures are restricted to the requirement of providing direct SI and SO for testing and thus suffers the vulnerability of leaving this side channel open to the attackers as well. Therefore, we propose a novel public-private partial scan chain based approach with the basic idea of removing the flip flops that store sensitive information from the scan chain. This will eliminate the scan chain side channel, but it also limits IC test. The key contribution in our proposed public-private partial scan chain design is that it can keep the full test coverage while providing security to the scan chain. This is achieved by chaining the removed flip flops into one or more private partial scan chains and adding protections to the SI and SO ports of such chains. Unlike the traditional partial scan design which not only fails to provide full fault coverage, but also incur huge overhead in test time and test vector generation time, we propose a set of techniques to ensure that the desired test vectors can be entered into the system efficiently. These techniques include test vector reordering, test vector reusing, and test vector generation based on a novel finite state machine (FSM) structure we have invented. On the other hand, to enable the test engineers the ability to observe the test output to diagnose the chip while not leaking information to the attackers, we propose two lightweight mechanisms, one based on linear feedback shift register (LFSR) and the

other one based on configurable physical unclonable function (PUF). Finally, we discuss a protocol on how in-field test can be realized using our public-private partial scan chain. We conduct experiments with industrial scan design tools to demonstrate that the required hardware in our approach has negligible area overhead and gives full test coverage with reduced test time and does not need to re-generate test vectors.

In sum, this dissertation focuses on the role of scan chain, a conventional design for test facility, in hardware security. We show that scan chain features can be leveraged to create practical IP protection techniques including IP watermarking and fingerprinting as well as IC identification and authentication. We also propose a novel public-private partial scan design principle to close the scan chain side channel to the attackers. Through this dissertation work, we demonstrate that it is possible to develop highly practical scan chain based techniques that can benefit both the community of IC test and hardware security.

SCAN CHAIN BASED HARDWARE SECURITY

by

Xi Chen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Gang Qu, Chair/Advisor
Professor Dana Dachman-Soled
Professor Manoj Franklin
Professor Robert Newcomb
Professor Larry Washington, Dean's Representative

Dedication

To my parents

Acknowledgements

First and foremost, I am deeply indebted to my advisor Prof. Gang Qu for his fundamental role in my doctoral work. He has always been kind and supportive since the first day I joined the group. His valuable guidance on my research and study helped me get through the tough times in my pursuit of the doctoral degree. I have been extremely lucky to have an advisor who cares so much about his students and is willing to contribute significant amount of his valuable time and ideas to help them. Over the last few years, he has advised me with patience in our research on scan chain based hardware security. Without his enthusiasm, motivation and immense knowledge, the completion of this doctoral dissertation would have been impossible. In addition to his academic support, I greatly value his personal and emotional support for my life. I quite simply cannot imagine a better advisor.

I would also like to express my sincere gratitude to our collaborator, Dr. Aijiao Cui, who is an expert in scan chain. She is always available whenever I needed help. I have benefited a lot from all the discussion we had in throughout my research.

I am also thankful to my dissertation committee members. Special thanks to Prof. Robert Newcomb for his careful proofreading and insightful comments on my dissertation. His diligence and pursuit of excellence would always inspire me. I want to acknowledge Prof. Manoj Franklin for his expertise in scan chain and all the great questions and suggestions he provided to advance my research. I'm also appreciative of Prof. Dana Dachman-Soled and Prof. Larry Washington for their time and valuable feedback that shaped my final dissertation.

Many thanks go out to the help and support from all the members in our group, especially Omid Aramoon, Zhaojun Lu, Mingze Gao, Qian Wang and Md Tanvir Arafin for all the thoughtful discussions and the wonderful time.

My Ph.D. study was in part sponsored by the Air Force Office of Scientific Research under the grant “MURI: Developing of Universal Security Theory for Evaluation and Design of Nanoscale Devices”.

Last but not least, I want to thank my family for their unconditional love and support. They experienced all the ups and downs of my whole life and stayed by my side as always.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AES	Advanced encryption standard
ATPG	Automatic Test Pattern Generation
BIST	Built-in self-test
CI	Control input
CUT	Circuit under test
DES	Data encryption standard
DFT	Design for Testing
ECC	Elliptic-curve cryptography
FPGA	Field-programmable gate array
FSM	Finite State Machine
MKR	Mirror key registers
IC	Integrated circuit
ID	Identification
IoT	Internet of Things
IP	Intellectual property
JTAG	Joint Test Access Group
LFSR	Linear feedback shift register

ODC	Observability don’t care
PCB	Printed circuit board
PI	Parallel input port
PO	Parallel output port
PUF	Physical unclonable function
RO	Ring oscillator
RSA	Rivest–Shamir–Adleman cryptosystem
RSN	Reconfigurable scan network
SAT	Boolean satisfiability problem
SDC	Satisfiability don’t care
SE	Scan enable
SI	Scan in
SIB	Segment insertion bit
SO	Scan out
SoC	System-on-a-Chip
TAP	Test access port
VLSI	Very large scale integration
XOR	Exclusive or

Chapter 1: Introduction

Besides delivering the correct functionality, the objectives of integrated circuit (IC) design have evolved from area and delay minimization to the optimization of testability, power consumption, and manufacturability. In the past 20 years, various IC security concerns have gained a lot of attention. In this dissertation, we study the role of scan chain, a traditional design for testing (DFT) technology, in hardware security. More specifically, we focus on two challenging problems: (1) developing practical scan chain based intellectual property (IP) protection techniques and (2) designing scan chains that are secure against side channel attacks.

In this chapter, we first introduce scan chain as a popular DFT technique in section 1.1. Then we discuss IP protection and give the rationale that scan chain can help to build robust and practical IP protection solutions in section 1.2. After that, we give a brief overview of the security vulnerabilities in scan chains and why it is hard to prevent scan chain side channel attacks in section 1.3. We present the key ideas and main contributions of this dissertation in section 1.4. The structure of this dissertation is given in section 1.5 to conclude this chapter.

1.1 Design for Testing Technique

Driven by the huge demands from the emerging applications and market, the semiconductor industry races to pack as many hardware as possible into their products, causing the very large scale integration (VLSI) of IC design to move quickly from thousands of logic gates in the late 1970's to nowadays billions of logic gates. Consequently, this leads to the enormous complexity inherent in the VLSI design. Meanwhile, competitors do everything possible to improve product performance and shorten the time-to-market. More importantly, as ICs have evolved from simple computational devices to the pervasive things in our daily life in the forms of cloud servers, industrial control systems, routers, autonomous vehicles, smart phones, implantable medical devices, smart home appliances as well as various devices in the Internet of Things (IoT) applications, they play a much more critical role in our quality of life and could be life threatening in many scenarios. Therefore, it was identified in the 1980s about the need of chip testing to ensure that the fabricated ICs are free of manufacturing defects and perform the required functionalities before they can be delivered to customers [1].

However, this turned out to be an extremely hard problem that often requires solving many NP-complete problems such as the Boolean Satisfiability (SAT) problem. It is known that the manufacturing process cannot be 100% error free. The defects in silicon could contribute towards the errors introduced in the physical device. Although formal verification can ensure the correct functionality, it may fail to detect manufacturing variation caused defects which are becoming more and more popular with the technology scaling to the end of Moore's Law [2]. Running all the functional tests on each copy of the manufactured devices to guarantee correct functionalities is extremely time-consuming and unrealistic because of the number of transistors on the device (could be in billions), the number of copies of the devices (might be in millions), and small amount of time and budget for testing. The concept of DFT was proposed as a practical solution where a number of test and debug features is added in design time to largely reduce test complexity and improve fault coverage. It has turned into an essential feature of the System-on-a-Chip (SoC) design methodology today.

Simply stated, DFT is some extra logic circuitry that the IC designers put in the design to facilitate the design to be testable after production, named production tests. By making possible that the production tests can be applied at the end of the VLSI manufacturing process, test engineers can validate that each fabricated chip is free of manufacturing defects. From this perspective, DFT is a detection technique to tell whether a physical device is faulty or not. Since it only detects production faults and is not designed to correct the faults, if a chip is found faulty, it will be discarded and never be shipped to customers. Since every single device must be tested after production, how to minimize the test cost, in terms of the time and power consumption required for the test, becomes a prominent issue. In addition, a practical testing method should have a very high probability of detecting faults on the chips to avoid possible catastrophic consequences of inadequate testing.

Scan chain is one of the most successful DFT techniques and can be found in almost all the modern designs. The insertion of scan chain, which involves adding extra logic and signals dedicated for testing, occurs after a design is verified to be functionally correct. A scan cell, which is the building block of the scan structure, consists of a conventional D flip-flop and a 2-to-1 multiplexer connected to the D input port of the D flip-flop. All the scan cells are then connected in the form of a chain, known as scan chain. Recall that system's internal states are stored in the memory or register, where the basic storage unit is a flip-flop that stores one single bit of information, 1 or 0. Scan chain allows test engineers to set the value of each flip-flop through a dedicated scan in (SI) port. It also allows the test engineers to observe the content of each flip-flop through a scan out (SO) port. In this way, test engineers are provided with the full controllability and observability of the circuit's internal states. Thus under the test mode, all the sequential elements are controllable by the external test pins, basically turning the sequential circuit into a combinational design. This solves the challenging problem confronting test engineers for a long time by avoiding sequential Automatic Test Pattern Generation (ATPG), which is much more complex than the combinational ATPG due to the unknown starting state and extremely long test sequences. By inserting scan chain in the design, only combinational ATPG is required to generate test patterns, which has greatly improved test efficiency and test reliability.

There have been many variations of scan chains. In this dissertation, we will focus on the following three types of scan chains: partial scan chain, multiple scan chains and reconfigurable scan networks (RSNs). Partial scan chain was proposed a couple of decades ago to minimize the area overhead and scan sequence length which will reduce both test time and test power. In a partial scan chain, only a subset of flip-flops will be scanned instead of all the flip-flops in the tradition scan chain. Multiple scan chains are applied to reduce test application time, but it requires efficient compression and decompression techniques to reduce test data volume. It partitions a single scan chain into several chains. RSNs, as its name suggests, allow the scan structure to be configured to reduce access time of scan structures since irrelevant segments will be bypassed during test. A detailed description of these scan structures will be presented in Chapter 2.

A few other well-known structured DFT techniques have been adopted in the industry for decades as well. Built-in self-test (BIST) allows the circuit to perform self-testing by adding an on-chip pattern generator to generate test patterns which are fed into the scan chain, and an on-chip result compressor to compress the scanned out responses of all patterns into a final signature, which is then compared with the expected signature to determine the correctness of a circuit. This technique lowers the test cost since there is no need to supply most of the test patterns from an external tester, thus reducing required test pins. Meanwhile, it can enable high speed tests that allow testing at the speed of actual operation. However, the fault coverage achieved by BIST does not meet test requirements in some cases. It does not scale well as the size of the circuit increases.

Boundary scan, also known as Joint Test Access Group (JTAG), provides test access to components embedded in a system. It was originally developed in the mid-1980s to address the increasing difficulty of testing printed circuit boards (PCBs). After its introduction as an industry standard in 1990, JTAG has been in widespread use. The original motivation for JTAG was boundary scan testing, a method for gaining direct control of the I/O pins at the boundary of a chip during test. This enables efficient testing of the interconnections between devices that are mounted on a circuit board. Today, everything from testing interconnects and functionality on ICs to programming flash memory of systems deployed in the field and everything in-between can be accomplished through JTAG. In that sense, JTAG has become a common hardware interface that provides external environments with a way to communicate directly with the chips on a board for debugging, programming and testing. It offers two significant advantages. First, its serial interface requires only a minimum set of test access pins. Second, it facilitates design re-use and standard protocols since it is an IEEE standard.

At the heart of a JTAG network is a Test Access Port (TAP) controller, the most common block used in support of on-chip testing. In essence, it is a state machine that controls the behavior of the JTAG network. The state machine can be divided into three sections. One section consists of the reset state and the run test state. The two remaining sections are essentially duplicates, except that one pertains to the data register and the other pertains to the instruction register. A JTAG network typically has four standard pins plus one optional pin. The Test Clock pin dictates the speed of the TAP controller. The Test Mode Select pin controls what action JTAG takes, that is the transitions in the FSM of the TAP controller. The Test Data Input pin feeds data into the chip and the Test Data Output pin reads data out of the chip. The fifth optional pin, Test Reset, is used to reset JTAG to a known good state. There are two types of registers associated with boundary scan. Each compliant device has one instruction register, holding the current instruction used by the TAP controller, which then decides what to do with the received signals. Two or more data registers may also be included.

In sum, DFT techniques provide an elegant solution to the challenging problem of IC testing. With the invention of various DFT techniques, in particularly the scan chain design, the test efforts are notably reduced to satisfy the complex test needs. The goal of this dissertation is to research the role of scan chain in hardware security, which is another important challenge for today's IC design. The following two sections provide the motivation of our work.

1.2 Intellectual Property Protection

Intellectual property (IP) refers to a category of properties that includes intangible creations of the human intellect, and primarily encompasses copyrights, patents, and trademarks. In this dissertation, we will discuss IPs in the semiconductor industry, where the IPs have a very similar concept as the traditional ones but require quite different protection schemes. In electronic design, an IP core or IP block is a reusable unit of logic, cell or integrated circuit that belongs to some designer or company. IP cores fall into three categories: hard IPs, firm IPs and soft IPs.

- Hard IP cores refer to designs that are delivered to the customer in the unchangeable forms such as a plug-and-play component, a memory system, or a final silicon layout. These cores are carefully designed and fine-engineered to provide the best possible performance in terms of area, delay, and/or power consumption. They cannot be customized for different process technologies, making them not portable among different manufacturers. Thus hard cores are the most restrictive form for IP core delivery.
- Firm IP cores are also known as semi-hard IP cores. They are a form of gate-level netlist where the users of the cores have the flexibility to place the module as per usage. For this purpose, firm IPs normally provide some level of user-programmable configurations. In firm cores, modifications are allowed to some extent (most of the time to the minimal level) but the portability is still limited.

The performance of firm IPs is not as good as the hard IPs, but is normally predictable to help designers make the decision of whether to reuse them.

- Soft IP cores are designs that can be integrated into large designs and re-synthesized. The most popular soft IPs are synthesizable register transfer level (RTL) codes. They are flexible and can be modified at ease to fit into the large design that they will be a part of. Soft IPs do not depend on any specific technology and have the best portability. However, their performance is unpredictable as it depends on many factors from the technology library being used to the experience of the designers.

As the size and complexity involved in designing ICs are continuously outpacing the designer's ability to use the available silicon in a meaningful manner, known as the productivity gap, IP reuse based design methodology was introduced in the early 1990's [3]. It effectively shortened time-to-market windows and reduced design cost. That being said, IP reuse based system design was quickly adopted as an industrial standard, where designs are shared among multiple parties in the forms of hard/firm/soft IPs. Because of the lucrative incentive of IP reuse and the sharing nature of it, this reuse based design methodology is vulnerable to various IP infringements such as IP theft and misuse. Indeed, IP protection was identified as one of the original challenges for design reuse to thrive [4]. Apparently, legal methods such as copyright, patent, and trade secrets are needed, but they are not sufficient due to the lack of technical methods to identify and trace IPs in order to protect the rights of both IP providers and IP buyers. A paradigm of digital IP watermarking, IP fingerprinting, and IC metering was proposed between 1998 and 2001 with most of the early efforts included in a monograph published in 2003 [5].

IP watermarking was first reported in 1998 [6-8]. Its concept is similar to watermarking multimedia artifacts, where secret information is embedded in the content as the proof of ownership of the artifacts. Practical watermarking techniques must be robust and resilient against potential attacks, be readily detected to prove authorship with high confidence, incur no or minimal negative impacts on the quality and performance of the artifact, and take reasonable efforts to embed and detect the watermark. In addition, IP watermarking must fully preserve IP's functionality, which makes it much more challenging. Over the years, watermarking methods have been applied to protect IPs in different forms, such as Verilog codes [9], combinational logic [6, 10], sequential circuits [11], finite state machines (FSM) [12], field-programmable gate array (FPGA) designs [13], physical designs [7] as well as computer-aided design (CAD) tools [14]. A survey can be found in [15].

Although watermarking can establish authorship when IP piracy such as illegal redistribution occurs, it offers little help in tracking down the guilty buyer who should be responsible for the illegal act. To overcome this problem, the concept of IP fingerprinting was introduced in 1999 [16, 17]. IP fingerprinting embeds each buyer's signature along with the designer's watermark in the IP such that each IP is unique. This attribute enables IP providers to trace individual buyers, which can help identify guilty buyers and thus protect legal buyers in case IP piracy occurs. In short, one can view fingerprints as customized watermarks. A practical fingerprinting method inherits all the aforementioned properties for watermarking. In addition, it should be resilient against various attacks that do not apply to watermarking methods, particularly the collusion attack where an attacker obtains multiple copies of the same IP with different fingerprints and compares them to reveal fingerprint locations and then removes or forges fingerprints. Furthermore, since the fingerprint will make each copy of the IP distinct, one problem that remained unsolved for almost two decades is how to design practical, mainly in terms of cost, IP fingerprinting methods. In this dissertation, we address this challenge by proposing a scan chain based practical IP fingerprinting scheme.

IC metering is a protocol that enables the IC designer to trace and control each copy of the fabricated ICs that contains the protected IPs to prevent a dishonest foundry from overbuilding chips [18]. This normally requires a non-alterable identification. The cost-effective serial number technique is perhaps the most popular and one of the earliest ways for IC identification.

A serial number can be physically indented on the device or stored permanently in the memory. Unfortunately, the fact that it can also be easily removed or forged makes it unsuitable to countermeasure IP theft such as illegal reproduction, redistribution, and foundry overbuilding. Several intrinsic unclonable IC tagging schemes based on silicon manufacturing variations have been proposed. The ICID tag technique was first proposed in 2007 [19]. In this technique, a sequence of control signals selects an array of transistors to drive a capacitive load. The output voltage differs for each chip due to inherent IC manufacturing process variations. Because such variations are random and uncontrollable, an ICID is considered an unclonable tag and thus can be applied against IC over-building. In addition, the unique challenge response pairs created by the physical unclonable functions (PUF) [20] have also been used for IC identification. The emerging PUF uses the unclonable intrinsic fabrication variations in delay, capacitance, threshold voltage, and other sources to produce a unique response to a given challenge to authenticate a device. These can be used for IC identification, but they cannot detect IP theft because illegally copied or over-built ICs will have different variations and hence different identifications from the original copy.

In this dissertation, we extend the existing scan chain based IP watermarking to IP fingerprinting and IC authentication. There are two different styles to connect one flip-flop to another during the scan chain design. It has been utilized to minimize test power consumption [21] and more recently to design IP watermark by deliberate selection of such connection styles [22]. We observe that this can be conveniently used to create IP fingerprint and IDs with two advantages. First, we propose to do this after chips are fabricated and thus make it affordable. Second, scan chain's controllability and observability features provide a non-intrusive way to verify such fingerprints or IDs. We will elaborate this in Chapter 3 and Chapter 4.

1.3 Security Vulnerabilities of Scan Chain

Nowadays, the Internet provides the essential communication to connect literally billions of people through various devices that are powered by one or multiple ICs. The need for secure storage and transferring of information has become indispensable under the escalating attacks. One ancient art that is still widely used today is cryptography [23], which is most closely associated with the development and creation of the mathematical algorithms used to encrypt and decrypt messages. These encryption/decryption algorithms are traditionally implemented in software for portability and flexibility, but the performance of software implementation is a notable problem. Hence, hardware implementations of cryptographic algorithms are widely deployed as an alternative solution due to its efficiency in run-time and power consumption. This becomes more and more evident with the development of IoT applications where resources are extremely constrained.

When we look at the shift of attacking in the past half century, we see that the attackers started with the attempts to break the cryptographic algorithms, with which they have enjoyed some early success. But such vulnerabilities normally are quickly fixed and today's cryptography is built on solid mathematical foundations and no one tries to break a secure system from its cryptography underpinning. Then the attacking efforts shifted to the vulnerabilities in the software implementation of these cryptographic algorithms and the networking protocols, which are not designed with security in mind. After several decades of battles between the attackers and the defenders, standards to build secure software and network to support the cryptographic algorithms were established. Next, the attackers put their eyes on the hardware implementations, where designer's top goal has always been performance. Therefore, we have seen many physical attacks in the past couple of decades. Among them, side channel attacks have been extensively studied to crack a large amount of supposedly secure systems. This type of attack takes advantages of the information leakage from the hardware, including timing [24], power [25], electromagnetic radiation [26], error message [27], etc., to obtain the secret data that have been processed. This dissertation focuses on the security of scan chain side channel.

As we have mentioned earlier, test engineers can shift in test patterns through the dedicated SI port and shift out test responses through the SO port. For example, by switching between normal execution mode and test mode, attackers can access the intermediate states during encryption which can help them retrieve the cipher keys. Successful attacks based on scan chain have been reported on many cryptographic chips, including DES [28, 29], AES [30], ECC [31], and RSA [32]. In the test-mode-only attacks, by exploiting scan chain's controllability and observability, it is also possible to break the system through a scan chain. In [33, 34], the authors explore the vulnerability of letting the system enter the unspecified or undesirable states, which they achieved through the insecure system implementation. When a scan chain is available, this becomes trivial due to the controllability of a scan chain.

We will elaborate these attacks and their countermeasures as well as other related work in a later chapter. Here we just mention that the countermeasures are mainly based on either controlling the switch between normal mode and test mode, or obfuscating the scan output values to confuse the attackers. We believe that securing a scan chain is a very hard problem because one cannot distinguish test engineers and scan chain attackers, who have the same goal to understand the internal states of the system through the controllability and observability provided by the scan chain. Controlling the switch between normal mode and test mode or obfuscating the scan output make it harder for the attackers to launch scan chain side chain attacks, but they also make IC test inconvenient. In this dissertation, we propose a novel approach based on the public-private partial scan chain design to provide both security and test coverage. The basic idea is to keep some flip-flops in private partial scan chains and control their SI and SO to avoid information leak to the attackers. We will establish the foundation of this approach in Chapter 5 and elaborate how we can secure the SI and SO ports in the private scan chains in Chapter 6 and Chapter 7, respectively.

1.4 Key Contributions

This dissertation addresses two important security problems related to scan chain by proposing (1) a practical scan chain based security primitive for IP protection and (2) a partial scan chain framework that can mitigate all the existing scan based attacks.

First, we observe the fact that each D-flip-flop has two output ports, Q and Q’, designed to simplify the logic and has been used to reduce the power consumption for IC test. The availability of both Q and Q’ ports provide the opportunity for IP protection. More specifically, we can generate digital fingerprint by selecting different connection styles between adjacent scan cells during the design of scan chain. This method has two major advantages: fingerprints are created as a post-silicon procedure and therefore there will be little fabrication overhead; altering the Q-SD or Q'-SD connection style requires the modification of test vectors for each fingerprinted IP and thus enables a non-intrusive fingerprint verification method. This addresses the overhead and detectability problems, two of the most challenging problems of designing practical IP fingerprinting techniques in the past two decades. Combined with the recently developed reconfigurable scan networks (RSNs) that are popular for embedded and IoT devices, we design an IC identification scheme utilizing the Q-SD and Q’-SD connection styles. We perform experiments on standard benchmarks to demonstrate that our approach has low design overhead. We also conduct security analysis to show that such fingerprints and IC IDs are robust against various attacks. Another ongoing work along this line is to develop a device authentication protocol based on RSNs.

In the second part of this dissertation, we consider the scan chain side channel attacks, which have been reported as one of the most severe side channel attacks to modern secure systems. We argue that the current countermeasures are restricted to the requirement of providing direct SI and SO for testing and thus suffer the vulnerability of leaving this side channel open to the attackers as well. Therefore, we propose a novel public-private partial scan chain based approach with the basic idea of removing the flip-flops that store sensitive information from the scan chain. This will eliminate the scan chain side channel, but it also limits IC test. The key contribution in our proposed public-private partial scan chain design is that it can keep the full test coverage while providing security to the scan chain. This is achieved by chaining the removed flip-flops into one or more private partial scan chains and adding protections to the SI and SO ports of such chains. Unlike the traditional partial scan design which not only fails to provide full fault coverage, but also incur huge overhead in test time and test vector generation time, we propose a set of techniques to ensure that the desired test vectors can be entered into the systems efficiently. These techniques include test vector reordering, test vector reusing, and test vector generation based on a novel finite state machine (FSM) structure we have invented. On the other hand, to enable the test engineers the ability to observe the test output to diagnose the chip while not leaking information to the attackers, we propose two lightweight mechanisms, one based on linear feedback shift register (LFSR) and the other one based on configurable physical unclonable function (PUF). Finally, we discuss a protocol on how in-field test can be realized using our public-private partial scan chain. We conduct experiments with industrial scan design tools to demonstrate that the required hardware in our approach has negligible area overhead and gives full test coverage with reduced test time and does not need to re-generate test vectors.

In sum, this dissertation focuses on the role of scan chain, a conventional design for testing facility, in hardware security. We show that scan chain features can be leveraged to create practical IP protection techniques including IP watermarking and fingerprinting as well as IC identification and authentication. We also propose a novel public-private partial scan design principle to close the scan chain side channel to the attackers. Through this dissertation work, we demonstrate that it is possible to develop highly practical scan chain based techniques that can benefit both the community of IC test and hardware security. Here is the list of our key contributions:

(1) The scan chain based fingerprint we proposed in this dissertation is by far the most cost-effective post-silicon digital fingerprinting approach, which has high practical value.

(2) The scan chain based fingerprint we proposed provides two levels of detection mechanisms to verify the embedded fingerprint. The non-destructive method is the most convenient yet secure among all the known approaches.

(3) The RSN based device identification and authentication protocol are both the first of their kind. The fact that they are compatible with industry standards makes them convenient to be adopted by IoT and embedded system designs.

(4) The public-private partial scan chain design is a breakthrough in secure scan chain design. The separation of public partial scan chain and private partial scan chains ideally can completely prevent information leakage to the attackers and thus making scan chain side channel attacks impossible. We have also provided a set of techniques that can provide provable full test coverage with the minimal cost. The FSM based approach to set values in the flip-flops in the private partial scan chains is novel. It combines both scalability and security, which allows it to be applied to real life designs and does not bring new vulnerabilities to the scan chain design, respectively.

1.5 Structure of the Dissertation

In Chapter 2, we provide the necessary background of scan chain. After introducing the basic concepts of scan chain, we focus on the following three features of scan chain that are directly relevant to this dissertation: the Q-SD and Q’-SD connection styles, partial scan chains, and the reconfigurable scan networks (RSN).

In Chapter 3, we present our scan chain based IP fingerprinting technique that utilizes the Q-SD and Q’-SD connection styles. We will elaborate the following problems: where to insert the digital fingerprint, how to detect the embedded fingerprint, and the security analysis of the approach against possible attacks. Simulations results on the design overhead incurred by embedding such fingerprint will also be reported.

In Chapter 4, we propose an RSN based IC identification scheme for embedded and IoT devices. This work shows how we can embed device IDs into designs following the industrial design standards. Using public benchmark circuits, we demonstrate that this approach can accommodate a large amount of distinct IDs.

In Chapter 5, we lay out the framework of our public-private partial scan chain design to prevent scan chain side channel attacks. We give a detailed motivation and rationale of our approach. We point out the key challenges in this approach, namely how to generate input test vectors, how to securely verify output test vectors, and how to provide the in-field test capability.

In Chapter 6, we address the challenge of generating input test vectors. We propose a hidden-test-vector graph and develop a set of test vector reordering and reusing, as well as an FSM based approach to ensure that our method will be able to test all the original test vectors with no need to regenerate any test vectors, and thus provide the full test coverage.

In Chapter 7, we report two approaches to obfuscate the output test vector in order to provide security to the private partial scan chains. One approach uses LFSR and the other uses configurable PUF. We also propose an in-field test protocol that enables test engineers to apply a new input test vector to the public-private partial scan chains and to observe the output test vector to detect whether there is any error.

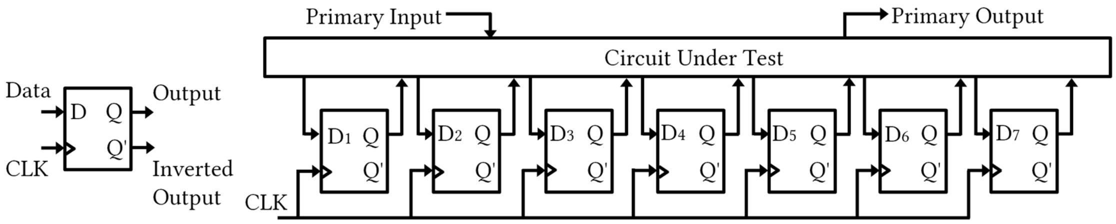
Chapter 8 concludes the dissertation with a revisit of scan chain's role in hardware security, our contributions and some of the future research directions.

Chapter 2: Preliminary on Scan Chain

2.1 Full Scan Chain

To improve testability of sequential circuits, scan chain, as a cost-effective technique, has been one of the most popular testing methods. The motivation of scan chain insertion is to convert sequential elements, which are flip-flops in this case, into accessible units, such that the internal states of the design can be easily controlled and observed via external pins. In a full scan design, all the flip-flops are included in the scan chain, making the CUT fully combinational. This makes ATPG particularly simple as no sequential test pattern generation is required. Using the much simpler combinational ATPG would be sufficient to obtain tests for all testable faults in the combinational logic. In addition, it is worth noting that only clocked D flip-flops can be used as state variables in the designs with scan chains.

In the IC design and fabrication process, the insertion of scan chain occurs after the design is verified to be functionally correct. The scan insertion procedure involves adding extra logic and signals dedicated for testing. Each conventional D flip-flop (Figure 1(a)) is replaced by a scan flip-flop, also called scan cell (Figure 2(a)), which introduces two extra input signals, scan mode input SD and scan enable SE, as well as one extra output signal, scan mode output Q_2. These scan cells are then connected in the form of a shift register, known as scan chain (Figure 2(b)), by connecting the scan mode output port Q_2 of one scan cell to the scan mode input SD of another scan cell. The SE signal is used to switch operating modes between normal mode and test mode.

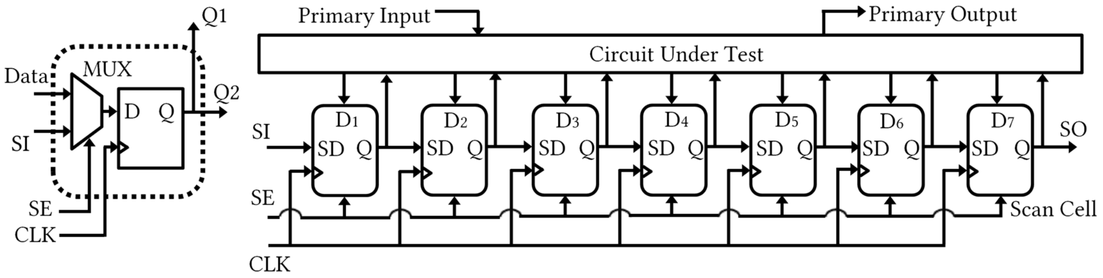


(a) A D flip-flop. (b) A circuit with 7 D flip-flops.

Figure 1: An example of a conceptual circuit with D flip-flops storing state variables.

Figure 1(a) is a traditional D flip-flop controlled by the clock signal CLK. 1-bit information D will be stored for a clock cycle and can be accessed through the Q and Q’ output ports. The value of D is then overwritten by the data input at the next clock signal. The characteristic equation of a D flip-flop is written as $Q_{next} = D$, $Q'_{next} = D'$.

Figure 1(b) is a conceptual circuit with 7 D flip-flops and the circuit under test (CUT) consists of only combinational logic gates. The CUT takes “Primary Input” and the contents of the 7 D flip-flops as input and produce a “Primary Output” as well as a new value for each of the D flip-flops. This implies that even with the same primary input value, the CUT may generate different primary output if the D flip-flops have different contents, which makes circuit test a very challenging problem since there is no input port to directly set the value for each flip-flop. A set of such values is called an input test vector and modern chips have thousands of flip-flops and require tens of thousands test vectors to ensure that the chip functions without defect and failure.



(a) A single scan cell. (b) The circuit after scan chain is built.

Figure 2: A single scan cell based on the traditional D flip-flop is displayed in (a). The circuit from Figure 1(b) after the scan chain is built is shown in (b). SI = scan input, SE = scan enable, CLK = clock, SO = scan output.

Scan chain is then inserted during the design time to tackle this issue. Figure 2 (a) shows a scan cell built on top of the D flip-flop. Instead of getting value from data input directly, the D flip-flop will take the value from a multiplexer (MUX) that is controlled by the SE signal. When SE = 0, the MUX outputs the “Data” value and the scan cell behaves the same way as a D flip-flop. However, when SE = 1, the MUX outputs the value from the scan input (SI) port which, provides great controllability and observability to IC test engineers. This relationship can be described as D = SE’ · Data + SE · SI.

Therefore, a scan-based design can operate in two modes, governed by the SE signal. When this signal is disabled, scan cells are connected to the circuit to behave as functionally expected, which is referred to as functional mode. With the signal enabled, the design switches to test mode when scan cells structurally form a shift register. Figure 2 (b) depicts the concept of scan chain. Note that each D flip-flop inside the scan cell still keeps its connection with the CUT. However, all the scan cells are connected together as a chain. The scan mode output Q_2 of each scan cell (except the last one) acts as the SI for the next scan cell. This change significantly improves testability by facilitating a cost-effective way to insert the desired input test vector to the flip-flops and to shift out for comparison. We elaborate this with the following example: suppose we want to test that when the input test vector is 1011000 for the 7 flip-flops from left to right, after the CUT executes with certain given primary input, whether the system reaches a state where the 7 flip-flops have 1110101 as their contents (known as output test vector). For the design without scan chain in Figure 1 (b), it is extremely hard to set the system at the state 1011000 and also challenging to verify that the system moves to state 1110101. However, this test becomes trivial with the help of scan chain. We first enable the SE signal SE = 1 and input 0, 0, 0, 1, 1, 0, 1 one by one from the SI port in 7 consecutive clock cycles. The first 0 will be shifted 7 times to the rightmost flip-flop D_7, the second 0 will be shifted 6 times to flip-flop D_6, and so on. As a result, the 7 flip-flops will be set to 1, 0, 1, 1, 0, 0, and 0, respectively, exactly the input test vector that is required. Then we set SE = 0 to switch the design back to normal mode and execute CUT. To verify the system state, we simply shift out the contents of the 7 scan cells through SO port and check whether they are 1110101 as expected. It is worth mentioning that the next input test vector can be simultaneously placed into the scan cells via SI port while we observe their current contents through SO port.

In summary, scan testing involves three stages. First, an input test vector is shifted in through SI port to set the circuit in a specific state after configuring the circuit in the test mode (SE = 1). Then the circuit is switched to the functional mode (SE = 0) so the CUT can run one or multiple clock cycles starting from the given system state with an input from the primary input port. The output can be observed from the primary output and system's updated state will be stored in the scan cells. Finally, the circuit will be switched back to the test mode (SE = 1) to allow the updated system state information to be shifted from the scan cells to the SO port. One can compare this information with the given fault-free response to check whether there is any defect or fault. The input vectors and their corresponding fault-free responses are referred to as test vectors.

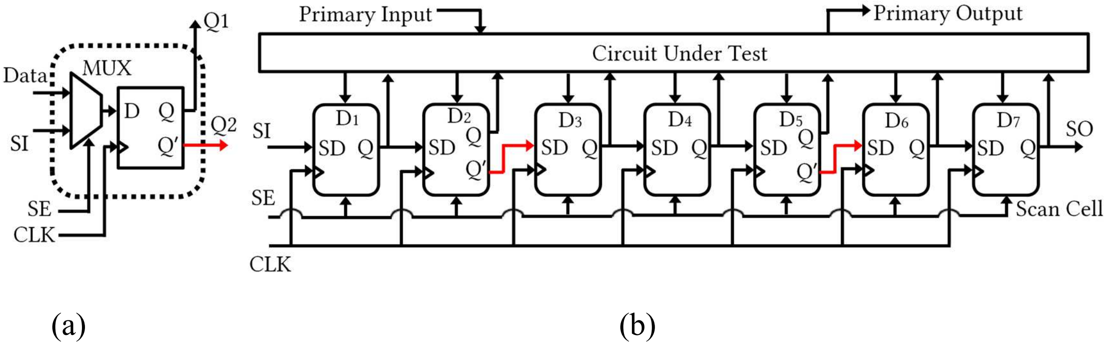


Figure 3: A single scan cell that output port Q' is used to connect the next scan cell is displayed in (a). The circuit of which the connection styles between (D_2, D_3) and (D_5, D_6) are changed to Q'-SD is shown in (b).

In fact, the connection style between adjacent scan cells is not restricted to Q-SD. Q'-SD connection style has been applied in the scan design to optimize test power consumption as well [45]. As shown in Figure 3(b), instead of connecting all the scan cells purely in Q-SD connection style (Figure 2(b)), the Q' port of D flip-flop can also be used for connection, where the second and fifth connection styles have been changed to Q'-SD. Recall that for D flip-flop, $Q'_{next} = D'$. Therefore, any bit of the test vectors going through a Q'-SD connection will be flipped. To test the CUT in Figure 3(b) with correct states, we need to adjust test vectors according to the positions of Q'-SD connection styles in the scan chain. Using the same example for Figure 2(b) where the scan chain has only Q-SD connection styles, to test the CUT with state 1011000, we need to adjust the input test vector to 1000100 to correctly set 1011000 in these 7 flip-flops. As the first two bits "10" in the original test vector don't go through any Q'-SD connection in the design of Figure 3(b), they will stay the same for the adjusted input test vector. The third to the fifth bits "110" go through the Q'-SD connection in the second position and are flipped once, thus need to be adjusted to "001". The last two bits "00" go through both Q'-SD connections in the design and are flipped twice, so they don't need to change.

2.2 Partial Scan Chain

In spite of all the advantages a full scan chain has for testing purpose, it introduces hardware area overhead and performance penalties in critical paths. Meanwhile, test vectors for full scan chains are lengthy. As an alternative solution, partial scan was proposed a few decades ago to provide a trade-off between the ease of testing and the cost associated with scan design. In a partial scan chain, only a subset of D flip-flops is converted to scan flip-flops and included in the scan path. However, it isn't widely adopted due to the low fault coverage in that the un-scanned flip-flops cannot be directly controlled and observed. Thus, sequential ATPG is required to generate test vectors, which has much higher complexity and remains a very expensive computational process. It has poor initializability, as well as poor controllability/observability of state variables. Cycles in the circuit are believed to be mainly responsible for the complexity. The test generation time is normally two orders of magnitude higher than that of a full scan chain. The scan test sequences are extremely long as well.

Figure 4 illustrates the design with a partial scan chain based on the same circuit with a full scan chain in Figure 2(b). We can see that two flip-flops, D_2 and D_5, remain as normal D flip-flops. They are not converted to scan cells like others and are not chained. As a result, no one can control or observe the contents of these two flip-flops through the scan chain. The partial scan includes 5 flip-flops, D_1, D_3, D_4, D_6, and D_7 and connects them into a single chain. In particular, scanned flip-flops and un-scanned flip-flops are controlled from separate clock primary inputs. CLK1 is the scan clock, which is only active in the scan mode. CLK2 is the normal clock. Both CLK1 and CLK2 need to be active in the normal mode.

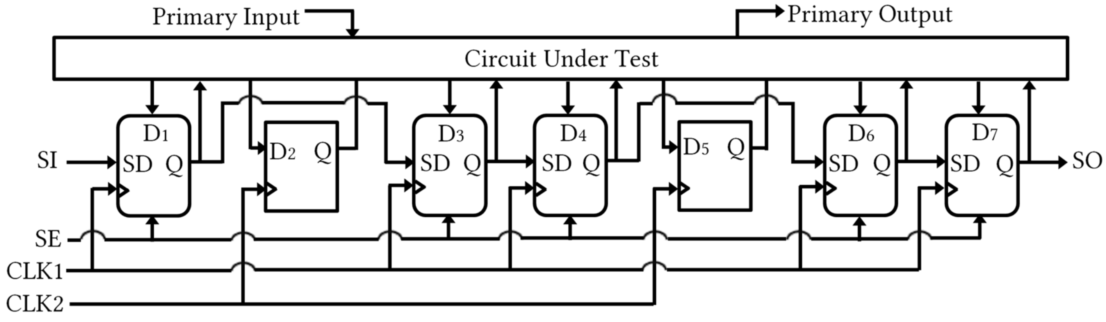


Figure 4: A partial scan chain where the second and fifth flip-flops are removed from the scan chain.

Partial scan aims at maximizing the fault coverage while minimizing area and performance overhead. This makes how to select flip-flops to scan a prominent issue while designing partial scan chains. To approach this problem, previous work can be classified into 3 board categories: structural analysis based [76], testability analysis based [77] and ATPG based [78]. Structural-analysis based techniques represent the sequential circuit as a topology graph such that combinational logic and registers are separated into different components of the graph. Then they attempt to remove all possible feedback by scanning flip-flops. However, removal of a minimal vertex set is an NP-complete problem and moreover, the fault coverage of a sequential ATPG cannot be guaranteed even with all cycles (except self-loops) removed. On the other hand, the testability analysis based approaches are easier to adopt in terms of computational complexity but they usually do not yield good fault coverage for circuits with more complex structures. ATPG-based techniques seek to utilize the information generated by the test generator to try and detect the aborted faults. This type of approaches takes in a list of aborted states that the test generator was unable to justify and tries to make those states reachable by selecting the minimal set of flip-flops.

2.3 Reconfigurable Scan Networks

Scan chains are extensively used to reduce the test complexity. They eliminate the need for sequential test pattern generation by making internal memory elements directly controllable and observable. However, in the traditional design of scan chains, where all scan registers are chained into a single scan chain, the time overhead of accessing each module’s scan registers can be too high. To reduce this overhead, reconfigurable scan networks (RSNs) are introduced, which enable dynamic reconfiguration of scan networks and allow cost-efficient access to on-chip instrumentations.

Figure 5 is an example of an RSN compliant with IEEE1687-2014. Scan data are shifted in via the SI port, through a fraction of scan registers, called scan segments, to the SO port. The bits *a* and *b* of Segment 1 are used to configure the active scan path in this case. Meanwhile, Segment 2 and 3 could be an interface to interact with on-chip instrumentations. Clearly, the time to access a scan segment in an RSN is proportional to the length of the active scan path, which can be significantly reduced by choosing access modes such that irrelevant segments are bypassed in the active scan path. For example, in Figure 5, given the test pattern for Segment 2, we need to first generate an access pattern which set *ab = 01* or *ab = 10* to ensure that Segment 2 is currently included in the active scan path, and then apply the test pattern to Segment 2 to facilitate testing. On the other hand, different active scan paths can also be configured in this design. Segment 2 can be bypassed if *ab = 11*. Both Segment 2 and 3 will be

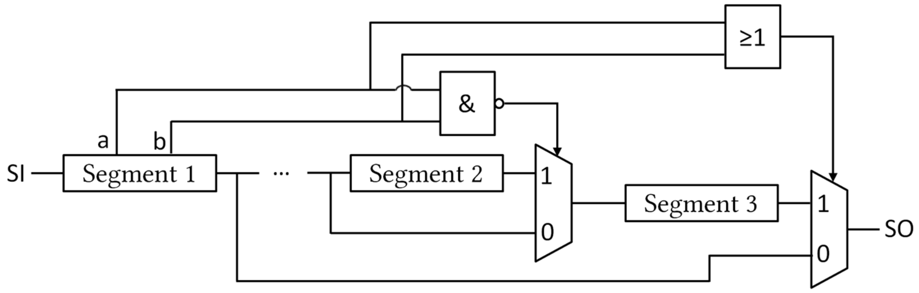


Figure 5: An example of a reconfigurable scan network. [79]

bypassed if *ab = 00*. Thus, RSNs offer high flexibility on the configuration of the scan path in order to reduce access time.

In the following, we review the definition of reconfigurable scan networks presented in [35], which covers the existing RSN standards, IEEE 1149.1-2013 [36] and IEEE P1687 (IJTAG).

An RSN has four data ports namely *scan-input*, *scan-output*, *reset input*, *clock input* as well as three control ports, *capture, shift and update* which are controlled by a 1149.1-compliant TAP [36]. RSNs are composed of *scan segments*, multiplexers or other combinational logic blocks. The scan segment consists of scan registers which are accessible through the scan-in and scan-out ports, and an optional shadow register. The block diagram of a scan segment is presented in Figure 6. The state of the shadow register determines the configuration of scan networks. Scan segments provide access to testing structures and enable distributed control over the on-chip instrumentations. Each scan segment should support three modes of operations, namely *shift*, *capture*

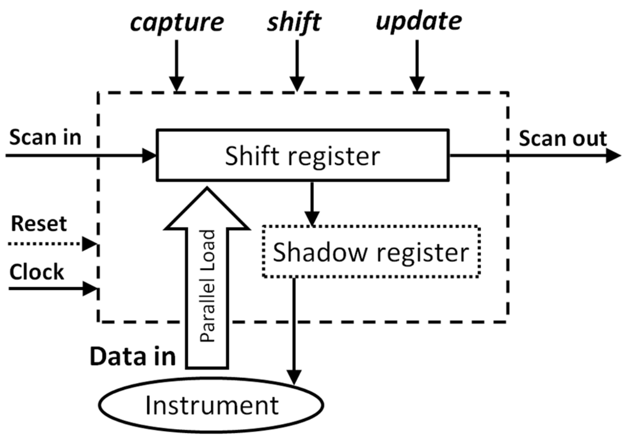


Figure 6: The block diagram of a scan segment.

and *update*, which are controlled by external control signals.

In the capture mode, the scan registers get overwritten by the data coming from the corresponding instrument (Data-in port). During a shift operation, the data from scan-in port is shifted through the scan registers to scan-out port. In the update mode, the data in scan registers is written to the optional shadow register, which determines the state of the scan segment. Scan segment might have another control port called *select* which determines whether the scan segment can perform capture, shift and update operations.

Scan segments are connected either by buffers or *Scan Multiplexers.* The latter selects the path that scan data goes through in the network and its select signal is referred to as *address* in the scan network literature, as shown in Figure 5. The internal control signals of scan segments such as *select*, and the *addresses* of scan multiplexers are determined by the output of combinational logic blocks, whose inputs are controlled by the value of shadow registers of scan segments and the primary data and control inputs of the RSN. A scan path is *active* if all the scan segments on the path are selected, and the *addresses* of all on-path scan multiplexers are set appropriately. To access a scan segment in the RSN, you need to put this segment on an active path. A read or write access to a scan segment, as defined by IEEE 1149.1 [36], is a three-step process called a *CSU* (Capture-Shift-Update) operation: in capture mode of a *CSU*, all the scan registers on the active scan path load the test results from their corresponding instrumentation. Then, this data will be shifted out during the following shift operation. Note that during shift operation, the new scan data will be shifted in as the data in the scan registers are being shifted out.

Finally, in the update mode of a CSU, the contents of scan registers in the active path get loaded to the corresponding shadow registers to reconfigure the scan path.

2.3.1 Segment Insertion Bit Based RSN

A segment Insertion Bit (SIB) is a hardware component proposed by IEEE P1687 [37] which can be used to reconfigure scan networks by bypassing or including scan chains in scan paths. It is a simple and flexible architecture that allows hierarchical control over the accessibility of individual instruments. A SIB is in principle a configurable bypass. As shown in Figure 7, it either bypasses a subordinate instrument or sub-network connected to its TO/FORM ports, or connects it to the higher level scan chain. The mode of operation is chosen by shifting a single configuration bit into the SIBs SI port.

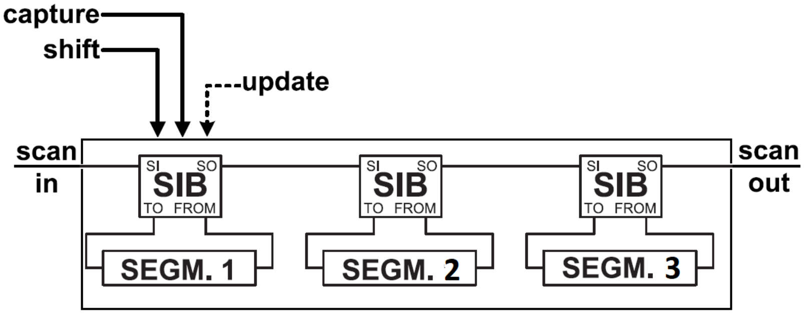


Figure 7: An example of SIB based RSN.

A possible implementation of the SIB proposed in [38] is presented in Figure 8. An SIB has a scan-input and a scan-output as well as four control inputs, *capture*, *shift*, *update* and *select*. It contains a 1-bit shift register *S* and a 1-bit shadow register

U. Note that the same set of external control signals drive scan segments and SIBs in a scan network. During the shift operation, based on the value of the shadow register *U* and the *select* signal, the data from the scan-in port either gets directed to the lower level scan segment of the SIB, referred to as Directing mode, or bypasses the lower level scan segment and directly goes to the scan-out port, referred to as Bypassing mode. The value of the shadow register *U* only gets updated from *S* if both *update* and *select* signals are activated. The capture operation is the same as scan segments.

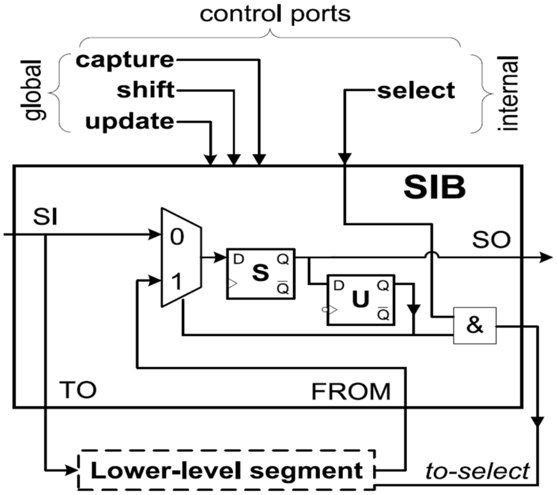


Figure 8: Implementation of a segment insertion bit [38]. *S* is a shift register and *U* is a shadow register.

Chapter 3: Scan Chain based IP Fingerprinting

3.1 The Need of IP Fingerprint

Fingerprints are the characteristic of an object that is unique and incontrovertible so they can be used to identify a particular object from its peers. Fingerprints have been used for human identification for ages and also been adopted in multimedia for copyright protection of the widely distributed digital data. In the semiconductor and integrated circuit (IC) industry, the concept of digital fingerprinting was introduced in the late 1990's with the goal of protecting design intellectual property (IP) from being misused [16, 17, 39]. With the promise of giving each copy of the IP, and hence the IC that uses the IP, a unique fingerprint, digital fingerprinting has become a hardware security primitive and enabling technique for applications such as IP metering, identifying IP piracy, detecting IC counterfeiting and overbuilding.

These early works demonstrated the feasibility of creating large amount of functional identical IPs with distinct implementations [16, 17, 39]. However, the proposed techniques are impractical because they create fingerprints in early IP design stages, making all the fingerprinted IPs require different masks for fabrication. Several practical fingerprint methods have been proposed recently [40, 41], where the authors modify the gate level layout of the design based on don't care conditions to add implementation flexibilities and create fingerprints during the post-silicon testing phase based on such implementation flexibility. But they suffer from moderate or high design overhead (in terms of area and delay), are not easy to be detected, and may be vulnerable to various attacks such as fingerprint removal and forgery.

Meanwhile, IC identification methods based on glitches or path delay variations, and the well-studied physical unclonable function (PUF) have also been reported [42, 43]. They rely on the uncontrollable fabrication variations to identify and authenticate ICs. However, when an IC is illegally reproduced or overbuilt, the illegal copies will have variations different from the original genuine copy, so they cannot be used for the protection of IC and IPs inside.

IP owners have to protect themselves as well as legal users from IP theft. Digital fingerprinting meets this requirement too. The uniqueness of fingerprint enables the trace of each copy of the IP, including those illegally resold IPs. Therefore, the IP owner can identify the dishonest user or the victim of the IP thefts. Not only is the origin of the design identified, but also the origin of the misappropriation can be tracked down.

3.2 An Illustrative Example

In this chapter, we propose a practical and low overhead scan chain based digital fingerprinting method that can be primarily used in the following two scenarios: (1) ***conveniently verify and trace the use of IP*** when the fingerprints are not tampered; (2) ***protect IP and the IC that uses the IP*** by detecting any attempts to remove or modify the fingerprinted scan chain.

Figure 9 depicts a 5-stage scan chain where the five scan cells (scan flip-flops, or SFFs) are labeled as D_1 through D_5 from left to right. It gives testing engineer the ability to put the core under test (CUT) in any desired state (represented as the values of the SFFs) by inputting the values, called test vectors, through the scan in (SI) port; then observe how the core behaves through the scan out (SO) port. Assume that in this case, we have two test vectors X_1=00000 and X_2=01001. The corresponding responses (or next states) are Y_1=00000 and Y_2=10110.

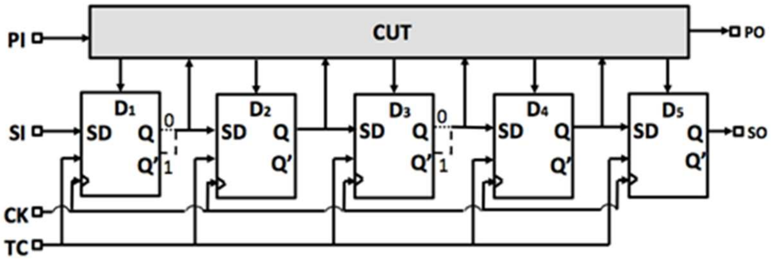


Figure 9: An example of a 5-stage scan chain where the connections between (D1, D2) and (D3, D4) are selected to embed fingerprints. A 2-bit fingerprint can be created by the way (Q-SD or Q'-SD) these two pairs of flip-flops are connected.

Our fingerprinting approach takes advantage of the fact that *scan cells can be chained by either the Q-SD or the Q'-SD connection style* [45, 46]. Suppose that we have identified two pairs of SFFs, (D_2, D_3) and (D_4, D_5), as the locations to embed the fingerprint. We use the Q-SD connection to embed a bit '0' and the Q'-SD connection as a bit '1' (see Figure 9). This will allow us to embed any 2-bit fingerprint, "00", "01", "10", or "11", by selecting different connection styles.

Table 1: The test vectors and their corresponding output responses for all possible 2-bit fingerprints

f_1f_2	X_1	Y_1	X_2	Y_2
00	01100	01111	00011	10110
01	01111	10011	00000	01010
10	00011	11111	01100	00110
11	00000	00011	01111	11010

Suppose the original design uses the Q-SD connection on both locations, that is, it carries the fingerprint “00”. To embed fingerprint “01”, for example, we will connect the Q’ port of D_4 to the SD port of D_5. As a result, when data moves from D_4 to D_5, its value will be flipped. Therefore, we have to change the two test vectors to X_1=00001 and X_2=01000 to ensure that the CUT is tested with states 00000 and 01001, respectively. Similarly, the output responses Y_1 and Y_2 will change in a similar fashion. Table 1 lists the two test vectors and their corresponding output responses for all the four possible fingerprinted designs.

To identify each copy of the design, we can simply check the test vector. If the test vector or its output response is different from Table 1, then the design is not genuine.

3.4 Scan Chain based Fingerprint: Idea, Advantages and Limitations

In our scan chain based fingerprinting method, we first utilize the Q-SD and Q'-SD connection styles, which are both available and used in scan design, to create digital fingerprints at the circuit level; then we modify the set of test vectors accordingly to keep the fault coverage. Therefore, the fingerprint is embedded both inside the scan chain in the form of Q-SD or Q'-SD connections and as the values of test vectors. This enables us two ways to detect the fingerprint: a destructive method that requires reverse engineering the chip to reveal the connection style of the scan chain; and a non-destructive method where the verifier only needs to check the test vectors. Compared to the existing fingerprinting methods, our approach has the following advantages:

1. **Practical** – The fingerprint locations in the scan chain can be selected before fabrication and we choose either Q-SD or Q'-SD connection style at post-silicon stage. So all fingerprinted designs can be fabricated with the same mask. The modification to the test vectors does not need any changes on the chip.

2. **Ultra-low overhead** – Fingerprints are added in the scan chain. This will not cause any performance overhead to the core design. We further demonstrate that there is little overhead to the scan chain design while maintaining the full test coverage.

3. **Non-destructive verification** – Fingerprints can be detected and verified from the scan input and output without physically opening up the chip.

Due to the simplicity and convenience of creating and verifying the fingerprints, it might be easy for the attacker to remove, modify, or forge a fingerprint. Hence, the challenge is how to implement this idea such that it will (1) be resilient to potential attacks, (2) maintain the property of easy fingerprint detection, but (3) not lose fault coverage, and (4) not cause significant overhead in testing time and power. Our proposed solution is based on the careful selection of fingerprint locations and data integrity techniques. It requires both Q-SD and Q'-SD connections to be available, which can be implemented with fuses or configurable logic [40, 41], and controllable at post-silicon stage, for example, by blowing the fuses or configuring the logic. We conduct a comprehensive security analysis which shows that our solution is robust unless attackers re-design the scan chain. Experimental results validate that it causes little or no overhead in terms of test time and power.

3.4 Current Digital Fingerprinting Schemes

A digital fingerprinting technique embeds unique features, known as fingerprint, in each copy of the IP to allow IP owners to trace each copy of the IP. This was first reported by Lach et al. in [39], where the authors used an FPGA design partitioning and tiling technique to embed distinct fingerprints in the originally watermarked design. However, this technique is relatively impractical since the problem must have a specific structure. Then Caldwell et al. [16] proposed a generic methodology to embed fingerprints in the solutions to optimization problems. The key idea is to apply iterative optimization in an incremental fashion to encode distinct fingerprints. Qu and Potkonjak [17] introduced a different method based on adding special constraints to the design specification such that multiple distinct copies of the design can be easily constructed from one seed design. Unfortunately, all these approaches create fingerprints in the earlier stage of the VLSI design cycle, inevitably increasing the design cost and therefore are impractical.

In [40], a satisfiability don't-care (SDC) condition based circuit fingerprinting technique is developed to create fingerprints at the post-silicon stage by using MUXs to replace certain library cells. SDC conditions describe the signal combinations that cannot occur, which give rise to the situation that two distinct circuits might have exactly the same truth table since the input combination producing different outputs does not occur. By locating gates that have SDCs leading into them, referred to as fingerprint locations, and finding alternative gates, different fingerprinted copies can be generated by using either the original gate or one of its alternatives at each fingerprint location. In [41], the authors proposed to utilize observability don't-care (ODC) conditions and add extra wires without changing the design's functionality. An ODC condition occurs when local signal changes cannot be observed at a primary output. Thus, two circuits can implement exactly the same function although they are physically different. This feature allows them to create a 1-bit fingerprint by fabricating the circuit with the flexibility that whether or not applying the local change does not affect functionality. In both methods, the design will be modified such that fingerprints, in the form of different layouts such as library cells and wires, can be generated at the post-silicon stage. While these methods are practical, they incur large design overhead in circuit area and delay.

3.4 Scan Chain based Fingerprinting Technique

We utilize the Q-SD and Q'-SD connection styles between SFFs to create the fingerprint for a design in the following steps [44]:

Step 1. Perform the normal scan design to obtain the best possible solution. This normally includes determining (1) a set of test vectors to achieve the best test coverage; (2) the order of the scan chain, that is, which SFF will be the next for a given SFF; (3) the connection style between each two SFFs.

Step 2. Identify the fingerprint locations. By deliberately choosing whether two adjacent flip-flops have a Q-SD or a Q'-SD connection, we can create a bit of information for the fingerprint. If the design has n flip-flops in its scan chain, we can embed any of the 2n possible n-bit fingerprints. Therefore, the challenging question is: how to select k pairs of SFFs as fingerprint locations to minimize the performance overhead in the fingerprinted copies.

Step 3. Develop fingerprint embedding protocols. This can be as simple as the one in the illustrative example where 0 and 1 are embedded as Q-SD and Q'-SD connection styles respectively. But a good fingerprint embedding protocol should balance (1) low design cost, (2) low or no performance degradation, (3) easy detectability, and (4) high robustness and resilience.

Step 4. Modify the set of test vectors. While fingerprints are in the forms of Q-SD or Q'-SD connection styles, we want to maintain the test vectors' fault coverage. Therefore, the set of test vectors have to be updated based on the fingerprint embedded in the design, as shown in the illustrative example in Section 3.2.

In the rest of this section, we elaborate the last three steps and how to verify the fingerprints

3.4.1 Identify Fingerprinting Locations

Besides fault coverage, a circuit's power consumption during test is another important concern for scan design. For a given set of test vectors, we can determine to use the Q-SD or Q'-SD style to connect two adjacent SFFs in order to minimize the switching activities, and hence power consumption [45]. At the i^th^ position (output of the i^th^ SFF) of the scan chain, let $Cost_Q^i$ be the total number of transitions when Q-SD connection style is used and $Cost_{Q'}^i$ be the total number of transitions when the connection style is Q'-SD, we have

$$Cost_Q^i = Input_{dif}^i \times i + Output_{dif}^i \times (n - i)$$

$$Cost_{Q'}^i = Input_{same}^i \times i + Output_{same}^i \times (n - i)$$

where n is the size of the scan chain, $Input_{dif}^i$ and $Output_{dif}^i$ are variables for number of times that consecutive bits differ ("01" or "10") at position i for all the test vectors and their output responses, respectively. $Input_{same}^i$ and $Output_{same}^i$ are similarly defined when consecutive bits are the same ("00" or "11"). The optimal connection style type for each position is determined by comparing the values of $Cost_Q^i$ and $Cost_{Q'}^i$: Q-SD is selected if $Cost_Q^i < Cost_{Q'}^i$, Q'-SD is selected if $Cost_Q^i > Cost_{Q'}^i$.

Once a fingerprinting location is chosen, its connection style will be decided by the value of the fingerprint bit, which will be random. Therefore we choose the positions with smaller $|Cost_Q^i - Cost_{Q'}^i|$ to keep the power overhead caused by distinct fingerprints at the minimal level.

3.4.2 Fingerprint Creation

After we choose k fingerprint locations, we can embed any k-bit fingerprint. However, as we will show in the security analysis, attackers may also reveal the fingerprint locations and make changes on the SFF connection styles or on the test vectors to alter or remove the fingerprint. To counter this attack, we propose to create the fingerprint bits by the following data integrity technique: (1) start with an m-bit fingerprint with $m \leq k$, (2) use this m-bit fingerprint as the seed to generate m' bits by a one-way hash function, (3) use the (m + m') bits as the fingerprint to guide the selection of Q-SD or Q'-SD connection styles at the selected fingerprint locations. This completes the fingerprinted scan design and we will discuss its security in the next section.

Next, we need to modify test vectors for each fingerprinted scan design to keep the fault coverage. For this, we have

<u>Lemma</u>. Let $x_1x_2\ldots x_n$ be an input test vector for the original scan design; $y_1y_2\ldots y_n$ be the corresponding output test vector. If a fingerprinted scan has altered the connection styles at SFF positions p_1, p_2, .., p_k, then the following input test vector $z_1z_2\ldots z_n$ will provide the same fault coverage on this fingerprinted scan as $x_1x_2\ldots x_n$ does in the original scan:

$z_i = x_i$ if $i \in [1, p_1] \cup [p_2,p_3] \cup [p_4,p_5] \cup \ldots$

$z_i = x'_i$ otherwise

The corresponding output test vector $w_1w_2\ldots w_n$ on this fingerprinted scan is given by:

$w_i = y_i$ if $i \in [p_k, n] \cup [p_{k-2},p_{k-1}] \cup [p_{k-4},p_{k-3}] \cup \ldots$

$w_i = y'_i$ otherwise

We omit the proof of this Lemma due to simplicity. One can easily verify this for the example in Table 1.

3.4.3 Fingerprint Detection

There are two ways to detect the fingerprint bits embedded by the above method. First, if the design is not altered, we can simply check the input/output pair of test vector on the fingerprinted design and compare it with the original design to determine the fingerprint bits. The data integrity technique enables us to further verify whether the fingerprint bits have been modified. To do this, we first recover the first m bits, then re-generate the hash bits and compare with the remaining m' bits. If we find a mismatch or other evidence that the design might have been altered, we can open up the chip and check the connection styles to confirm it, which is the second method to detect the embedded fingerprint.

3.5 Security Analysis

In this section, we first briefly explain why both IP owners and users need to be protected and how our proposed scan chain based fingerprinting technique achieves such protection. Then we conduct a security analysis of our fingerprinting method on the potential attacks and the corresponding countermeasures.

Both the IP owners and legal IP customers need to protect their rights. IP designers want assurance that their designs will not be illegally redistributed by customers and customers also want to ensure what they bought is legitimate and should any IP piracy occur, they will not be the victim. For example, in case the design is resold without the owner’s approval, the involved users should be tracked and punished, not other IP users.

Our approach can easily be implemented by local rewiring to determine a specific connection style of certain pairs of scan cells. Since such changes are local and do not require scan chain reordering or rerouting (as needed for existing approaches), they will not cause high design overhead. However, we need to modify test vectors because different fingerprinting configurations of the Q-SD and Q’-SD connections will change the flip-flop contents for testing. As we have discussed earlier, this can be easily accomplished by the Lemma in section 3.4.2. Meanwhile, this gives us an alternative way to detect fingerprints: instead of physically de-packaging the chip and verifying the connection styles, we can conveniently determine the fingerprint by checking the test vectors.

A naïve attack on the proposed approach would be to send out ones and zeros through the scan chain, trying to figure out the connection styles to reveal fingerprints. However, it is infeasible to learn the connection styles by merely observing the scan out results. For example, for a 5-stage scan chain, if the attacker scans in “00000” and then gets “00000” from the SO port, the possible connection style might be “00000”, “00011”, “01111” et al. as long as the number of different bits between each style is even. It would be unlikely to determine which way the scan chain is truly implemented if the scan chain is long enough. Thus this attack does not work. Next we analyze various possible attacks and discuss briefly how the proposed approach is resilient to these attacks or how corresponding countermeasures can be added.

3.5.1 Fingerprint Denial

The adversary may conceive to simply deny that the fingerprint information has been inserted. Instead, he may declare that the fingerprint is merely a coincidence without changing the functionality or performance of the IC. We could defeat this attack by proving that the probability of such coincidence is sufficiently low. In the proposed scheme, it is reasonable to assume that the connection style of each position in scan chain can be two alternative connection styles with the same probability. Thus, the probability that a non-fingerprinted design carries a specific m-bit fingerprint will be $1/2^m$. Obviously, a longer fingerprint indicates a stronger authorship proof. On the other hand, since the fingerprinted design would incur more or less power overhead compared to the optimized design [45], it will make no sense for a designer to choose a specific connection style which conflicts with that determined by optimization criteria if it is not specially designed for fingerprinting.

3.5.2 Fingerprint Modification

In fingerprint modification attacks, apparently the adversary will attempt to modify the fingerprints. Our analysis is based on the two different detection methods in section 3.4.3.

1) Detecting by checking test vectors

If we rely on checking the test vectors to identify the embedded digital fingerprint, an adversary can arbitrarily change test vectors (and hence destroy the fingerprint) to make it difficult or impossible to identify the fingerprint in this way. It is normally undesired to perpetrate such attack because adjusting only test vectors will lead to lower fault coverage. Without proper test coverage, a circuit may malfunction and the end user may not know. Furthermore, it becomes meaningless when we view the layout in the chip to check fingerprint locations.

2) Detecting by opening up chips

If it is possible for us to open up the chip to verify fingerprints, it will be reasonable to assume that attackers will have the same capability and they can modify the interior structure of the scan chain. In this case, the attacker can randomly change the connection styles between SFFs, the same way we embed a fingerprint, to destroy the fingerprint.

The data integrity technique proposed in section 3.4.2 can be used to defeat these attacks. By making fingerprint bits dependent, we not only allow the fingerprint to be verified, but also make it difficult for the attacker to forge. A successful forgery requires both the m-bit seed and the m'-bit hash to be changed consistently. Although it is possible to change the connection styles between SFFs, it will be challenging to make the change which can maintain the property that the m' bits are the hash of the first m bits.

Another powerful attack to any fingerprinting method is the so-called collusion attack. In such an attack, the attacker has multiple copies of the ICs with different fingerprints. He can compare the SFF connection styles of these ICs and find the differences of their connection styles to reveal some fingerprinting locations. Once such locations are identified, an illegal fingerprint can be forged. Similarly, the data integrity approach can detect attacks where connection styles are arbitrarily changed. Therefore, such copies will not be considered authentic.

3.5.3 Fingerprint Removal

This attack refers to the case when an attacker can retrieve the original design without any fingerprint. This will be impossible when an attacker can only access the input/output test vector pairs. The best an attacker can do in that case are those listed in Section 3.5.1 and 3.5.2. However, when an attacker is capable of opening up the chip, he can certainly make all the connections to be Q-SD to remove the fingerprint. We mention that this might be practical for IP blocks of small size, but for very large scale ICs, the efforts of de-packaging the chip and reverse engineering to obtain the netlist will be very expensive, making it unlikely for them to remove the fingerprinted scan chain and redesign the circuit without any fingerprint.

3.6 Experimental Results

In the experiment, we used the Design Compiler under Synopsys to synthesize and obtain netlists from the designs from ISCAS89, ISCAS99 and LGSynth93 benchmark suites. Detailed information about the benchmark circuit is given in Table 2. The second column denotes the number of flip-flops in the circuit, in other words, the length of the scan chain. 'T*org*' in the third column indicates the number of transitions during testing by the originally optimized scan design, respectively. The DfT Compiler and TetraMax under Synopsys are, respectively, used to create the original scan chain and generate the test patterns. The 10-bit and 128-bit fingerprints are, respectively, embedded into the experimental designs. All experiments were run on a 3GHz HP Z620 work station with Linux operating system and 12 GB of memory.

Table 2: Information of benchmark circuits from ISCAS89, ISCAS99 and LGSynth93

Circuit	Number of flip-flops	T_{org}
S38584	1166	3.31E+08
S38417	1564	1.13E+09
S35932	1728	6.72E+07
B17	1315	2.31E+09
B17_1	1316	2.31E+09
B18	2908	2.45E+10
B18_1	2904	2.30E+10
B19	5816	1.55E+11
B19_1	5709	1.44E+11
DMA	1831	2.21E+09
usb_funct	1517	1.30E+09
ac97_ctrl	1876	6.73E+08
pci_bridge32_1	1485	9.30E+08
pci_bridge32_2	1828	1.40E+09
des_perf	8808	1.32E+10
ethernet	10015	1.37E+11
vga_lcd	16904	1.59E+12
Average	**4041**	**1.23E+11**

Table 3 and 4 show the fingerprinting results on the ISCAS and LGSynth93 benchmark circuits using the proposed fingerprinting method. ΔT represents the percentage increments from T_{org} to T_{fp}, where T_{fp} denotes the number of transitions during testing by the fingerprinted scan design. The column, '*n*' under that of 'ΔT' denotes the maximum number of connections among N connections that can be altered by fingerprinting while maintaining the overhead on transitions smaller than ΔT**.** To evaluate the overhead due to multiple different fingerprints, we use the pseudo-random generator to generate 10 random numbers between [1...*n*] to index 10 connections among the *n* qualified connections. We then compute the average and worst case overhead of transitions caused by the 1024 different fingerprinted designs, which are implemented by different configurations of the 10 selected connections. The columns 'ΔAT' and 'ΔWT' denote the average overhead and worst case overhead of transitions respectively. Table 3 and 4 give the transition overheads introduced by 10-bit small size fingerprints. We can see that for a design, a smaller ΔT corresponds to a smaller *n*, which means a smaller pool of the qualified connections. To guarantee the overhead less than 0.1%, at least 70 qualified connections (in the design S35932) can be found. This can enable a sufficiently large pool of 2^{70} fingerprints. Also, the average overhead of 1024 different fingerprinted design can be controlled within 0.007%. The average of worst case overhead is no more than 0.014%.

Table 3: Average Overheads of Transitions due to 1024 different fingerprints on Benchmark Circuits, where ΔBT, ΔAT and ΔWT denote the best, average and worst case overhead respectively.

Circuit	ΔT= 1%			ΔT=0.5%		
	n	ΔAT(%)	ΔWT(%)	*n*	ΔAT(%)	ΔWT(%)
S38584	458	1.11E-02	2.21E-02	329	7.07E-03	1.41E-02
S38417	599	9.06E-03	1.81E-02	426	7.96E-03	1.59E-02
S35932	231	1.41E-02	2.81E-02	161	1.38E-02	2.75E-02
B17	366	1.69E-02	3.38E-02	266	1.11E-02	2.21E-02
B17_1	383	1.23E-02	2.45E-02	285	5.46E-03	1.09E-02
B18	1059	4.70E-03	9.41E-03	779	3.69E-03	7.37E-03
B18_1	1069	3.21E-03	6.41E-03	782	3.26E-02	6.52E-03
B19	2306	1.88E-03	3.75E-03	1770	1.30E-03	2.61E-03
B19_1	2263	2.57E-03	5.13E-03	1735	1.39E-03	2.78E-03
DMA	793	5.03E-03	1.01E-02	572	5.19E-03	1.04E-02
usb_funct	456	8.53E-03	1.71E-02	326	6.89E-03	1.38E-02
ac97_ctrl	599	9.13E-03	1.83E-02	426	7.73E-03	1.55E-02
pci_bridge32_1	661	9.40E-03	1.88E-02	474	5.72E-03	1.14E-02
pci_bridge32_2	786	6.51E-03	1.30E-02	563	4.26E-03	8.52E-03
des_perf	3575	1.29E-03	2.57E-03	2550	7.91E-04	1.58E-03
ethernet	2042	2.76E-03	5.52E-03	1609	1.45E-03	2.90E-03
vga_lcd	7056	5.30E-04	1.06E-03	5978	4.16E-04	8.32E-04
Average	**1453**	**7.00E-03**	**1.40E-02**	**1119**	**6.87E-03**	**1.03E-02**

Table 4: Average Overheads of Transitions due to 1024 different fingerprints on Benchmark Circuits, where ΔBT, ΔAT and ΔWT denote the best, average and worst case overhead respectively.

Circuit	ΔT=0.2%			ΔT=0.1%		
	n	ΔAT(%)	ΔWT(%)	*n*	ΔAT(%)	ΔWT(%)
S38584	207	5.46E-03	1.09E-02	148	4.00E-03	8.01E-03
S38417	276	3.22E-03	6.44E-03	199	2.21E-03	4.42E-03
S35932	101	1.26E-02	2.51E-02	70	8.76E-03	1.75E-02
B17	168	7.48E-03	1.50E-02	120	3.16E-03	6.32E-03
B17_1	185	4.35E-03	8.70E-03	135	3.91E-03	7.82E-03
B18	500	1.81E-03	3.62E-03	355	1.16E-03	2.31E-03
B18_1	500	1.81E-03	3.63E-03	353	1.40E-03	2.80E-03
B19	1156	7.39E-04	1.48E-03	826	7.30E-04	1.46E-03
B19_1	1133	6.85E-04	1.37E-03	803	6.88E-04	1.38E-03
DMA	373	2.08E-03	4.16E-03	269	1.65E-03	3.30E-03
usb_funct	205	4.65E-03	9.30E-03	142	2.93E-03	5.86E-03
ac97_ctrl	268	3.96E-03	7.93E-03	188	2.71E-03	5.41E-03
pci_bridge32_1	301	3.57E-03	7.13E-03	212	2.70E-03	5.40E-03
pci_bridge32_2	359	2.19E-03	4.38E-03	255	2.04E-03	4.09E-03
des_perf	1622	6.79E-04	1.36E-03	1154	4.49E-04	8.98E-04
ethernet	1079	1.08E-03	2.15E-03	771	6.95E-04	1.39E-03
vga_lcd	3993	3.21E-04	6.42E-04	2862	1.85E-04	3.70E-04
Average	731	3.33E-03	6.66E-03	521	2.32E-03	4.63E-03

Figure 10 demonstrates the overhead of transitions due to 100 different 128-bit fingerprints. ΔBT, ΔAT and ΔWT denote the best, average and worst case overhead respectively. We select 6 typical benchmark circuits with scan chain length of 1166, 1876, 2908, 5816, 8808, 16904 in Figure 10. In this case, 128 random numbers between [1…n] are generated to index 128 fingerprinting positions. After fingerprinting locations are determined, we randomly choose 100 connection styles out of all the possible connection styles to test the transition overhead by large size fingerprints. From Figure 10, we can see that the overhead is negligible for circuits with long scan chains. The worst case transition increments are less than 0.05% when scan chain length is longer than 5816 (B19). The overhead becomes lower when the circuit has a longer scan chain. On the other hand, for designs with relatively short scan chains, the result shows our proposed scheme could also control the worst case overhead under 0.18% (S38584 with scan chain length of 1166). In addition, the difference between best and worst case overhead is quite small, which indicates that the overhead is predictable and well controlled in our fingerprinting scheme.

3.7 Summary

Scan chain fingerprinting is an ideal solution to fingerprinting circuits that utilize scan-chains for DFT. The overhead is minimal as its only real effect is to increase the power usage of the device during testing. This method can also create fingerprints with more than sufficient length for most production lines. With this we can create larger than necessary fingerprints that can either be entangled or simply include more information, making it more difficult for attackers to counterfeit or break.

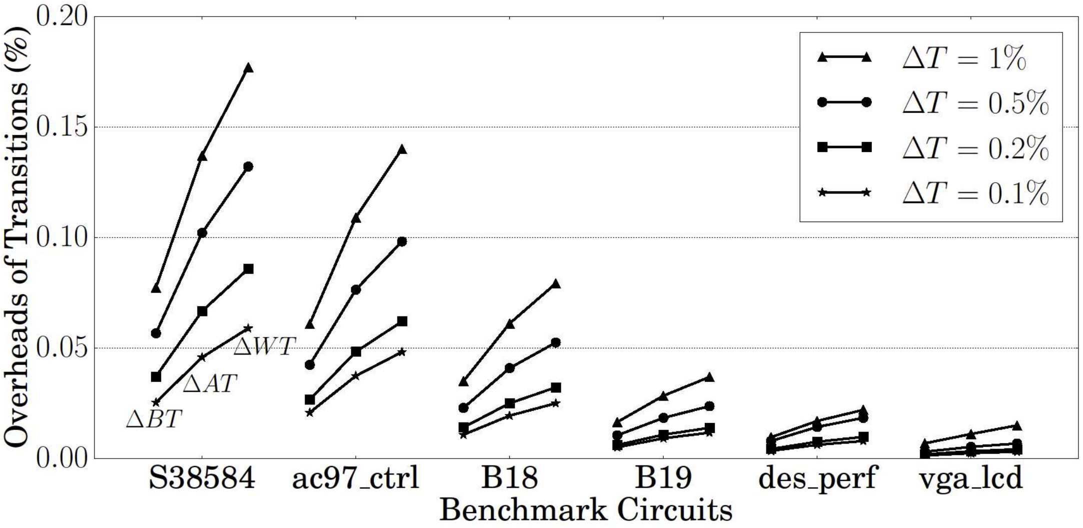


Figure 10: Overheads of transitions on six circuits due to 100 different 128-bit fingerprints. The scan chain length of these six circuits are 1166, 1876, 2908, 5816, 8808, 16904 respectively. For each circuit, we show the overheads under the limitation of ΔT = 1.

Chapter 4: RSN based IC Identification for Embedded Device

4.1 Identification of Embedded and IoT Devices

The notion of embedded systems has been around for about half a century and it boomed in the late 1990's when the embedded devices were networked. With the continuing advances and the convergence of multiple technologies, ranging from wireless communication to the Internet and from embedded systems to micro-electromechanical systems, the Internet of Things (IoT) emerged in the last decade in the form of large volumes of embedded devices connected by the Internet infrastructure to perform specific applications. Since then, IoT has been growing with an unprecedented pace and found applications in medical and healthcare monitoring, smart home and building surveillance, as well as in nation-wide infrastructures such as power grid, transportation systems, and environmental monitoring systems.

Security and privacy are among the key concerns for the development of IoT applications. It is pointed out that both the IoT and its Things are developed rapidly without appropriate consideration of the profound security challenges involved and the regulatory changes that might be necessary [47, 48]. A January 2014 article in Forbes listed many Internet-connected appliances that can already "spy on people in their own homes" including televisions, kitchen appliances, cameras, and thermostats [49]. Embedded devices in automobiles such as brakes, engines, locks, hood and truck releases, horn, heat, and dashboard have been shown to be vulnerable to attackers who have access to the onboard network. The vehicle-to-vehicle and vehicle-to-infrastructure communication makes everyone’s driving habit and daily commute routing public [50].

The serial number is perhaps the most popular and one of the earliest ways for IC identification. A serial number can be physically indented on the device or stored permanently in the memory. However, the fact that it can be easily removed or forged makes it unsuitable to countermeasure IP theft such as illegal reproduction, redistribution, and foundry overbuilding.

Several intrinsic unclonable IC tagging schemes based on silicon manufacture variation have been proposed. In [42], a technique was created to determine a circuit’s fingerprint through its glitches. In [43], the delay path variations are used to create the fingerprint for a circuit. Recently, a circuit identification method was presented in [52], where the authors embed chip IDs by replacing standard cells in the netlist with partial polymorphic gates. Upon activation of the control signal, the polymorphic gates will behave differently for certain input combinations and thus can be used to authenticate the chip. The unique challenge response pairs created by the physical unclonable functions have also been used for IC identification. These approaches are based on intrinsic fabrication variations, but they cannot detect IP theft because illegally copied or over-built ICs will have different variations and hence different identifications from the original copy. Therefore, the IP cannot be traced and authenticated.

Mathematically strong and well-developed cryptographic techniques exist for all kinds of security related applications such as data encryption/decryption, user and devices authentication, secure computation and communication. Most of these crypto security primitives or protocols are (extremely) computationally expensive (for example, performing the modular exponentiation operation for large numbers of hundreds of bits). Unfortunately, in the IoT domains, the devices are resource constrained and do not have the required computational power, memory, or (battery) power for such operations. As a result, in many IoT applications, both data and control communications, such as those between wearable/implantable medical devices and doctors or patients, are in plain text, which creates serious vulnerabilities.

4.2 RSN based IC Identification: Idea and Advantages

Reconfigurable scan architectures have been proposed [53] for decades. Compared to traditional scan design, RSNs allow flexible and scalable access to on-chip instrumentations in case of large scale integration, while significantly reducing test time. Recently, RSN with nearly arbitrary structure and functionality has been standardized by the IEEE P1687 [37]. The first generalized model enabling efficient formal verification and automatic generation of access patterns was presented in [54], which applies to a wide range of RSN architectures.

In this chapter, we propose a hardware security primitive as an alternative solution to the security of embedded and IoT devices. We utilize the testing infrastructure in these devices, which is compliant with IEEE 1149.1-2013 [36] and IEEE P1687 (IJTAG) [37], to create a unique identifier at the circuit level for each device which can be verified through a standard testing interface. More specifically, we adopt RSN and develop a fingerprint protocol to configure distinct RSN for each IC by utilizing the different connection styles between scan flip-flops. The testing vector set will need to be modified consequently to reflect the different RSN configurations and thus can be used as IC identification (ID). In addition, these IDs can be used to fingerprint the design or intellectual property (IP), they can facilitate IP metering and tracking, and they can also be used as the key for lightweight encryption and decryption.

The different connection styles in scan chain have been used in the literature for IP watermarking [46] and IP fingerprinting [51]. However, IP cores are highly used, in the forms of hard IP or firm IP, in the design of embedded and IoT devices [5]. The design details of these IP cores are unavailable; therefore, the previous IP protection techniques [46, 51] cannot be applied as they require changes to be made inside the IP cores. In our approach, we take advantage of the fact that such devices are tested by RSN and create unique device IDs at RSN without going into the IP cores. We apply it to the standard industrial design interface and demonstrate its usability in providing lightweight security for embedded and IoT devices. We analyze our approach to show that it will not introduce any design or performance overhead. Meanwhile, study on the ITC’02 benchmark indicates that the RSN configuration can easily accommodate 10^7 to 10^{186} unique device IDs.

4.3 RSN Based IC Identification Technique

Our IC identification scheme is built on top of the SIB-based RSNs as shown in Figure 11. It takes advantage of the fact that shift register *S* and shadow register *U* in each SIB can be chained by either the Q-D or the Q'-D connection style [21, 46]. In this approach, if the Q-D connection is used to chain *S* and *U* registers, the embedded ID bit is ‘0’, and if the Q'-D connection is used, the corresponding ID bit would be ‘1’. Therefore, for each SIB in the design, one identification bit can be embedded.

Suppose that the original design only uses Q-D connections for all SIBs in the RSN. Then, the chip ID of this design would be all 0s. To generate a new chip ID, the designer has the option of choosing among existing SIBs to modify their *S*/*U* connection styles. If *k* SIBs exist in the design, the designer can create unique digital IDs for up to 2^k chips.

As one might notice, when a Q'-D connection is used for *S*/*U* connection of an SIB, the negated value of *S* will be loaded to *U* during an update operation, which would make the original test inputs incorrect. Therefore, to ensure that all the instruments can be tested correctly, we need to adjust the test vectors for scan segments whose SIBs have been modified (Q'-D connection is used for their *S*/*U* registers). The adjustment only needs to be made to the test input which is shifted in during each update operation. We refer to this test input as *configuration sequence* as it determines the scan network topology after its corresponding update operation.

To adjust each configuration sequence, the following rules need to be followed

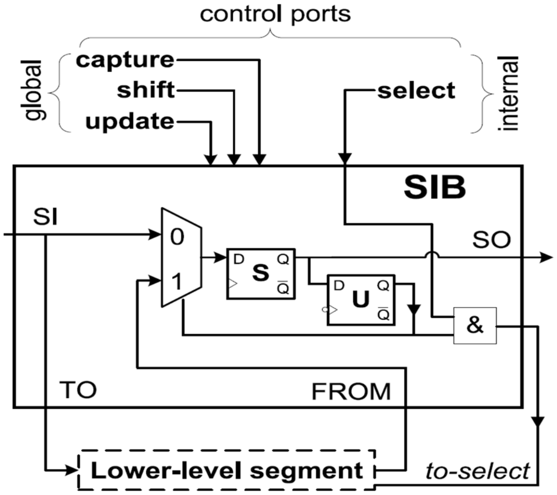


Figure 11: Implementation of a segment insertion bit [38].

for each bit in the sequence.

Rule 1. If the bit corresponds to an SIB whose *S*/*U* connection style is Q'-D, the value of this bit should be set to '0' for activating the directing mode and to '1' for enabling the bypassing mode.

Rule 2. If the bit corresponds to an SIB whose *S*/*U* connection style is Q-D, the value of this bit should be set to '1' for activating the directing mode and to '0' for enabling the bypassing mode.

These rules make sure that no matter what the style of *S*/*U* connection is in each SIB, always the correct value is stored in the shadow register and scan networks can be configured correctly. In the scan network depicted in Figure 12, suppose that the original design uses Q-D connections for all three SIBs, i.e. the design carries an ID value of '000'. In this case, to access only scan segments 1 and 3, a configuration sequence of '101' should be shifted in before the update operation. As mentioned before, this configuration sequence only works for this specific ID, and if the S/U connection style of any SIB changes, this sequence needs to be modified. For example, if an ID equal to '101' is assigned to the scan network in Figure 12, the configuration sequence for accessing scan segments 1 and 3 would be '000'.

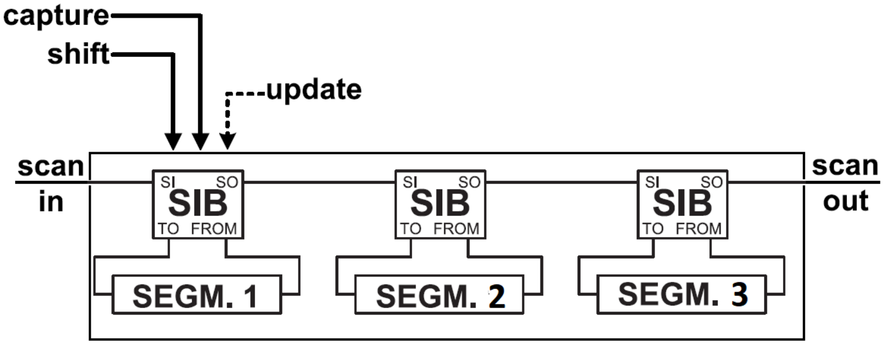


Figure 12: An example SIB based RSN for demonstrating test input adjustments.

Compared to the existing IC identification methods, our approach offers four advantages. First, it is practical as the ID bit locations in the scan network can be selected before fabrication, and the assignment of digital IDs are done at post fabrication stage. Therefore, all the designs can be fabricated with the same mask. Second, it incurs negligible overhead since the identification bits are added in the scan network, which will not affect the performance of core design. Third, it offers an additional non-destructive verification method which unlike other existing methods does not require de-packaging of the IC. Finally, and most importantly, it does not require any scan chain information from each of the IP cores and is suitable for embedded devices.

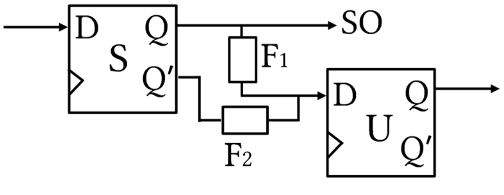


Figure 13: Programmable connections of S and U registers in ID-SIB.

4.3.1 Implementation

To implement the presented chip identification method, we propose to replace each original SIB in the design with a slightly different version of SIB called ID-SIB. The only change we made on the original SIB is that the connection style of ID-SIB's *S* and *U* registers can be programmed in post-fabrication stage, as shown in Figure 13. The connection programming is done by blowing up one of the two fuses of each ID-SIB in the scan network. In Figure 13, if the designer blows fuse F2, the *S*/*U* connection will be a Q-D style, and the corresponding identification bit for this ID-SIB would be '0', and if she chooses to blow the other fuse, the connection would be of Q'-D style, and the ID bit would be equal to '1'.

4.3.2 Security Analysis

To analyze the security of IC identification schemes, researchers consider two attack scenarios, ID modification and ID removal. For our chip identification scheme, the removal attack can be perceived as an instance of modification attack, for removing the chip ID, i.e. changing all the Q'-D connections in SIBs back to Q-D connections can be viewed as a modification attack targeting chip ID of all 0s. Therefore, in this section, we only focus on the ID modification attacks.

In ID modification attacks, adversary's goal is to change the ID of the chip. One possible motivation for an adversary to mount these type of attacks is to resale the chip to blacklisted customers for higher prices and avoid getting detected by the chip vendor. If the digital IDs of illegally distributed chips are not modified, the identity of the rogue customer responsible for selling these chips can be easily tracked by the chip IDs.

An adversary can mount ID modification attacks, only if he is capable of de-packaging, reverse engineering the chip and changing the connections of *S* and *U* registers in ID-SIBs. While we believe these assumptions about capabilities of adversaries are not realistic, especially in case of very large-scale ICs, we suggest choosing ID bits by the data integrity technique proposed in our previous work [51] to eliminate the possibility of such powerful attacks. Based on this technique, embedding ID bits for an IC is a 4 step process: (1): choose *N* ID bit locations, and replace the corresponding SIBs with ID-SIBs, (2): choose random values for *m* ID bits with $m < N$, (3): use this *m*-bit ID and an IC-specific key (K_{IC}) as the input to a one-way hash function to generate (*N*-*m*) bits, (4): use the final *N* bits as the ID bits to guide the selection of *S*/*U* connection styles at the selected ID locations. In this technique, the location of the *m*-bit ID and the value of the K_{IC} should be kept private to the IC vendor.

The proposed data integrity technique makes it difficult for the attacker to forge a chip ID, since a successful forgery requires knowing the value of K_{IC} and the exact location of the *m*-bit ID, which are only known to the IC vendor. Although it is possible for the adversary to change the connection styles between *S* and *U* registers in ID-SIBs, it will be challenging to make the correct changes that can maintain the property between ID bits.

4.4 Experimental Results

To validate our proposed IC identification scheme, we first see how many unique device IDs can be generated with our approach for real life circuits. Then, we discuss the design overhead.

4.4.1 Benchmark Circuits

To evaluate our identification scheme, we use the SIB based RSN benchmarks described in [54] which are based on ITC’02 SOC benchmark set [55]. Each ITC’02 benchmark circuit is specified by the modules in the SOC and their hierarchical structure, and modules are described by the numbers of their input, output, bidirectional terminals, scan chains and their lengths, test sets, and the (x, y) coordinate of their center on the SOC layout.

In the SIB based scan network benchmarks, two scan registers are designated for input and output pins of each module. In this design, doorway SIBs include or exclude lower level submodules, and instrument SIBs connect or bypass scan segments, depending on the input and output scan registers of each module from the active scan path as described in [56]. In Table 5, the details of the ITC’02 SOC benchmarks and their corresponding SIB based RSN designs are listed.

4.4.2 Potential in Creating Unique IDs

As described in section 4.3.1, to embed the identification bits, each SIB in the scan network needs to be replaced with an ID-SIB. Therefore, the number of potential ID bits for each chip is equal to the number of SIBs in its scan network, which is given in Table 5. As one can see in Table 5, with the exception of q127110, all the other benchmark circuits can potentially embed a good number of ID bits with the minimum of 40 bits (A586710) and maximum of 621 (P93791) ID bits, which correspond to 1.09×10^{12} and 8.70×10^{186} unique device IDs, respectively. Even in the minimum case, 1.09×10^{12} is a couple orders of magnitude higher than the number of devices in most of the real life embedded and IoT applications.

4.4.3 Design Overhead

The proposed IC and device identification approach has negligible performance overhead as the digital ID bits are only added in the SIBs of the scan network, which wouldn’t cause any overhead to the IP core design. Moreover, the overhead incurred on testing instruments is also negligible since no extra hardware is integrated into the design, and all the changes are local which avoids rerouting. For different RSN configurations, the testing vector can be justified, which is a one-time cost, so there will no change in test coverage.

Table 5: Characteristics of the ITC'02 Benchmarks and their corresponding SIB based Scan Networks.

Designs	Characteristics of the ITC’02 Benchmarks				Number of SIBs	Number of unique device IDs
	Modules	Levels	Scan segments	Register bits		
u226	10	2	40	1,416	50	1.13E+15
d281	9	2	50	3,813	59	5.76E+17
d695	11	2	157	8,229	168	3.74E+50
h953	9	2	46	5,586	55	3.60E+16
g1023	15	2	65	5,306	80	1.20E+24
f2126	5	2	36	15,789	41	2.19E+12
q127110	5	2	21	26,158	25	3.35E+07
p228110	29	2	254	29,828	283	1.55E+85
p34392	20	2	103	23,119	123	1.06E+37
P93791	33	2	588	97,984	621	8.70E+186
T512505	31	2	128	76,846	160	1.46E+48
A586710	8	2	32	41,635	40	1.09E+12

4.5 Summary

In this chapter, we proposed a novel IC identification approach which, compared to other existing schemes, is more practical, has lower design overhead and provides a non-destructive verification method. This method takes advantage of the difference connection styles in the scan chain to create unique device IDs. The testing vectors will be justified accordingly to maintain the test coverage, which becomes one way for the authentication of the device ID. It can be conveniently implemented on embedded and IoT devices to utilize their testing infrastructure compliant with IEEE 1149.1-2013 and IEEE P1687 (IJTAG). Experimental results indicate that on standard benchmark circuits, we can generate unique device IDs a couple of orders of magnitude higher than what typical embedded and IoT applications would need.

Chapter 5: Public-Private Partial Scan Chains

5.1 Rationale of Using Partial Scan Chain to Prevent Information Leak

Computer hardware has long been an attractive target for attackers to hack into any computing and communication system. Starting from the legendary power analysis attacks discovered 20 years ago [25] to the recent Intel Spectre [57] and Meltdown [58], the USENIX 2017 best paper winner CLKSCREW [59], the CCS 2017 best paper winner dolphin attack [60], the S&P 2016 best paper winner A2 hardware Trojan [61], security vulnerabilities in hardware design have been exploited for malicious purposes such as stealing sensitive information and gaining unauthorized control of the system. With the emerging IoT applications, where the devices are extremely resource constrained, many proven secure but computationally expensive cryptography protocols cannot be applied on such devices. Thus there is an urgent need to understand the hardware vulnerabilities and develop cost effective mitigation methods.

One established field in the semiconductor and integrated circuit (IC) industry, known as IC test, has the goal of ensuring that fabricated ICs are free of manufacturing defects and perform required functionalities. The concept of DFT has been integrated in the commercial IC design and fabrication for several decades. As its name suggests, DFT is to add test and debug features during design time in order to (1) reduce test time complexity, (2) improve test's fault coverage, and (3) enable in-field test and debug (that is the capability to test and debug after an IC is deployed). These are the three most important objectives of an IC test.

Test engineers need to access internal information of a system or an IC, such as the contents of registers, so as to diagnose the source of failures. Furthermore, they need a means to control a system's internal state to facilitate test. Scan chain is one of the most successful DFT techniques and can be found in almost all the modern designs. As we have introduced in Chapter 2, a scan chain basically connects all the flip-flops, which are the fundamental memory units that can store one bit of information, to form a chain and provides a scan in (SI) port and a scan out (SO) port to offer test engineers full controllability and observability of the scan chain and therefore system states for testing purpose. A test scan-enable signal is added to control whether the system is running in normal mode or test mode. The backbone of industrial DFT supports is based on a scan chain through the interface of JTAG [62].

Ironically, attackers have the same motivation as test engineers, that is, gaining controllability and/or observability of the system, but for malicious purposes. Scan chain gives them a perfect side channel to penetrate into a system through the JTAG interface. Cryptographic keys have been successfully cracked with information obtained from scan chain on many secure systems, including DES [28], AES [30], ECC [31], and RSA [32]. During test, by exploiting controllability and observability, it becomes theoretically easy to set values of flip-flops on the scan chain and consequently force the system into any state that an attacker wants to have [33, 34]. We will elaborate these attacks and their countermeasures in the next section. Here we just mention that existing countermeasures are mainly based on either controlling the switch between IC's normal mode and test mode, or obfuscating scan output values to confuse the attackers [63-66]. Inevitably these will introduce inconvenience and overhead against the aforementioned three primary objectives for IC test.

In this part of the dissertation, we propose a novel approach that can effectively defeat all the existing scan chain based attacks. The idea behind our approach is quite straightforward: *if you want to protect sensitive information and system states, then simply do not leak such information through the scan chain, no matter whether it is encrypted/obfuscated or not.* The rationale is similar to that in the tag game, known as Marco-Polo, played in the pool: if no one answers “Polo” to the call of “Marco”, the seeker will have no clue of how to tag other players. More specifically, traditional scan chain design has all the flip-flops connected in sequence to form a full scan chain. This provides test engineers the access to each flip-flop, resulting in full test coverage. But it also gives attackers full controllability and observability on the scan chain en route to access and control the system. In our approach, we will build *a partial scan chain which does not contain certain flip-flops, such as those storing sensitive data*. This completely denies the attacker’s accessibility and controllability on the un-chained flip-flops.

This concept of partial scan chain is not new. It has been used in IC test mainly to reduce the time complexity of test [80]. However, it fails to provide the test coverage that a full scan chain could have because test engineers cannot access those flip-flops that are not on the partial scan chain. This can be a serious problem for IC test. Sequential ATPG has been introduced as a solution for this, but it has many drawbacks including the increased test time, test power, and the inability to provide full test coverage. In this part, we will investigate this security-testability tradeoff and propose a novel approach that integrates partial scan chain design with a hardware implemented auxiliary FSM and some obfuscation circuitry in order to deliver secure scan chains with provable full test coverage.

Our approach features the following characteristics: first, by the design of partial scan chain, it is secure against all the known scan chain based attacks. Second, we develop a set of techniques to guarantee that all the original test vectors can be applied on the new secure partial scan in order to provide the same test coverage. Third, we propose a protocol to enable in-field test capability. It can be shown that potentially we can reduce both test time and test power, the two most important metrics for IC test. Finally, our design incurs chip area overhead to implement the auxiliary FSM and the obfuscation circuitry. Although the overhead on FSM is fixed and will become negligible as the size of the design increases, the overhead on the obfuscation circuitry could be proportional to the number of un-chained flip-flops.

In the remaining of this chapter, we first survey the existing scan chain based attacks and the reported countermeasures. Then we present the basic idea of our proposed public-private partial scan chain design and discuss the key challenges. We describe the general framework of this novel secure scan chain design and leave our answers to the design challenges to Chapter 6 and Chapter 7.

5.2 Scan Chain based Attacks and Countermeasures

In previous chapters, we have talked about all the advantages that scan chain has for testing purposes. Nevertheless, its security vulnerabilities have also been well-exploited. The two remarkable features of scan structure: full controllability and observability of the system states, not only give test engineers the convenience of testing and debug, they also open a backdoor to attackers. In this section, we discuss the rationale of various attacks. A detailed survey of known scan based attacks and existing countermeasures is also presented.

Scan based attacks can be broadly classified according to whether the attacker takes advantages of the scan chain's controllability or observability. Controllability offers an easy way for test engineers to apply test data from outside to the on-chip circuitry. However, it also allows malicious attackers to place the system to any state of their control, such as accessing certain protected states or scanning in corrupted data. This can be done with ease: set SE = 1 to force the system into test mode; configure the system to the desired state via SI port; reset SE = 0 to enter the normal functional mode to launch the attack. In other words, the attacker turns the SI port into a new channel for fault injection attacks [27]. It is easy to imagine that sensitive information from the keys used in the encryption algorithms to the functionality of the design can be easily extracted through such fault injection attacks.

On the other hand, observability refers to the capability to observe the contents of flip-flops deeply embedded in the design that would be hard to observe without scan chain. Clearly, attackers can also take advantage of this feature to obtain the internal states of the circuit at any point during test mode in a non-invasive manner. When crypto chips are performing encryption or decryption algorithms at the circuit level, some of the flip-flops on the scan chain will contain intermediate results of cryptographic computations. By switching the chip to test mode and then shifting out contents in these sensitive flip-flops, attackers can analyze the observed results to retrieve secret information. Most of the early scan based attacks that successfully cracked the proven secure encryption engines [28-32] belong to this category. The main threat of such attacks is that they can reduce the attacking efforts from years and months to days and hours [67].

Knowing that completely removing the access to scan chain is not an option, most of existing methods take one of the following approaches or some sort of combinations of the two. First, because scan based attacks need to switch between test mode and normal functional mode, it could be effective to tightly control when and how the SE signal is switched [30]. Second, almost all the scan based attacks rely on the observability of flip-flop contents from the SO port. Encryption and obfuscation methods have been proposed to confuse the attackers or limit their ability to understand the SO values [69-71].

Unfortunately, these approaches will introduce inconvenience and overhead for IC test, let alone the fact that they defer the scan design security to the security of encryption, obfuscation, and authentication protocols. Although there exist such proven secure protocols, their applicability to scan chain and in particular, scan chain on the resource constrained IoT devices is questionable.

In summary, scan test must be provided to both test engineers and end users, and it is desirable to allow them to apply new input test vectors. The security of scan chain cannot rely on controlling user's access to the scan chain because the system will not be able to distinguish an attacker from a legitimate user. On the other hand, it has been proven that scan chain vulnerability can be severe to system security.

Therefore, we need new countermeasures that can properly balance between security and testability (preferably full test coverage).

A lot of research efforts have been made to exploit the possible security loopholes due to the insertion of scan chain. Several scan-based attacks have been demonstrated. Hardware implementations of DES and AES are compromised that secret keys are discovered by differential attacks. Later, it was shown in [72] that scan chain also posed a security threat on stream ciphers. Meanwhile, public-key ciphers [31, 32] have been proven to be vulnerable to scan attacks. A survey paper summarizing the scan-based side-channel attacks was presented in [67]. It conducts a detailed investigation on the scan-based attacks on symmetric and public-key cryptographic hardware implementations. Various attack models are included and existing scan attack countermeasures are evaluated.

5.2.1 Scan based Side Channel Attacks

The first scan attack in the literature was proposed in [28], targeting the hardware implementation of DES. This is a two-phase scheme. First, by loading pairs of known plaintexts with one-bit difference in functional mode, the attacker determines the internal structure of the scan chain, such as the locations of input registers, which keep the plaintexts, and intermediate registers which store the intermediate results of cryptographic computations. This is achieved by using the procedures of observability attack as we previously mentioned. Then after analyzing the DES algorithm, the attacker is able to retrieve the secret key by applying only three known plaintexts.

Later, an attack on AES [30] was proposed by the same authors. The first step is to locate the intermediate registers by differential attack, the same as that on DES [28]. Then by executing only the first round of the encryption algorithm in functional mode and further dividing the first round into each distinct operation for analysis, the authors found that if there are a certain number of 1s in the first round result, the input pairs at the Substitution-box's input can be uniquely determined. By repeating the procedure, the attacker can recover the whole key.

Attacks against stream ciphers have also been presented. In [72], the Linear Feedback Shift Register (LFSR) based stream ciphers are targeted. In this scenario, the attacker can run the CUT for a certain number of clock cycles and scan out the states of internal registers. By observing the bit in a fixed position of the scan-out vectors for several clock cycles, the attacker can discover the bit-by-bit correspondence between the LFSR and the scan-out vectors. Thus, the LFSR-based stream cipher can be cracked.

Not only are symmetric-key algorithms attacked, recently public key ciphers are also under scan-based attacks. Elliptic curve cryptography (ECC) [31] and Rivest-Shamir-Adleman (RSA) [32] are demonstrated to be susceptible to scan attacks. The procedures of observability attack are applied to get the values of intermediate results. Then the attacker monitors a 1-bit time-sequence in the scan path to locate the register specific to the intermediate value of interest. The secret key is retrieved one-bit by one-bit.

5.2.2 Countermeasures to Scan based Attacks

A straightforward countermeasure to all scan based attacks is to unbound the scan chain after production [68]. However, this solution impedes the in-field diagnostic capabilities. Moreover, the scan chain can still be controlled and observed by physically probing the device. A similar approach is to limit the input test vectors to those that are provided by the chip maker. This will also prevent controllability attacks. But the cost of verifying the input test vectors could be very high given that there might be tens or hundreds of thousands test vectors. In addition, this will reduce the power of in-field test where the users may want to use some specific test vectors for diagnosis of potentially new faults.

In [69], a countermeasure called scan chain scrambling was introduced. The scan chain is first divided into small sub-chains. Then a multiplexer is inserted between scan chain segments. Under malicious attack, a random number generator is used to reconnect the sub-chains at a given frequency, which would produce unpredictable scan output. The main drawback of this method is the significant timing and area overhead it incorporates. Moreover, statistical analysis of the data scanned out from the chip can still reveal the scan structure and even the secret information. A lock & key technique was developed in [70], which was conceptually similar to the scan chain scrambling method. The scan chain is divided into smaller sub-chains of equal length. Instead of connecting the output of each sub-chain to the input of all other sub-chains by multiplexers, the lock & key scheme uses a seeded LFSR to randomly select a sub-chain to be filled by bits of the test vector when the key is incorrect. The disadvantage lies in the poor scaling for complex systems, where the number of sub-chains can be very large. The same authors proposed another low-cost secure scan solution by integrating a test key into test vectors [71]. Dummy flip-flops are inserted in the design but not connected to the combinational circuits. LFSR is then used to randomize the scan-out response when an incorrect key is integrated into the test vector. Thus, any reverse engineering attempt based on the scan-out response would become unlikely. However, extra dummy flip-flops require modification of the scan insertion process. Meanwhile, the key being added would incur overhead in test time.

An interesting alternative was proposed in [30], where mirror key registers (MKR) were provided to prevent the secret key from entering the scan chain when in test mode. The authors define two modes of operation: insecure and secure mode. MKRs work like normal registers during insecure mode except that a special key instead of the actual secret key is loaded. Scan test can be normally operated. While in the secure mode, the MKRs load the actual secret key. In this scheme, the contents of MKRs cannot be scanned out since switching back from secure mode to insecure mode requires a power off reset. However, for systems where the key is hardwired instead of being stored in a non-volatile memory, the proposed method does not work. Furthermore, the duplication of the entire key would incur a relatively high hardware overhead.

Several countermeasures focus on directly disabling the scan-out operations [73, 74]. An on-chip comparison is proposed in [73]. The scheme slightly changes the test procedure. Instead of shifting out test responses, the fault-free responses are shifted inside the circuit to be compared with the actual responses. Although the observability is minimized, this might reduce diagnostic resolution. Another approach in [74] uses a sensor to count the number of cycles in functional mode. Scanning out responses is forbidden if the circuit has been in functional mode for several cycles, which indicates the test is not happening. In [81], inverters are added between scan flip-flops to obfuscate the contents of these flip-flops. However, inverting values becomes completely useless against differential scan attacks.

5.3 Secure Partial Scan Chain: Idea and Challenges

As we have discussed earlier in this chapter, we observe that no matter how well existing countermeasures can protect the control over scan structure or obfuscate scan output, contents of flip-flops are still exposed to both test engineers and attackers and thus remain vulnerable. This leads us to the idea of removing certain flip-flops from the scan chain to create the so-called partial scan chain. These un-chained flip-flops are no longer accessible through the scan facilities such as SI and SO ports. Therefore information stored in these flip-flops will not leak outside the IC.

Figure 14 illustrates this partial scan chain idea. We can see that scan cell D_1 is

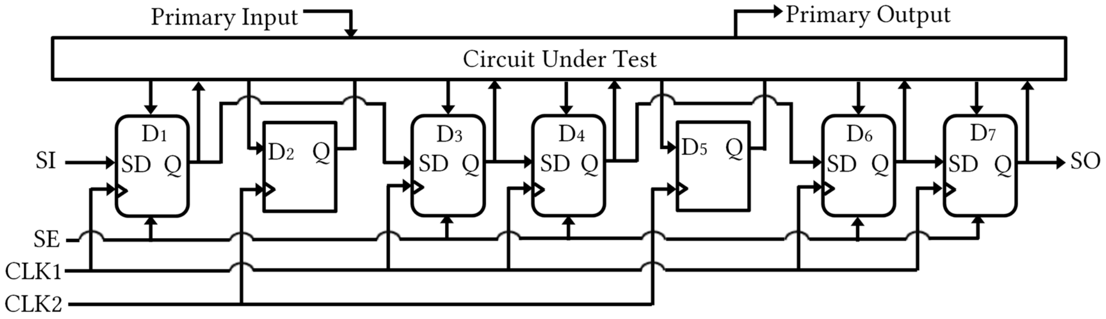


Figure 14: A partial scan chain where the second and fifth flip-flops are removed from the scan chain.

directly connected to D_3 and D_4 is directly connected to D_6, leaving the two flip-flops, D_2 and D_5, outside of the scan chain and remain as normal D flip-flops. When the scan enable signal SE is disabled, all the flip-flops will work in the normal mode and get values from the CUT. In the test mode with SE enabled, the partial scan chain works without flip-flops D_2 and D_5. As a result, no one can control or observe the contents of these two flip-flops through the scan chain side channel. This will have zero information leakage on the data stored in the un-chained flip-flops and thus provide the highest security level. However, it also introduces several key challenges to IC test:

Controllability Challenge. How to set input test vectors to these un-chained flip-flops? In the traditional full scan chain, each flip-flop can receive its designated values through the SI port and the scan chain. In existing partial scan chain design, sequential ATPG is used to generate test vectors that are different from those for the full scan chain. This results in long test time and reduced test coverage. We aim to use the same input test vectors for full scan chain to keep full test coverage and thus facing the challenge of how to control the un-chained flip-flops.

Observability Challenge. How to verify the output test vectors on these un-chained flip-flops? Similar to the controllability challenge, because the un-chained flip-flops are not on the scan chain, we will not be able to observe their contents from the SO port. If these is any fault occurred in these flip-flops, we cannot detect it, resulting in the reduction of test coverage. However, exporting the contents of the the un-chained flip-flops to the outside of the IC will defeat our purpose of securing scan chain from information leak. This is a problem that needs to be addressed.

Security Challenge. How to solve the above two challenges without compromising the security provided by the un-chained flip-flops? What makes the controllability challenge and the observability challenge difficult is the requirement that any solution needs to ensure that the contents in the un-chained flip-flops are secure, which is the sole purpose of our partial scan chain. Secure analysis against both known scan chain attacks and other potential attacks targeting our proposed design needs to be conducted.

These challenges can be better understood after we describe the framework of public-private partial scan chain next. Then we will provide our solutions to these challenges in Chapter 6 and Chapter 7, respectively.

5.4 Framework of the Public-Private Scan Chains

The basic idea of securing the scan chain behind our approach is to remove flip-flops that store sensitive information from the full scan chain in order to restrict the access to them through the SI and SO ports. In this section, we elaborate our partial scan designs as solutions to the challenge of designing secure partial scan while providing full test coverage.

5.4.1 Generic Design of the Public-Private Partial Scan Chains

Removing certain flip-flops, which we will refer to as *un-chained flip-flops*, limits test engineer's ability in testing the design. In order to guarantee the same full testability that can be achieved by the full scan design with a given set of test vectors, it will be necessary and sufficient to give test engineers both the controllability and observability of the un-chained flip-flops such that they can enter the input test vector into the CUT and then observe the test output to detect fault. Figure 15 depicts the generic structure of our proposed public-private partial scan chain.

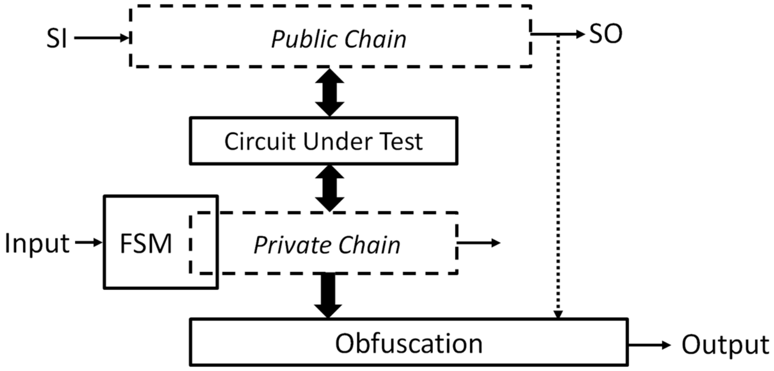


Figure 15: Structure of the generic public-private secure partial scan chains.

On the top we have the public scan chain which is a normal partial scan chain contains flip-flops that do not need access control. Through the SI and SO ports, everyone can observe and control the values in these flip-flops when the scan chain is set in the test mode. The novelty in our approach is the private scan chain at the bottom, which provides both full test coverage and security as we will explain below.

First, to keep the same fault coverage as the full scan chain can provide with a given set of test vectors, the standard partial scan chain design method is to re-generate test vectors based on only the flip-flops in the partial scan chain. The industrial tool known as sequential ATPG was developed for this purpose. However, this does not guarantee full test coverage, and can incur overhead in both test time and power. Therefore, we propose a different approach by building a private scan chain to connect all the un-chained flip-flops. This makes it possible to provide the full test coverage by using the same set of test vectors as in the case of full scan chain and also avoids the burden of re-generating test vectors. However, if we use the traditional SI and SO ports for the private scan chain, that will be the traditional multiple chain design and does not provide any security guarantee.

We observe that what the test engineers need is a way to enter the input test vector to each of the flip-flops, including those un-chained ones that are not on the public partial scan chain. As shown in Figure 15, we use a hardware implemented FSM to generate these input test vectors and then shift them into the un-chained flip-flops. The interface to the outside will be a control input port from which input values to the FSM can be entered in a sequence to direct the FSM to the target ending state. At the ending state, the flip-flops in the FSM will have values required in the test vector. We will elaborate this in Chapter 6.

For the same reason, the test engineer does not necessarily need to know the value of each flip-flop, as long as the output values from the chip matches the output test vector, the test is passed. Therefore, we propose to add an obfuscation unit to hide the real output values. Existing encryption and output scrambling methods are based on the same idea. Next, we outline the two different implementations of the obfuscation units and the detailed design will be described in Chapter 7.

5.4.2 Private Chain with a Single Partial Scan Chain

Figure 16 shows the structure of the proposed public-private partial scan design with a single private scan chain. It consists of three major components:

- *Block A:* a public partial scan chain with *n* flip-flops that is controlled by the regular SI and contributes to but cannot be directly observed from the SO;
- *Block B:* a private partial scan chain with the *k* un-chained flip-flops that we want to control access. These flip-flops are connected to the CUT but not to the SI/SO ports. The first *l* un-chained flip-flops are used to implement an FSM with additional combinational logic. An external control input CI is used to feed input to the FSM.
- *Block C:* a set of *l* dummy flip-flops that temporarily store the values of the *l* un-chained flip-flops. These dummy flip-flops and those in the public partial chain will feed into an exclusive-or (XOR) gate to produce the SO. A linear feedback shift register (LFSR) is used as most of the time we have *l* << *n*, that is, the public partial chain is normally longer than the private partial chain.

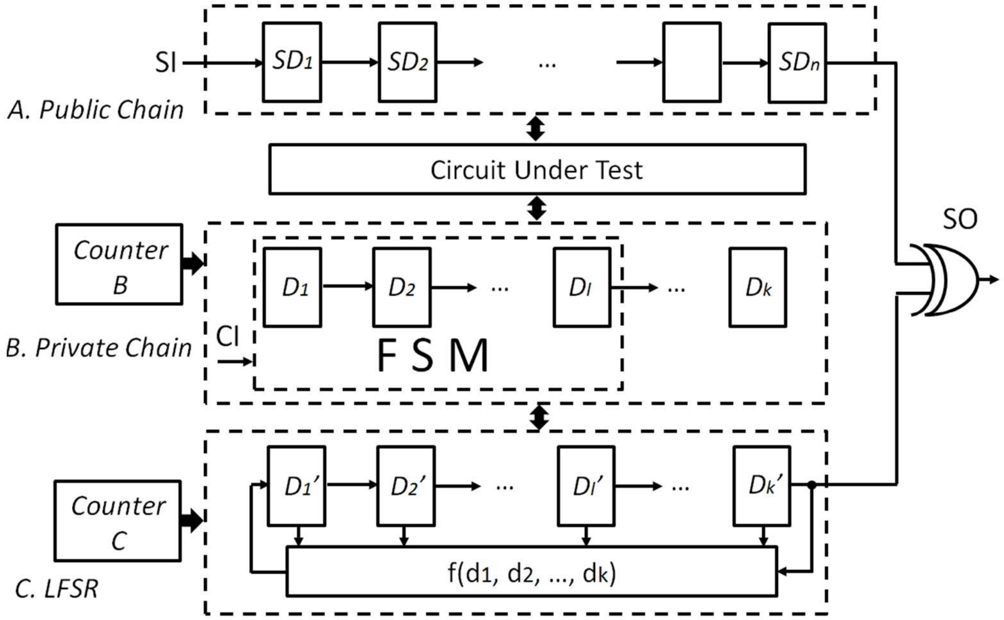


Figure 16: Structure of the public-private secure partial scan chains with LFSR based scan output obfuscation.

The *n+k* flip-flops in *Block A* and *Block B*, which if chained together would have formed a full scan chain, are connected to CUT to support the normal functionality of the CUT. The output signals from *Block A* and *Block C* will be XOR-ed and the result can be observed from the SO port.

To generate a test input and output pair, we push the values of the flip-flops in *Block A* through the SI port of the partial scan and provide a specific input sequence through the external input of the FSM in *Block B* to set values for the un-chained flip-flops. Note that these two procedures are performed in parallel and can be designed without test input vector loading overhead (for example, in case when *n>>k*). After the CUT executes with the desired test input vector, the *n+k* flip-flops in *Block A* and *Block C* will store system state information with the *k* flip-flops in *Block C* storing a backup copy for flip-flops in *Block B*. The test output vector is generated by bit-wise XOR-ing the flip-flops in *Block A* and *Block C*.

As shown in *Block B* of Figure 16, the first *l* un-chained flip-flops are directly connected to extra combinational logic to form a sequential circuit that implements an FSM controlled by the external input signal/vector CI. In the meantime, the remaining *(k-l)* un-chained flip-flops also need to be set accordingly. To achieve this, all the *k* un-chained flip-flops are connected like a shift register (or another partial scan chain) as illustrated in Figure 16. Every time the first *l* un-chained flip-flops reach a set of desired values, they will be right shifted to the next *l* flip-flops in the chain. Afterwards, the first *l* flip-flops will switch back to the extra combinational logic while their contents will be reset to the reset state of the FSM. Then the FSM will transit according to the control input sequence from CI to store the next set of desired values into the first *l* un-chained flip-flops. During these transitions, the other *(k-l)* un-chained flip-flops should remain unchanged. This process will be repeated until all the *k* un-chained flip-flops are correctly set to the values equal to those in the test input vector. The motivation of this approach is to avoid the hardware overhead in implementing large FSM with *k* flip-flops. A Counter B is implemented to control the timing such as when the FSM should reset.

In a nutshell, by adding FSM in *Block B* for controllability and *Block C* for observability, the unchained flip-flops in *Block B* can be tested just as if they are in the full scan design. We will elaborate the LFSR based scan output obfuscation in Chapter 7.

5.4.3 Private Chain with Multiple Partial Scan Chains

We now present the structure of the proposed public-private partial scan design with multiple private scan chains, where the scan output is protected by a configurable PUF instead of the LFSR.

Figure 17 depicts the structure of this proposed public-private partial scan chains. The CUT will be connected to both the public scan chain on the top and the private scan chains at the bottom. The flip-flops in the public chain can be accessed directly through the regular SI and SO ports. A parallel input (PI) ports and parallel output (PO) ports provide the interface to the flip-flops in the private chains. The key challenge to achieve full testability is how to control and access the private scan chains, which we will highlight the main ideas and then elaborate details in Chapter 7.

Compared to the single private scan chain design with an LFSR, there are several notable difference here. First, the SO port in the public chain provides direct output and it does not involved in the obfuscation of the output from the private chains. Second, the un-chained flip-flops form multiple chains and there are parallel output (PO) ports. Third, instead of the LFSR, a configurable PUF circuitry is used and the PUF bits are used to XOR the parallel output from the multiple private chains.

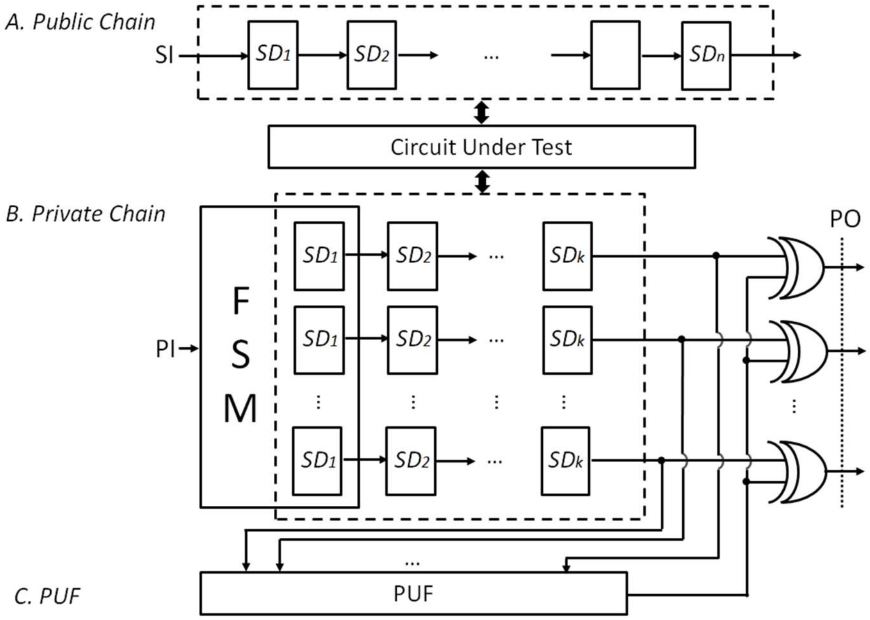


Figure 17: Structure of the public-private secure partial scan chains with configurable PUF based scan output obfuscation.

The generation of the input test vector with the multiple private scan chains is the same as that in the single private chain except that the *k* un-chained flip-flops in the FSM will serve as the start of *k* parallel private chains as shown in Figure 17. As a result, instead of shifting one bit at a time, *k* bits can be shifted simultaneously in this implementation, making it faster to have the input test vector set.

To validate the test output, the parallel private chains will shift out their contents and a comparison with the expected output would tell whether there is any fault or defect. Most of the existing countermeasures to observability based attacks can be applied to prevent the contents from leaking to the attackers. In our approach, we use a cost-effective secure method based on the configurable RO PUF [75].

5.5 Summary

In this chapter, we give the rationale of the proposed public-private partial scan chain framework that can provide full testability and security. We briefly mention that a hardware implemented FSM will be used to generate the input test vectors for the private scan chain(s). We also outline the key ideas in our approach with focus on how the un-chained flip-flops are connected in the private scan chain(s). Two implementation of the private scan chain with single and multiple partial chains are discussed and further details will be elaborated in Chapter 6 and Chapter 7.

To conclude this chapter, we mention that our approach guarantees the same test coverage of a full scan chain because we can test all the given input test vectors. At the same time, the separation of public partial scan chain and private partial scan chain(s) provides security to the un-chained flip-flops. Finally, recall that full testability also includes the capability of in-field test with both manufacturer-provided test vectors and any new designed ones to diagnose unknown faults. The proposed public-private partial scan chain structure enables this capability and we will elaborate this in Chapter 7.

Chapter 6: Controllability Challenge: Input Test Vector Generation

Recall that we have discussed three challenges for our secure public-private partial scan chain in section 5.3: controllability challenge, observability challenge and security challenge. In this chapter, we work on how to address the controllability challenge – how to set input test vectors to these un-chained flip flops in order to guarantee full test coverage. To begin with, we present our solutions by a small example. Then we elaborate each of the techniques that we have developed and conclude with experimental evaluations.

6.1 Illustrative Example and Problem Formulation

Assume that Table 6 lists all the 5 test vectors designed to detect the defects in the circuit shown in Figure 14. Note that the second bit and the fifth bit cannot be accessed through the partial scan chain. We underline these bit values in Table 6.

In a full scan chain as we have discussed earlier, we can enter from the SI port the first input test vector tv1, 0000000, test the circuit and then shift out the response from the SO port for comparison with the expected output test vector 0110010. Meanwhile, the second input test vector tv2, 1011000 will be scanned into the scan chain for the second test. This process repeats till all the test vectors are tested.

When flip-flops D_2 and D_5 are removed from the scan chain, we will not be able to enter the underlined bits directly to these two flip flops. The controllability challenge seeks ways to set the underlined values into D_2 and D_5. Our basic solution to this challenge is test vector reordering. For example, assuming that initially both D_2 and D_5 have value 0, we can start with tv1 and enter the 5-bit input vector, 00000, to the 5 flip-flops in the scan chain. This along with the initial values in D_2 and D_5 will give us the original 7-bit input vector for tv1. After test, D_2 and D_5 will have 1 and 0 as their contents. Now instead of testing the next text vector, 1011000, whose values at D_2 and D_5 do not match their current contents, we will test tv4. This is because the current contents of D_2 and D_5, 1 and 0, match the desired values for D_2 and D_5 in tv4. Hence, we simply shift the input test vector 10010 through SI port and the circuit will be ready for test vector tv4. We can continue this process as long as we can find a test vector whose input values at the positions of the unchained flip-flops match the current contents in these flip-flops.

Unfortunately, this solution does not solve the problem completely for two reasons. (1) In the above example, after applying tv4, D_2 and D_5 will have 01, but there is no untested input vector with 01 at these two positions. So which test vector

Table 6: Test vectors for the 7-flip-flop design in Figure 14

Index	Input Vector	Output Vector
tv1	0$\underline{0}$ 00$\underline{0}$00	0$\underline{1}$ 10$\underline{0}$10
tv2	1$\underline{0}$ 11$\underline{0}$00	1$\underline{1}$ 10$\underline{1}$01
tv3	0$\underline{1}$ 01$\underline{1}$01	1$\underline{0}$ 01$\underline{0}$01
tv4	1$\underline{1}$ 00$\underline{0}$10	0$\underline{0}$ 10$\underline{1}$11
tv5	1$\underline{1}$ 01$\underline{0}$00	0$\underline{1}$ 11$\underline{0}$01

we are going to use next and how to ensure that D_2 and D_5 will have the correct values for the next test vector? (2) When there are multiple input vectors that match the current contents of D_2 and D_5, for example, tv4 and tv5 after testing tv1, which one should we choose? In this example, if we choose tv5, we can further test tv4. However, if we choose tv4, we won't be able to test tv5. It appears that the selection of the next test vector does matter, which could make this problem more challenging. Fortunately, as we will prove in section 6.3, we can always find one optimal solution regardless which test vector we choose.

We now give a formal formulation of the controllability challenge and present our solutions in the rest of this chapter. Consider a circuit with N flip-flops and M test vectors, $\{(X_i, Y_i)\}_{i=1,2,\ldots,M}$, generated by a commercial ATPG tool to provide the desired testability. Denote the N-bit input test vectors by $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iN}\}$ and the N-bit output test vectors by $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{iN}\}$. During test with a full scan chain, each input test vector X_i will be shifted into the scan chain. Then the corresponding primary input values will be applied on the circuit and the circuit will run at the functional mode for one or more clock cycles. The test response, which is the values stored in the N flip-flips, can be read out from the SO port and will be compared with the expected output test vector Y_i for fault detection and diagnosis.

In the proposed secure partial scan design, L flip-flops will be removed from the full scan chain for security concerns. Let $P = \{k_1, k_2, \ldots, k_L\}$ be the set of L removed flip-flops. One cannot access these L flip-flops via SI or SO ports. Therefore, the partial scan chain will not provide any controllability and observability on these hidden flip-flops. Security is achieved at the cost of losing testability. The controllability challenge seeks to answer whether and how to deliver full testability through the partial scan chain:

Secure partial scan chain with full test coverage. Consider a circuit with N flip-flops and M test vectors, $\{(X_i, Y_i)\}_{i=1,2,\ldots,M}$, where X_i are the input test vectors and Y_i are the output test vectors. L flip-flops are removed from the scan chain. Can the resulting partial scan chain provide the full testability and security simultaneously? More specifically, can one (i) apply each of the M input test vectors X_i on the circuit and test the responses against the corresponding output test vectors Y_i, and (ii) apply test vectors other than $\{(X_i, Y_i)\}_{i=1,2,\ldots,M}$ for in-field test, without (iii) accessing the data stored in the L hidden flip-flops.

In the rest of this chapter, we first propose a graph representation for the portion of test vectors that is not in the partial scan chain (that is, the underlined bit values in Table 6). Then with the help of this representation, we elaborate our approaches of test vector reordering, reusing, and secure generation to provide full test coverage.

6.2 Hidden-Test-Vector Graph

Recall that $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iN}\}$ is the input test vector and $P = \{k_1, k_2, \ldots, k_L\}$ is the set of L removed flip-flops. Denote

$X_i|_P = \{x_{is}\}$ where $s \in P$

$X_i - X_i|_P = \{x_{is}\}$ where $s \notin P$

Similarly, we can define $Y_i|_P$ and $Y_i - Y_i|_P$. Thus, $(X_i - X_i|_P, Y_i - Y_i|_P)$ is the portion of the test vector (X_i, Y_i) that can be accessed on the partial scan chain. $(X_i|_P, Y_i|_P)$ are the bits in the input-output test vectors that are for the un-chained flip-flops, which we will refer to as hidden input-output test vectors.

We convert the M test vectors $\{(X_i, Y_i)\}_{i=1,2,\ldots,M}$ into a directed graph, which we call **hidden-test-vector graph**, described as follows:

(1) each node, tv_i, represents a test vector (X_i, Y_i)

(2) a directed edge from node tv_i and node tv_j exists if and only if $Y_i|_P = X_j|_P$

Figure 18 is the hidden-test-vector graph converted from the five test vectors listed in Table 6. For example, there is an edge from tv1 to tv5 because the hidden output vector of tv1 matches the hidden input vector of tv5. The hidden-test-vector graph has the following special property which plays a crucial role in our proposed test vector reordering approach.

Shared neighbor property. In the hidden-test-vector graph, if two nodes have a common child then they will share all their children. Similarly, if two nodes have a common parent node, then they will share all the parent nodes.

[Proof] Suppose that two nodes (X_i, Y_i) and (X_j, Y_j) have a common child (X_k, Y_k), we have $Y_i|_P = X_k|_P$ and $Y_j|_P = X_k|_P$. So $Y_i|_P = Y_j|_P$, which means that the two test vectors (X_i, Y_i) and (X_j, Y_j) have identical hidden output vector. For any child (X_s, Y_s) of node (X_i, Y_i), $X_s|_P = Y_i|_P = Y_j|_P$. So (X_s, Y_s) must also be a child of node (X_j, Y_j). For exactly the same reason, any child of (X_j, Y_j) is also a child of (X_i, Y_i). Therefore, nodes (X_i, Y_i) and (X_j, Y_j) will share all their children, that is, nodes with hidden input vector equals to $Y_i|_P$. The other part of the property can be proved in the same way.

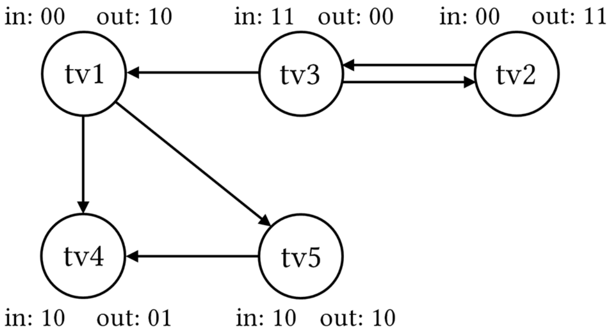


Figure 18: The hidden test vector graph based on the five test vectors in Table 6. “in” represents the hidden input vector $X_i|_P$ and “out” is the hidden output vector $Y_i|_P$.

6.3 Test Vector Reorder

As we have seen in the illustrative example in the beginning of this chapter, if a test vector’s hidden input portion matches the hidden output vector of another test vector (e.g. tv4 and tv1), we can use the partial scan chain to test these two test vectors one after another. Formally, we have

Lemma 1. In the hidden-test-vector graph, for any directed path, all the test vectors corresponding to the nodes along the path can be tested via the partial scan chain if we can set the first node’s hidden input vector on the L hidden flip-flops.

[Proof]. If we can set the first node’s hidden input vector on the L hidden flip-flops, simply entering the rest test input vector by the SI port of the partial scan chain will enable us to test the first test vector. After the test, according to the definition of the hidden-test-vector graph, the L hidden flip-flops will have the same values as the second input test vector requires. Using the same method, we can test the second test vector by entering its other input vector through the partial scan chain. This process can continue till we reach and test the last test vector in the path.

For example, in Figure 18, there is a path tv2→tv3→tv1→tv5→tv4. If we can set the two hidden flip-flops to be 00, which is the hidden input vector for tv2, we can test all the 5 test vectors without accessing the hidden flip-flops again.

From Lemma 1, we see that if we can find n paths to cover all the nodes in the hidden-test-vector graph, we only need to access the hidden flip-flops n times to test all the M test vectors. Because access to the hidden flip-flops is prohibited or expensive, we propose the following minimization problem:

Test vector reordering problem. Given a hidden-test-vector graph, find the minimal number of paths to cover all the nodes.

Recall that the well-studied NP-complete Hamiltonian path problem: in the mathematical field of graph theory, a Hamiltonian path is a path in an undirected or directed graph that visits each vertex exactly once [82]. If one can solve the test vector reordering problem and the optimal solution has only one path, that path will be a Hamiltonian path. If the optimal solution has more than one path, the answer to the Hamiltonian path problem will be "No". This seems to suggest that we will not be able to solve the test vector reordering problem optimally in polynomial time. However, as we will show next, the hidden-test-vector graph is not a general graph. It has the shared neighbor property which allows us to solve the test vector reordering problem optimally and efficiently.

Lemma 2. In the hidden-test-vector graph, when a node has multiple outgoing edges, choosing any edge can lead to an optimal solution to the test vector reordering algorithm.

[Proof] Without loss of generality, consider Figure 19(a) where node A has two outgoing edges to node B and node C. We first consider the case when there is no edge between B and C. Nodes B and C will either have A as their only common parent node or they have other common parent nodes. When A is the only common parent, it's obvious that choosing either edge AB or edge AC gives us exactly two paths as the optimal solution, that is S_1→A→B→S_2 and C→S_4 in the first case, and S_1→A→C→S_4 and B→S_2 in the second case.

Now assume that there is a node D that is the parent of B or C. Note that because of the shared neighbor property, node B and node C will share node D as a parent (see Figure 19(b)). If we choose edge AB, assuming that an optimal solution contains paths S_1→A→B→S_2 and S_3→D→C→S_4, we now construct an optimal solution, in terms of the number of paths in the solution, with edge AC being chosen instead of edge AB. Clearly, we can build two paths S_1→A→C→S_4 and S_3→D→B→S_2, which cover all the nodes that the original two paths S_1→A→B→S_2 and S_3→D→C→S_4 cover. Combined with other paths in the optimal solution when choosing edge AB, this gives another optimal solution when edge AC is chosen.

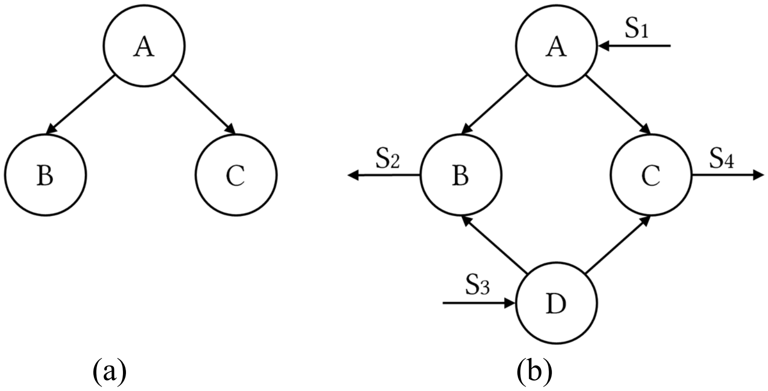


Figure 19: Proof of Lemma 2.

A special situation that needs to be taken into account is, as shown in Figure 19(b), when $S_1 \rightarrow A \rightarrow B \rightarrow S_2$ and $S_3 \rightarrow D \rightarrow C \rightarrow S_4$ (or simply S_2 and S_3) are connected. In that case, an optimal solution uses only one path $S_1 \rightarrow A \rightarrow B \rightarrow S_2 \rightarrow S_3 \rightarrow D \rightarrow C \rightarrow S_4$ to cover this portion of the graph. Using $S_1 \rightarrow A \rightarrow C$ does not seem to be able to cover $B \rightarrow S_2 \rightarrow S_3 \rightarrow D$. We consider two paths $S_1 \rightarrow A \rightarrow C \rightarrow S_4$ and $S_3 \rightarrow D \rightarrow B \rightarrow S_2$ (a loop indeed) and then insert the second path/loop between the link $A \rightarrow C$. This insertion can be conveniently done by matching the hidden input vector of each node in the loop with the hidden output vector of nodes in other paths. In this example, we will find the hidden input vector of node B matching the hidden output vector of node A. So we can break the path at $A \rightarrow C$. The shared neighbor property guarantees the existence of edge $D \rightarrow C$ because $D \rightarrow B$ is part of the loop. After the insertion, we have exactly the same solution as before: $S_1 \rightarrow A \rightarrow B \rightarrow S_2 \rightarrow S_3 \rightarrow D \rightarrow C \rightarrow S_{4.}$

For the case when nodes B and C are connected by an edge, we can use the same method to prove that from any optimal solution that chooses edge AB, a solution that has the same number of paths can be constructed by choosing edge AC. This completes the proof of Lemma 2.

The proof of Lemma 2 shows an effective method to construct an optimal solution to the test vector reordering problem:

Step 1. Start with an arbitrary node (preferable one without any incoming edge) to build a path by choosing any outgoing edge and continue until a node with no outgoing edge is reached.

Step 2. Repeat step 1 for all the nodes that are not covered by any path.

Step 3. For all the paths built in step 2, if the path is a loop, check to see whether it can be inserted into another path. If the start/end node of the loop matches any edge in another path from step 2, then the loop can be inserted in this particular edge's position in the path. Thus, the loop and the path found can merge into a single path.

Apparently, each edge will be visited at most once in Step 1, so it takes O(|E|) time to build all the paths in Step 1 and Step 2. Since a loop needs to be compared with all the other edges of all found paths until a position is found, O(|V|) time is required in the worst case. Hence, for all the loops, it takes $O(|V|^2)$ to check for loop insertion in Step 3. Therefore the above algorithm's complexity is $O(|E| + |V|^2)$, where |V| and |E| are the number of nodes and edges in the hidden-test-vector graph, respectively.

6.4 Test Vector Reuse

The rationale behind the test vector reordering technique is to continue running the test vectors with the minimum number of times to set the hidden input vectors (we need to load these values at the start of each path). In order to save test time and power, we avoid testing the same test vector more than once. However, such cost could be lower than that to load values to the hidden flip-flops, which we will discuss in the next subsection. Therefore, if a node is on multiple paths in the hidden-test-vector graph, reusing test vectors can connect these paths and save the efforts of loading the hidden flip-flops. We first show this concept by the following example.

Consider the six test vectors in Table 7, where only the hidden bits in the test vectors are shown for simplicity. After test vector reordering, three paths are obtained: tv_1→tv_6→tv_2, tv_3→tv_4 and tv_5. If we can reuse tv_6, two paths tv_3→tv_4 and tv_5 can be merged to one: tv_3→tv_4→tv_6→tv_5, which will save the efforts to load the hidden flip-flops. However, this reuse process leads to test time overhead due to the testing of the same test vector (tv_6 in this case) twice. Therefore, a parameter $maximum\_steps$ can be defined as follows to control the maximum number of test vectors we can reuse to connect two paths.

$$maximum\_steps = \frac{\text{cost of loading hidden flip flops}}{\text{cost of running one test vector}}$$

In this section, we propose and study the following problem:

Table 7: Test vectors with the bits in P indicated

tv_1: (11, 01)	tv_2: (00, 11)	tv_3: (10, 10)
tv_4: (10, 01)	tv_5: (00, 00)	tv_6: (01, 00)

Test vector reusing problem. Given a hidden test vector graph and an optimal solution to the test vector reordering problem, reduce the number of paths by reusing nodes under the $maximum\_steps$ constraint.

Let {S_1, S_2, …} be an optimal solution to the test vector reordering problem. For each path S_i, let h_i be its head node and t_i be its tail node. We convert the test vector reusing problem to the traveling salesman problem (TSP) and then use some TSP solver to solve the problem.

Step 1. For each node t_i, perform a breadth first search in the hidden test vector graph till all the head nodes h_j ($j \neq i$) are found or maximum_steps is reached.

Step 2. Create a weighted directed "path" graph, where each path S_i is a node and an edge from S_i to S_j indicates h_j was found from t_i in Step 1. The distance between t_i and h_j in the hidden test vector graph is used as the weight of edge $S_i \rightarrow S_j$.

Step 3. Apply a solver to the traveling salesman problem (TSP) on each of the connected component in the above "path" graph.

6.5 Hidden Input Vector Generation by Finite State Machine

After test vector reordering and reusing, it is our hope that all the test vectors will form a long path so we can do the test without resetting the values in the hidden flip-flops. If this fails, we have to find a way to access the hidden unchained flip-flops. Recall that we un-chain these flip-flops for security purpose. Hence, security has to be considered when we design mechanisms to access these hidden flip-flops.

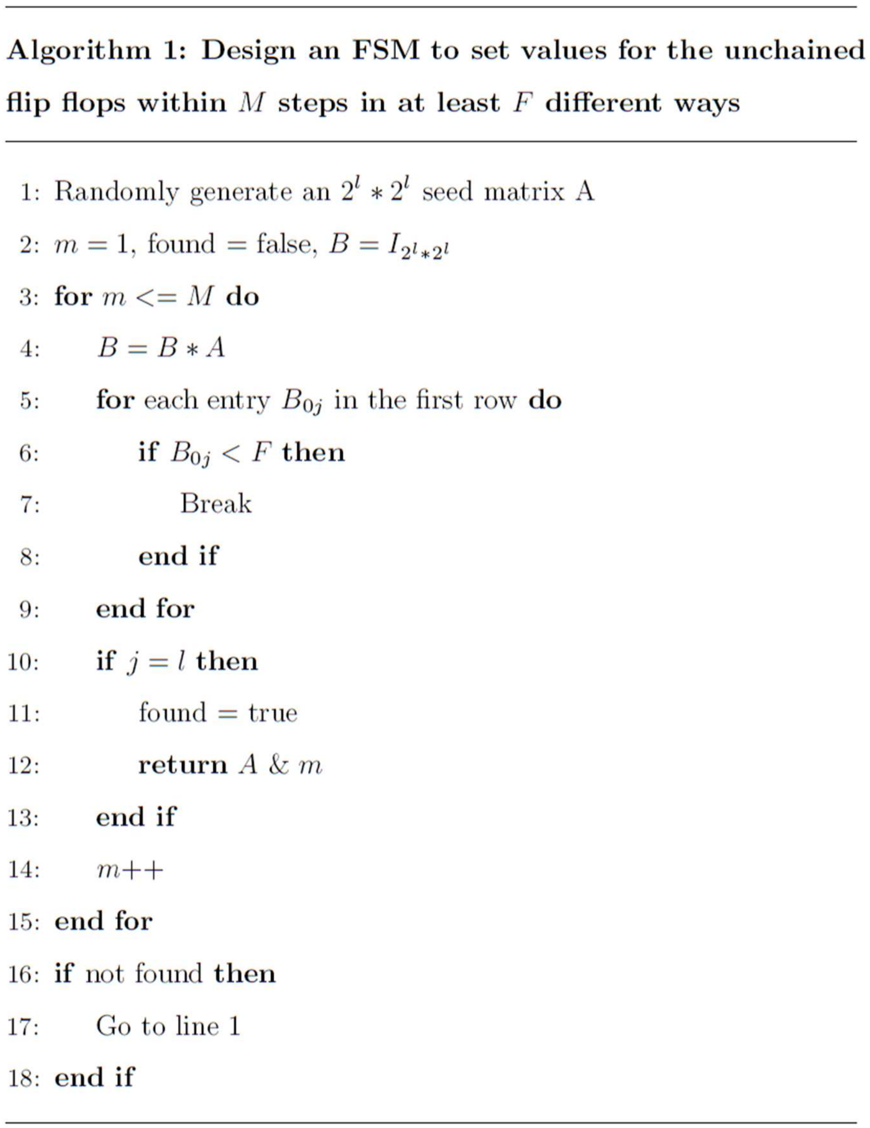
In this section, we describe how we build an FSM to gain control of the un-chained flip-flops. The basic idea is that as we traverse the FSM, whose states are represented by a set of flip-flops, the contents of these flip-flops will change. Once they change to the values that we want to set in the hidden flip-flops or a portion of them, we can stop traversing the FSM and copy the state of the FSM to the hidden flip-flops.

There are two requirements for the design of this FSM: (1) there should be a reset state to reset (not necessarily to be all 0's) the value for each of the hidden flip-flops; (2) all the other 2^k-1 states (except the reset state) should be reachable from the reset state, where k is the number of hidden flip-flops. This allows us to attain any required states by first resetting these flip-flops and then executing corresponding inputs to travel along the FSM from the reset state to the desired state. In addition, since k, the number of hidden flip-flops, may be large and the cost of implementing an FSM with large size could be high, it is also important to consider the cost of the FSM design.

We propose to implement an FSM with only $l << k$ flip-flops to control the FSM design and implementation cost. The first l hidden flip-flops will be directly connected to the extra combinational logic to implement an FSM with external input and control signals. The details on how to implement this FSM in hardware and integrate it to the proposed scheme have been discussed in section 5.4.2 and 5.4.3. Here we discuss how to build such FSM from scratch.

Algorithm below shows how to build an FSM such that there are at least F different ways from an initial state to reach all the states in the FSM with exactly M steps, where both F and M are user defined parameters. More specifically, M is an upper bound on the number of steps which indicates the maximum steps we would allow a distinct path to go through in order to reach a desired state. This will be determined by the timing requirement of the scan chain architecture. *F* is a security-specific parameter to obfuscate the hidden input test vector's dependency on the FSM input sequence. For example, if one *l*-bit sequence needs to be generated *t* times to feed into the *k* un-chained flip-flops, and we do not want to use any FSM input sequence more than once (otherwise, when the FSM has the same reset state, the same input sequence will lead to the same values of the *l* flip flops, which will leak the information that certain portions of the hidden input vector are identical.), we must have $F>t$ different ways to generate this *l*-bit sequence.

We create a random 2^lx2^l seed matrix A in line 1, where each entry a_{ij} represents the number of transitions from state i to state j in the seed FSM. It is worth noting that the sum of all the entries in each row, which is the out-degree of each state in the FSM, must equal to 2^s, where *s* is the number of input bits to the FSM. For example, when there are 3 bits for the FSM input, the entries a_{ij}'s should sum up to $2^3=8$. The value of s is considered as a tunable parameter to balance the design hardware overhead (e.g. input pins) and time to generate the desired values for the hidden flip flops.



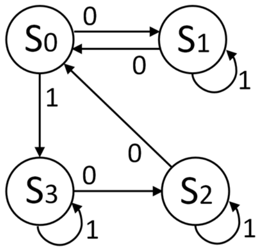
Algorithm 1: Design an FSM to set values for the unchained flip flops within M steps in at least F different ways

```
1: Randomly generate an $2^l * 2^l$ seed matrix A
2: $m = 1$, found = false, $B = I_{2^l * 2^l}$
3: for $m <= M$ do
4:     $B = B * A$
5:     for each entry $B_{0j}$ in the first row do
6:         if $B_{0j} < F$ then
7:             Break
8:         end if
9:     end for
10:    if $j = l$ then
11:        found = true
12:        return $A$ & $m$
13:    end if
14:    $m$++
15: end for
16: if not found then
17:    Go to line 1
18: end if
```

Let the first row of the matrix represent the initial state (or the reset state), to determine how many different paths there are from the initial state to each of the state after *m* steps for the initial FSM, we can simply raise A to the power of *m* (line 4). Thus, by directly checking all the entries in the first row, we can decide whether we have at least F distinct paths from the reset state to all the states in *m* steps (line 6). Once such an FSM is found (line 10), we return the seed matrix A and the number of steps *m* that we need to take in the FSM to reach any state. The FSM with different parameters are implemented and the associated area overhead will be evaluated and reported in Section 6.6.

Figure 20 gives an illustrative example for Algorithm 1 with $l = 2$ unchained flip-flops and $F = 4$ different ways to reach each of the four states $\{S_0, S_1, S_2, S_3\}$ from the initial state $S = S_0$. Without loss of generality, we assume that there is only one input bit in the FSM to be built, i.e. $s = 1$. Figure 20(a) is the seed matrix A and the state transition graph of its corresponding FSM is given in Figure 20(c). For example, the two 1's in the first row indicate that there is one way to go from the initial state S_0 directly to either state S_1 or state S_3. This can be easily verified in Figure 20(c).

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 5 & 3 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 3 & 5 & 4 \end{bmatrix}$$



(a). Seed Matrix A (b). Matrix A^4 (c). An FSM corresponds to Matrix A

Figure 20: An illustrative example of Algorithm 1.

The fourth power A^4 of the seed matrix A is listed in Figure 20(b), where we can see that all the four entries in the first row equal to 4 (indeed, according to Algorithm we just covered, we need all entries to be greater than or equal to 4). This indicates that there are exactly four different ways to reach each of the state from the initial state after 4 steps. For example, from the initial state S_0, on input sequence 0001, the FSM will go through the transition of $S_0 \rightarrow S_1 \rightarrow S_0 \rightarrow S_1 \rightarrow S_1$ and reach state S_1.

Similarly, 4-bit input sequences 0100, 0111, or 1000 will all lead the FSM to state S_1. The security feature of this FSM will be further elaborated in the next chapter.

When we need more and longer paths to reach each state, we can conveniently compute A^m for all the *m*-transition paths. Multiple paths to the same state can enhance security because an adversary will not be able to bind an input sequence from the FSM with the state it will reach. It further increases the difficulty for the adversary to break the FSM and gain control of the hidden flip flops.

6.6 Experimental Results

The goal of our experimentation is to demonstrate the advantages of our approach compared to the traditional partial scan design and to evaluate the practical concerns about our approach. Information about the benchmark circuits and design tools used in the experiments are summarized in Table 8.

Table 8: Benchmarks and Design Tools

Benchmark Circuits	*ISCAS'89*
Synthesize Tool	*Design Compiler*
Scan Chain Insertion	*DFT Compiler*
Test Pattern Generation	*TetraMax ATPG*
FSM Generation	*C++*
FSM Implementation	*Verilog*

Comparison with the traditional partial scan design with ATPG for sequential circuit. For each of the benchmark circuits, we randomly remove 10% of the flip-flops from the full scan chain. For real secure chip design, flip-flops that are designated to store sensitive data should be considered first. In addition to our approach, it is also possible to perform a normal partial scan design for sequential circuits. Table 9 reports the comparison among this approach, our approach, and full scan approach. The total number of flip-flops is shown in column 2. Column 3 denotes the test coverage of normal partial scan after re-running ATPG, where we do see that an average of 0.6% faults are not covered. Column 4 gives the test time increase of partial scan, which on average is almost 3.0X longer than the test time of full scan approach. Finally, the partial scan needs sequential ATPG to generate test patterns, whose run time ranges from 2X to almost 3000X with an average of 627.7X longer than that of the full scan chain as shown in the last two columns. As a comparison, since our scheme applies exactly the same test vectors as the full scan design, we do not need to re-run ATPG and can achieve full test coverage. For test time, because our partial scan is 10% shorter than the full scan and the extra steps to control and observe the un-chained flip-flops are done in parallel during the shifting phase of the partial scan, test time will be reduced by about 10%.

Performance evaluation of the FSM. In our approach, an FSM needs to be implemented to provide multiple paths from the initial state to other states. Figure 21 demonstrates the area overhead, normalized to the area of the benchmark, of the 16-state, 32-state, and 64-state FSM, which has average of 4.41%, 4.98%, and 5.07%, respectively. More importantly, this area overhead of FSM won’t change dramatically when the number of states is fixed, which indicates that such overhead is negligible for large circuit.

Table 9: Performance Comparison of Full and Partial Scan Chains.

Circuit	Number of FFs	Fault Coverage (%)	Test Time Increase (%)	ATPG run time (s)	
				Full	Partial
s9234	145	99.6	211.2	0.4	63.4
s5378	176	99.8	309.8	0.5	1.0
s15850	513	99.2	609.2	2.0	1081.2
s13207	625	98.2	345.8	1.2	3552.9
s38584	1275	99.8	212.4	4.0	57.0
s38417	1564	99.7	104.1	90.7	3534.8
Aver.	/	99.4	298.8	/	2x~2961x

Figure 22 and Figure 23 show the number of transitions, i.e. the length of the input sequence to the FSM, required to reach other states in a 16-state FSM with 1-bit, 2-bit, and 3-bit input of the FSM (Figure 22), and in a 16-state/32-state/64-state FSM (with 2-bit control input, Figure 23), respectively. Clearly, we can see that as one asks for more distinct paths, increasing the number of transitions or the number of input bits to the FSM is more effective than increased the number of states, i.e. the number of flip-flops in the FSM.

6.7 Summary

Directly removing sensitive flip-flops from the scan chain can completely take away attackers' access to these data, which in turn provides high security of the scan design. In this chapter, we have discussed how to gain control of the unchained flip flips through two test vectors manipulation techniques and a novel FSM construction method. To maintain the same full test coverage for the proposed public-private partial scan chains, we put forward a three-step solution, including test vector reordering, reusing and secure generation. The first two steps are implemented at the test vector level which requires no changes to the hardware design. However, they cannot guarantee to enable the test of all required test vector. The secure test vector generation deployed an FSM to provide test engineers with full controllability of the un-chained flip-flops, as a complement to the first two steps. In this way, no information from the hidden flip flips will be leaked. The experimental results show that the FSMs can be implemented in hardware with little overhead. Our partial scan chain outperforms the traditional partial scan chain in terms of test time, fault coverage, and run time in test vector generation. We implement FSMs of various sizes and perform simulation on ISCAS'89 benchmark circuits and the results show that the additional area due to the FSM is negligible even for such small benchmark circuits.

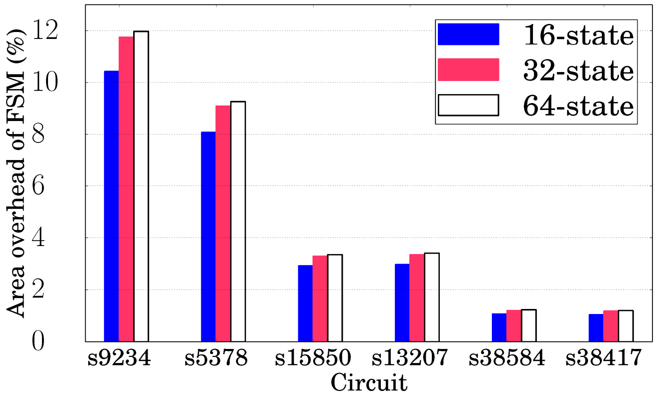


Figure 21: Area overhead of the 16/32/64-state FSMs.

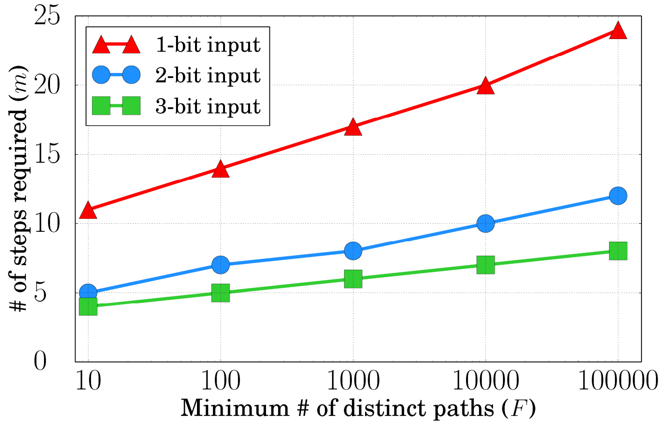


Figure 22: Impact of the number of FSM control input (CI) bits.

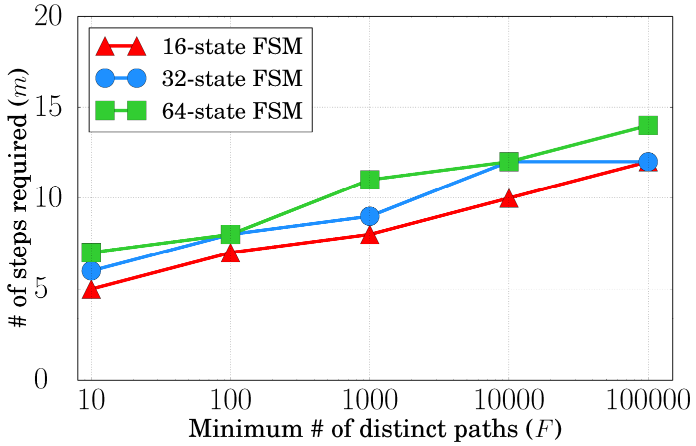


Figure 23: Impact of FSMs with different number of states.

Chapter 7: Observability Challenge: Output Test Vector Obfuscation

In addition to the controllability challenge discussed in Chapter 6, observability challenge and security challenge need to be addressed for the secure partial scan design. In this chapter, we focus on these two challenges. From the discussion in Chapter 6, a test engineer can control the value of the un-chained flip-flops via an FSM. It’s equally important for the test engineer to observe the test responses, including those from the un-chained flip-flops, to complete the chip testing. If the values of the un-chained flip-flops are directly shifted out from some output port similar to the SO port, attackers may also be able to observe them to obtain sensitive information.

In this regard, we propose two lightweight mechanisms, one based on linear feedback shift register (LFSR) and the other one based on configurable physical unclonable function (PUF), with which we can verify the correctness of test responses without leaking any information to the attackers. Both methods achieve this by utilizing the XOR gates to obfuscate the contents of the un-chained flip-flops first and then send them out for test verification. The LFSR based scan output obfuscation approach has been shown in Chapter 5 (Figure 16), where the unchained flip-flops are connected to form a single chain, thus only one output port is needed. The PUF based scheme (Figure 17) works with multiple private chains and requires multiple output ports in addition to the hardware support for PUF circuitry. Note that in both methods, the test output vectors will not be the original test response directly from CUT, they need to be modified for test purpose.

7.1 LFSR based Scan Out for the Un-chained Flip-flops

For our convenience, we redraw the proposed LFSR based scan design in Figure 24, which is part of the structure in Figure 16. The *k* flip-flops D_1' to D_k' are connected to the CUT to load test responses during capture phase. Since they don't play any role in the functionality of the design, we refer to them as *dummy cells*. To read out their contents without losing data security, the bits from the partial scan chain in *Block A* and these dummy cells are XOR-ed before they can be read from the SO port. This gives a way to verify the test response on the k un-chained flip-flops without directly leaking their information. For example, if the test response from the n flip-flops in the public chain is 10101010 and that in the k un-chained flip-flops is 001, the expected test output vector will be $10,101,010 \oplus 01,001,001 = 11, 100, 011$, where the values 001 in the k un-chained flip-flops have been repeated (implemented

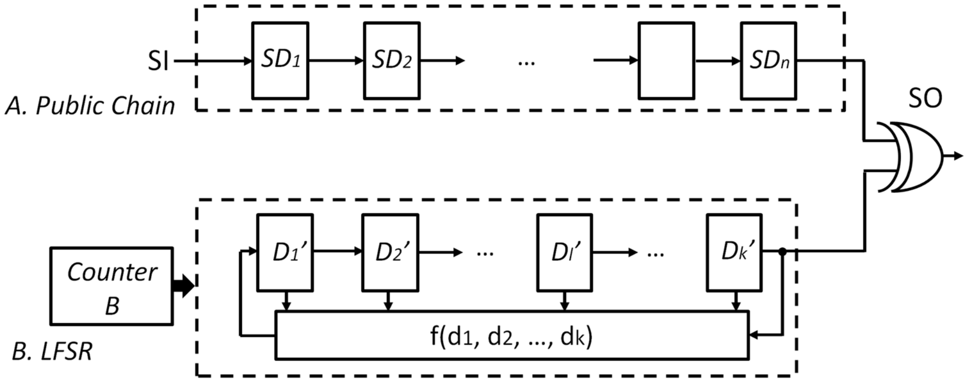


Figure 24: An LFSR based obfuscation of scan output.

as an LFSR).

However, the XOR gate might give false results. For instance, if two bits both with expected value 1 are flipped to 0 simultaneously, the fault will not be detectable since the XOR gate's output would be 0 in both cases. To reduce this false rate, dummy cells are connected in a way to create a Linear Feedback Shift Register (LFSR). The test response is used as the seed. The LFSR is cycling concurrently with the shifting phase in the partial scan chain in *Block A*. After n clock cycles when the shifting phase ends, a Counter B is deployed to clear the contents of the dummy cells to further prevent information leakage. This defines a deterministic way on how the contents in dummy cells will be altered by the LFSR. Hence, the test responses can be predicted, verified, and provided to the test engineer as the expected test output for the given test input vector. During the in-field test, a test engineer can compare the real response with the expected value to check if fault occurs.

Note that there is no guarantee of zero false (a full scan chain cannot guarantee this either because its scan flip-flops may be faulty too). However, with the LFSR running cyclically, the chance of any error in the un-chained flip-flop being undetected will be reduced exponentially because that requires all the corresponding flip-flops in the partial scan chain in *Block A*, those that will be XOR-ed with the faulty un-chained flip-flop to fault simultaneously, which is a rare event. More precisely, we have the following:

> Suppose that an error occurs on a flip-flop in the public partial chain in *Block A* with probability α, on a un-chained flip-flop with probability β, and occurs when flip-flop content is copied from the un-chained flip-flops to the dummy flip-flops

with probability γ, assuming that all the errors occur independently, for each test vector, an error remains undetected if and only if one of the following sets of flip-flops all have errors:

(i) SD_i, SD_{i+k}, SD_{i+2k}, …, and D_i

(ii) SD_i, SD_{i+k}, SD_{i+2k}, …, and D_i'

(iii) $D_{i,}$ and D_i'

The total probability of these cases is

$$p = \alpha^{\lfloor n/k \rfloor}(\beta(1-\gamma)+(1-\beta)\gamma)+(1-\alpha)^{\lfloor n/k \rfloor}\beta\gamma$$

which is

$$\alpha^{\lfloor n/k \rfloor}(\beta+\gamma-2\beta\gamma)+(1-\alpha)^{\lfloor n/k \rfloor}\beta\gamma$$

and can be approximate as

$$\approx \alpha^{\lfloor n/k \rfloor}\beta \approx \alpha^{n/k}$$

where the first approximation comes from the fact that γ, the probability of error when copy from one flip-flop to another with a short wire connection, is extremely low (that is $\gamma<<\alpha, \beta$); and the second approximation comes from the fact that a computation error occurs at any flip-flop is equally likely, i.e. $\alpha=\beta$.

Therefore, an error during test remains unnoticed through the scan structure in Fig. 24 will be extremely low. Furthermore, we see that the use of LFSR plays a major role in this low probability. It reduces the error probability exponentially from α to approximately $\alpha^{n/k}$. We will analyze the security of this scheme in section 7.4.

7.2 Configurable PUF based Output Obfuscation

Another scheme to prevent information leakage from the private chains, is to XOR each bit from the private chain with a random secret bit stream as shown in Figure 17. The random bit stream is recommended to be generated by the hardware to ensure its security. We adopt the highly flexible ring oscillator PUF proposed in [75].

A ring oscillator (RO) consists of odd number of inverters. Figure 25 depicts the architecture that gives us the flexibility to select inverters for the construction of ROs. A multiplexer will be added after each inverter to control whether the inverter will be included in the RO. This is achieved by the selection bit of the multiplexer. If the selection bit is "1", the corresponding inverter will be included in the RO; if the selection bit is "0", the inverter will not be used and the signal will go through the wire to the next inverter (so the corresponding inverter will not be used in the RO).

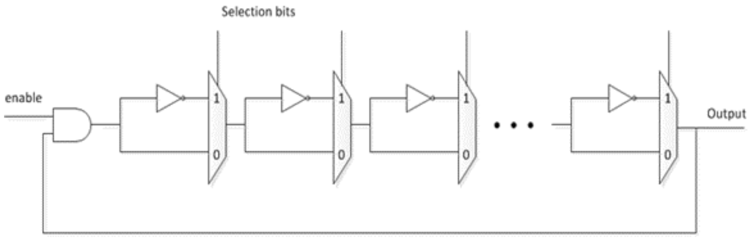


Figure 25: Architecture of the flexible ring oscillator PUF [75].

When the same set of selection bits are given to two such flexible ROs, the delay of the two "identical" ROs they have constructed will be different by normally a very small margin because of fabrication variation. A PUF bit can be created based on such difference. As shown in Figure 17, a vector of such PUF bits is used to XOR each bit from the last flip-flop in the private chains to produce the parallel output PO.

We now describe how the test vectors can be built and provided to the test engineers. A sequence of input values can be provided at the PI port to run the FSM and create the desired input test vector at the flip-flops on the private chains. Then the CUT will be set to the state for test. After test, the flip-flops, including the un-chained private ones, will contain data value that needs to be verified. When these values reach the last flip-flop in each parallel scan chain, they will be used as the selection bit to build two flexible ROs whose delay will be compared to generate a PUF bit. A vector of such random PUF bits will be generated at the same time. We XOR these PUF bits and the flip-flop's contents and shift them out as the output test vector. One advantage of PUF based approach is that the PUF bits will be different from chip to chip, making the test output vectors also distinct for each chip. More about the security of this approach will be analyzed in section 7.4.

There are many hardware design issues to be considered when we implement this approach. One of the most important challenges is timing. Because that the contents of the flip-flops in the parallel private chains need to be XOR-ed with the PUF bits, and these contents are also used as the selection bits for the flexible RO PUF, we need to allow sufficient delay for the PUF bits to be generated before collected the test output vector from the PO ports. Another important feature is the robustness of the PUF bits as it is well-known that RO PUF can be sensitive to operating environment variations such as temperature, operating voltage, humidity, and circuit aging [75]. This is a well-studied topic in PUF literature and we will not elaborate in this dissertation. Interested readers can read some of the survey work on PUF such as [83].

7.3 Protocol for In-Field Test

The test vectors provided to the test engineers can be pre-computed as aforementioned and released to the user together with the chip. However, another important feature for the full testability is whether arbitrary input test vectors can be applied to the circuit for test and whether the test response can be verified to detect chip defects, in particular after the chip is deployed which is known as in-field test. The true challenge of in-field test lies in that it may require new test input vectors, test them on the circuit, and verify whether the test response is as desired. This can be easily achieved when full scan chain is provided and the user can obtain the desired test responses from the system specification or other means. But for our proposed public-private secure scan design, users cannot access the private portion of the scan chain and thus dedicated methods need to be implemented to provide such in-field test capability.

Due to the nature that the circuit itself cannot distinguish attackers from test engineers (user authentication could fail), we believe that such in-field test feature has to be accomplished by on-demand protocols such as the one we will describe below in this section.

Suppose that the user wants to test input vector V, the circuit provider or the service team will create the test input vector V_{pub}, which is identical to the portion of V that belongs to the public partial chain; and V_{pri}, which is the input for the FSM to generate the portion of V that belongs to the private chains. The desired response R will also consist of two parts: R_{pub} and R_{pri}. R_{pub} is identical to those in R, while R_{pri} will be generated following the methods such as those in sections 7.1 and 7.2 with the help of the LFSR and PUF, respectively. In the LFSR-based solution, the test response depends on the seed and the LFSR configuration. In the PUF-based solution, after the circuit is fabricated, the delay information for the inverters on the RO PUFs can be collected and an off-chip emulator can be built to predict the PUF bits. The quadruple (V_{pub}, V_{pri}, R_{pub}, R_{pri}) is then sent to the field for test as the test vector. Once the user receives this quadruple, the in-field test can be conducted as follows: V_{pub} will be fed through SI port, V_{pri} will enter the circuit through the CI/PI port; then after the test, R_{pub} and R_{pri} can be observed directly from the SO and/or PO ports respectively. If any bit of the observed output does not match R_{pub} and R_{pri} as expected, fault or defect is detected. The user can contact the circuit provider or the service team for further diagnosis.

7.4 Security Analysis

Scan based side-channel attacks have been classified into two categories, observability attacks and controllability attacks. In this section, we analyze the efficiency of our approach against these two types of attacks as well as potential attacks specifically targeting our proposed public-private secure partial scan chain.

Assumption on the attackers. Similar to attacking models used in existing literatures, we assume that the attacker 1) is unable to de-package the chip and probe internal signals; but 2) has access to the control pins related to test, such as SI/SO/SE and the FSM control signal in our approach; and 3) can inject input through the test interface and observe the corresponding test output (and match with the expected output).

Observability based attacks. In the scan-based observability attacks, the attacker first applies a stimulus at the primary inputs, then runs the circuit in functional mode and takes snapshots of the circuit's state at any point by shifting out data via SO port. By analyzing the observed results, the attacker may retrieve useful information. Our partial scan based approach prevents this type of attacks by removing certain flip-flops from the scan chain thus their contents will not be available directly through the SO port. Potential attacks to the obfuscated test output are discussed below in the session of "specific attacks targeting the proposed partial scan".

Controllability based attacks. In this popular scan-based attack, the attacker uses the SI port to load specific vectors into the system to control internal states and observes the output responses. To launch this attack on our proposed public private secure scan design, the attacker needs to be able to control the un-chained flip-flops. In our approach, the attacker can inject input sequence from the FSM control input, but he does not know the transitions of the FSM and cannot push specific values to the un-chained flip-flops. Thus any controllability attack will become ineffective. This will be elaborated more next.

Specific attacks targeting the proposed partial scan. As shown above, our approach can mitigate both types of existing scan side channel attacks. But this relies on the assumption that the contents in the un-chained flip-flops cannot be controlled or observed. We now discuss some potential attacks to this assumption.

Replay attack. First, the attacker may analyze the FSM control signal sequences to guess the state of the FSM and *gain controllability to the un-chained flip-flops*. More specifically, if the same FSM control signal sequence is used multiple times, it is ensured that the same state in the FSM will be reached. Thus the attacker may be able to reason and figure out the physical meaning the FSM state and gain more inside information about the FSM and consequently the un-chained private chain. Then, the attacker can use this control signal sequence when the corresponding FSM state is needed to launch any attack. More dangerously, when sufficient number of FSM control signal sequences is replayed and analyzed, it becomes possible for the attacker to reconstruct the functionality of the FSM and hence gain the entire controllability of the private chain.

We have considered this attack in the design of our approach. Remember that in the Algorithm discussed in Section 6.5, we have required that from the reset state of the FSM, there should be sufficient number of paths to reach each state so we will not use the same path to reach the same state more than once. In addition, all states of the FSM can be reached in the same number of transitions or all the FSM control signal sequences will have the same length. This not only simplifies the design of the FSM, but also prevents any information leak as the attacker will not be able to filter out any unreachable states or classify the FSM states based on the length of the control signal sequence. Moreover, contents of the un-chained flip-flops cannot be observed directly, making it difficult for the attacker to determine which state the FSM has reached.

Differential analysis attack. Second, the attacker may try various means to *gain observability of the un-chained flip-flops*. One appealing way is to inject known data to the partial scan chain repeatedly and use the same or similar primary input to launch differential analysis attacks. By fixing the values on the partial scan chain and the primary input, this approach does make the primary output values and the SO output values dependent only on the contents of the un-chained flip-flops. It definitely becomes possible to detect whether the un-chained flip-flops contain the same data or not. However, such detection will never be deterministic in the following sense. If the un-chained flip-flops have the same data, that is, the FSM states are the same, then the test responses are guaranteed to be the same. But the opposite is not true. When the attacker observe the same test output vector, it may not be the case that the test responses are the same (due to the output obfuscation procedure) and more importantly, different starting FSM states can produce the same ending states (which will be part of the test responses and stored in the hidden flip-flops in the private portion of the scan) with different FSM control signal sequences. Such false positive could mislead the attacker. Nevertheless, how much damage this differential analysis attack, and machine learning based attacks to be more general, may cause, for example, to which extent it can infer the values in the un-chained flip-flops, is still under further investigation.

In-field test vulnerability. Finally, it is important to mention that removing those un-chained hidden flip-flops from the scan chain does help to enhance data security. But from the point of view of test, it reduces the testability, in particular in the case when fault occurs on the hidden flip-flops. The well-studied partial scan design for the purpose of reducing test time has provided guidance on which flip-flops can be removed to minimize the loss of testability and how to generate sequential test vectors to provide probabilistic guarantees on the correctness of the test. In-field test could be a challenge.

On the other hand, the proposed approach can support in-field test with new test vectors conveniently with the interaction between chip user (or in-field testing engineer) and the chip builder (or customer service team). To do this, one needs to submit the complete input test vector to the system designer (or the service team). The designer can then use the FSM implementation information in the Algorithm described in Section 6.5 to find an FSM control signal sequence to load the desired values to the hidden flip-flops. Meanwhile, the corresponding test response can be obtained based on the system specification and the obfuscated test output should be computed and provided to the chip user for in-field testing purpose.

This has potential vulnerabilities because an attack can request the target FSM state or some states from which he knows how to reach the target state as the condition for in-field test. When this is the case: the designer will provide the information that the attacker needs to launch the attack. At least it may reveal the information on which flip-flops are hidden and how to set them to be specific values. However, because this in-field test capability requires the direct interaction between the in-field tester and the circuit designer (or service team), several mechanisms such as user authentication, pay per use, and device bounding can be adopted to limit the damage it could cause. This is out of the scope of this dissertation and we will not elaborate in more details. Interested readers can find these mechanism in [67].

7.5 Performance Analysis

The common performance metrics for scan design include test coverage, test time, test power, hardware overhead due to the insertion of scan chain, and in-field test. In the previous sections we have explained that our proposed public-private secure partial scan chain can guarantee both full test coverage and in-field test. We will discuss the rest of the performance metrics next.

Hardware overhead. In addition to the hardware overhead of the traditional scan chain (e.g. the change of D flip-flops to scan cells, various control signals and logic, as well as routing), our proposed scan design requires a hardware implemented FSM and the hardware for test output obfuscation. These extra hardware will incur overhead in terms of chip area and power consumption. Since the power consumption is highly correlated to the size of the circuit (including the flip-flops) and its switching activity, we expect the power overhead will not be an issue during the chip's normal operation and, we will focus on the area overhead only.

Remember that in the design of the private partial scan chains, the size of the FSM remains the same as the number of the chains, not the number of the flip-flops. When the circuit is large, we can increase the length of the private chains without changing the FSM design and implementation. Therefore the area overhead from the hardware implemented FSM will be fixed and can be considered as negligible for large systems in real life. However, a larger FSM or an increased number of paths to each FSM states could bring better security at the cost of larger area overhead.

For the test response obfuscation circuitry, in the LFSR based approach, area overhead comes from the dummy flip-flops, the counter and other logic that are needed to implement the LFSR. This does scale with the number of the dummy flip-flops. In the PUF based method, the area overhead comes from the PUF design. For the highly flexible RO PUF we used in this dissertation, the area cost is related to the number of parallel private chains.

Test time and test power. There is no need to generate any new test vectors to achieve full test coverage in our approach. So the test time will be determined by the time to shift in the input test vectors and shift out the output test vectors, that is, the length of the scan chain. One of the key motivations and advantages of the traditional partial scan chain design is its efficiency in test time and test power because of the shorter chain(s). Our approach is indeed a special case of the traditional partial scan chains. Therefore, it inherits these advantages and will take less time than a full scan chain to test the same set of test vectors. The test power is determined by the simultaneous switches in the flip-flops. In both our design and a traditional full scan design, all the flip-flops may switch at the same time depending on its current content and its current input value. So the parallel private chains will not incur any definite power overhead. On the contrary, this parallel structure gives us the opportunity to optimize test power.

7.6 Summary

In this chapter, we focus on the observability challenge of the proposed public-private partial scan chain design architecture. We illustrate the basic idea of achieving security by the private chains, whose scan input vectors are obtained by a hardware implemented FSM and test outputs are obfuscated by an LFSR or PUF. We demonstrate that our approach not only achieves full test coverage, it also reduces test time and has the potential to save test power too.

Chapter 8: Conclusions and Future Work

8.1 Conclusions

Scan Chain is an industrial standard embedded in hardware design to facilitate chip testing and fault diagnosis. In this dissertation, we study two important security problems related to scan chain: how to utilize scan chain for hardware intellectual property (IP) protection and how to mitigate the increasing scan chain side channel attacks.

First, we take advantage of the availability of both Q and Q' ports on the flip-flops to design practical IP protection methods. We demonstrate the generation of digital fingerprints by selecting different connection styles between adjacent scan cells during the design of scan chain. This method is perhaps the most practical known fingerprinting scheme because fingerprints are created as a post-silicon procedure which will incur little fabrication overhead and test vectors are modified based on the different connection styles which provides a convenient non-intrusive fingerprint detection and verification method. As another example, we show how to build chip identification based on the reconfigurable scan network, an IEEE embedded devices design and test standard. We perform experiments on standard benchmarks to show that our approach has low design overhead. We also conduct security analysis to show that such fingerprints are robust against various attacks.

Second, we argue that the current countermeasures to scan chain side channel attacks are restricted by the requirement of providing a full scan chain with direct SI and SO ports for testing purpose and thus conceptually they will all suffer the vulnerability of leaving the scan chain side channel open to the attackers through the

SI and SO ports as well. Therefore, we propose a novel public-private partial scan chain based approach with the basic idea of removing the flip- flops that store sensitive information from the scan chain. This will eliminate the scan chain side channel, but it also limits chip testing. The key innovation in this dissertation is that we provide a complete solution to achieve full test coverage, including in-field test capability, and security through partial scan chains. We analyze our approach and show that the required hardware has negligible area overhead and gives full test coverage with reduced test time and does not need to re-generate test vectors. The proposed public-private partial scan chain can successfully defeat all the known scan chain side channel attacks and potential attacks specifically designed to target our scan architecture.

Through the example of scan chain, this dissertation demonstrates that conventional design facilities could be reused for security purposes. As attackers continue to exploit hardware design vulnerabilities, the work in this dissertation shows that specific hardware features combined with cryptographic solutions is a promising direction to secure system design.

8.2 Future Work

We briefly mention three lines of future research directions: new scan chain enabled IP protection applications, design concerns and prototyping of the public-private partial scan chains, and more generic hardware security.

IP protection. We describe a practical IP fingerprinting methods using the Q and Q' connection styles. Previously, IP watermarking method has been proposed based on the same idea. We believe that other scan design features such as partial scan chains, reconfigurable scan chains, and the generation of test vectors can all be leveraged for hardware design IP protection. Moreover, the reconfigurability of the RSN as well as the in-field test capability could be used to build chip authentication protocols.

Secure scan design. We give the rationale of behind the public-private partial scan chain and elaborate some design details. As the first work on this novel concept, there are a lot of follow-up research and development directions. (1) Due to the limited resource available to us, we are unable to perform any system prototype or simulation to evaluate the test time, test power, and other performance metrics. This will be mainly an engineer effort that might be very time-consuming, but it is absolutely necessary for the adaption of the proposed secure scan design methodology by industry. (2) Our approach starts with the distinction of flip-flops in the public partial chain and those hidden ones in the private partial chain(s) with the assumption that design is completed. This can create layout and routing problem because the flip-flops belong to the same partial scan chain might be physically apart to meet the functional design requirement. It will be important to consider functional and security requirements simultaneously to optimize the design. One interesting problem to ask is how many and which flip-flops should not be included in the public scan chain. (3) Another assumption in our problem formulation is that we assume that the order of the test vectors can be changed arbitrarily. This may not be true for delay fault detection and diagnosis where a specific set of test vectors have to be applied in a given order. Unfortunately, the approaches we have discussed in this dissertation cannot be used directly and most likely new methods have to be developed. (4) We have discussed some potential attacks to the proposed public-private partial scan, like other security problems, it will be interesting to study how to break this secure scan design and what countermeasures can be used to defeat these new attacks. (5) There are many hardware design issues to be considered when we implement the test output obfuscation methods. One of the most important challenges is timing. For example in the PUF based approach, because that the contents of the flip-flops in the parallel private chains need to be XOR-ed with the PUF bits, and these contents are also used as the selection bits for the flexible RO PUF, we need to allow sufficient delay for the PUF bits to be generated before collected the test output vector from the PO ports. (6) Finally, we believe that more attention should be paid to in-field test. Sophisticated attackers could request specific in-field test cases to collect data and then use learning based attacks to infer the functionality of the FSM that generate the contents of the hidden flip-flops.

Bibliography

[1]. M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing For Digital, Memory and Mixed-Signal VLSI Circuits*, Norwell, MA: Kluwer, 2000.

[2]. International Technology Roadmap for Semiconductors, https://www.semiconductors.org/main/2015_international_technology_roadmap_for_semiconductors_itrs/

[3]. M. Keating and P. Bricaud. "Reuse Methodology Manual for System-on-a-Chip Designs", Kluwer Academic Publishers, 1998.

[4]. Virtual Socket Interface Alliance, "System Chip Letter", Issue 2, Summer 1998.

[5]. G. Qu, M. Potkonjak, *Intellectual Property Protection in VLSI Designs: Theory and Practice*, Springer Science & Business Media, New York, 2003, ISBN: 978- 0-306-48717-0, http://dx.doi.org/10.1007/b105846.

[6]. A.B. Kahng, J. Lach, W.H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, G. Wolfe, “Watermarking techniques for intellectual property protection,” in: Proceedings of the ACM/IEEE Design Automation Conference, 1998.

[7]. A.B. Kahng, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, G. Wolfe, “Robust ip watermarking methodologies for physical design,” in: Proceedings of the ACM/IEEE Design Automation Conference, 1998, pp. 782–787.

[8]. G. Qu, M. Potkonjak, “Analysis of watermarking techniques for graph coloring problem,” in: Proceedings of the IEEE International Conference on ComputerAided Design, 1998, pp. 190–193.

[9]. L. Yuan, R. Pari, G. Qu, “Soft IP protection: watermarking HDL source codes,” in: 6th Information Hiding Workshop, 2004, pp. 224–238.

[10]. A.B. Kahng, J. Lach, W.H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, G. Wolfe, “Constraint-based watermarking techniques for design intellectual property protection,” IEEE Trans. Comput.- Aided Des. 20 (10) (2001) 1236–1252.

[11]. A.L. Oliverira, “Robust techniques for watermarking sequential circuit designs,” in: Proceedings of the ACM/IEEE Design Automation Conference, 1999, pp. 837–842.

[12]. L. Yuan, G. Qu, “Information hiding in finite state machine,” in: 6th Information Hiding Workshop, 2004, pp. 340–354.

[13]. A.K. Jain, L. Yuan, P.R. Pari, G. Qu, “Zero overhead watermarking technique for fpga designs,” in: Proceedings of the Great Lakes Symposium on VLSI, 2003, pp. 147–152.

[14]. L. Yuan, G. Qu, A. Srivastava, “VLSI CAD tool protection by birthmarking design solutions,” in: Proceedings of the Great Lakes Symposium on VLSI, 2005, pp. 341–344.

[15]. A.T. Abdel-Hamid, S. Tahar, M. Aboulhamid, “A survey on IP watermarking techniques”, Design Automation Embedded System 9 (2004) 211–227.

[16]. A.E. Caldwell, H.-J. Choi, A.B. Kahng, S. Mantik, M. Potkonjak, G. Qu, J.L. Wong, “Effective Iterative Techniques for Fingerprinting Design IP,” IEEE Trans. Comput. Aided Des. 23 (2) (2004) 208¨C215.

[17]. G. Qu, M. Potkonjak, “Fingerprinting intellectual property using constraint addition”, in Design Automation Conference, 2000.

[18]. F. Koushanfar, G. Qu, and M. Potkonjak. “Intellectual Property Metering”, 4th Information Hiding Workshop (IHW’01), pp. 87–102, LNCS Vol. 2137, Springer-Verlag, April 2001.

[19]. K. Lofstrom, W. R. Daasch, D. Taylor, IC identification circuit using device mismatch, in: Proceedings of the IEEE Solid State Circuits Conference, 2000, pp. 372–373.

[20]. U. Rührmair, S. Devadas, F. Koushanfar, Security based on physical unclonability and disorder, in: M. Tehranipoor, C. Wang (Eds.), Introduction to Hardware Security and Trust, Springer, New York, 2012, pp. 65–102.

[21]. S. Gupta, T. Vaish, and S. Chattopadhyay, “Flip-flop chaining architecture for power-efficient scan during test application,” in Proc. Asia Test Symp., Calcutta, India, Dec. 2005, pp. 410- 413.

[22]. A. Cui, G. Qu, and Y. Zhang, “Dynamic Watermarking on Scan Design for Hard IP Protection with Ultra-low Overhead”, IEEE Transactions on Information Forensics & Security (TIFS), Vol. 10, No. 11, pp. 2298-2313, November 2015. DOI: 10.1109/TIFS.2015.2455338.

[23]. J. Katz and Y. Lindell, Introduction to Modern Cryptography, Chapman and Hall/CRC, Boca Raton, FL, 2008.

[24]. P. Kocher, “Timing attacks on implementations of diffie-bellman, RSA, DSS, and other systems,” pp.104-113, CRYPTO, 1996.

[25]. P. Kocher, I. Jaffe and B. Jun, “Differential Power Analysis”, pp.388-397, CRYPTO, 1999.

[26]. K. Gandolfi and C. Mourtel and F. Olivier, “Electromagnetic Analysis: Concrete Results”, in Workshop on Cryptographic Hardware and Embedded Systems (CHES), volume 2162. Springer-Verlag, May 2001.

[27]. E. Biham and A. Shamir, “Differential fault analysis of secret key cryptosystems”, in Proceedings of Advances in Cryptology–CRYPTO’97, Springer-Verlag, 1997, pp. 513-525.

[28]. B. Yang, K. Wu, and R. Karri, “Scan-based side-channel attack on dedicated hardware implementations of data encryption standard,” in *Proc. of the Int. Test Conf.* (*ITC*), Washington DC, USA, Oct. 2004, pp. 339-344.

[29]. H. Kodera, M. Yanagisawa, and N. Togawa, “Scan-based attack against DES cryptosystems using scan signatures,” in Proc. Asia Pacific Conf. Cir. Syst. (APCCAS), Kaohsiung, Taiwan, Dec. 2012, pp. 2-5

[30]. B. Yang, K. Wu, and R. Karri, “Secure scan: A design-for-test architecture for crypto chips,” *IEEE Trans. CAD Integr. Cir. Syst.*, vol. 25, no. 10, Oct. 2006, pp. 2287-2293.

[31]. R. Nara, N. Togawa, M. Yanagisawa, and T. Ohtsuki, “Scan-based attack against elliptic curve cryptosystems,” in *Proc. of Asia and South Pacific Des. Autom. Conf. (ASP-DAC)*, Taipei, Taiwan, Jan. 2010, pp. 407-412.

[32]. R. Nara, K. Satoh, M. Yanagisawa, and N. Togawa, “Scan-based side-channel attack against RSA cryptosystems using scan signatures,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no.12, Dec. 2010, pp. 2481-2489.

[33]. C. Dunbar and G. Qu, “Designing Trusted Embedded Systems from Finite State Machines”, *ACM Transactions on Embedded Computing Systems (TECS)* Vol. 13, No. 5s, September 2014.

[34]. C. Dunbar and G. Qu, “Towards Building Trusted Systems: Vulnerabilities, Threats, and Mitigation Techniques”, in Secure System Design and Trustable Computing, pp. 301-328, Springer, ISBN 978-3-319-14971-4, 2016.

[35]. R. Baranowski, M. A. Kochte and H. J. Wunderlich, "Modeling, verification and pattern generation for reconfigurable scan networks," *2012 IEEE International Test Conference*, Anaheim, CA, 2012, pp. 1-9.

[36]. "IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Standard 1149.1-2013", 2013.

[37]. IJTAG, "IJTAG - IEEE P1687," Mar. 2012. [Online]. Available: http://grouper.ieee.org/groups/1687.

[38]. R. Baranowski, M. A. Kochte and H. J. Wunderlich, "Fine-Grained Access Management in Reconfigurable Scan Networks," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 937-946, June 2015.

[39]. J. Lach, W. H. Mangione-Smith and M. Potkonjak, "FPGA Fingerprinting Techniques for Protecting Intellectual Property", Proceedings of CI-CC, 1998.

[40]. C. Dunbar and G. Qu, "Satisfiability Don't Care Condition Based Circuit Fingerprinting Techniques", in Proceedings of 20th Asia and South Pacific Design Automation Conference, Chiba, Japan, 2015.

[41]. C. Dunbar and G. Qu, "A Practical Circuit Fingerprinting Method Utilizing Observability Don't Care Conditions", Design Automation Conference (DAC'15), June 2015.

[42]. H. J. Patel, J. W. Crouch, Y. C. Kim and T. C. Kim, "Creating a unique digital fingerprint using existing combinational logic," in Circuits and Systems, IEEE International Symposium on, Taipei, 2009.

[43]. Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprinting," in Hardware-Oriented Security and Trust, IEEE International Workshop on, Anaheim, CA, 2008.

[44]. G. Qu, C. Dunbar, X. Chen, A. Cui, "Digital Fingerprint: A Practical Hardware Security Primitive" in "Digital Fingerprinting", pp 89-114, Springer New York, ISBN 978-1-4939-6601-1, 2016.

[45]. S. Gupta, T. Vaish, and S. Chattopadhyay, "Flip-flop chaining architecture for power-effcient scan during test application", in Proc. Asia Test Symp., Kolkata, India, Dec. 2005, pp. 410-413.

[46]. A. Cui, G. Qu and Y. Zhang, "Ultra-Low Overhead Dynamic Watermarking on Scan Design for Hard IP Protection," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 11, pp. 2298-2313, Nov. 2015. doi: 10.1109/TIFS.2015.2455338.

[47]. Christopher Clearfield, “Why the FTC Can't Regulate the Internet Of Things”, Forbes, 18 September 2013.

[48]. G. Qu and L. Yuan, “Design Things for the Internet of Things – An EDA Perspective”, IEEE/ACM International Conference on Computer Aided Design (ICCAD'14), November 2014.

[49]. Joseph Steinberg. "These Devices May Be Spying On You (Even In Your Own Home)". Forbes. 27 January 2014.

[50]. C. Dunbar and G. Qu, “A DTN Routing Protocol for Vehicle Location Information Protection”, Military Communications Conference (Milcom’14), October 2014.

[51]. X. Chen, G. Qu, A. Cui and C. Dunbar, "Scan chain based IP fingerprint and identification," *2017 18th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, 2017, pp. 264-270.

[52]. T. Wang, X. Cui et al, “A Novel Circuit Authentication Scheme based on Partial Polymorphic Gates” *in Proceedings 22th Asia and South Pacific Design Automation Conference (ASP-DAC),* 2017.

[53]. S. Narayanan and M. A. Breuer, "Reconfigurable scan chains: A novel approach to reduce test application time," *Proceedings of 1993 International Conference on Computer Aided Design (ICCAD)*, Santa Clara, CA, USA, 1993, pp. 710-715.

[54]. R. Baranowski, M. A. Kochte and H. J. Wunderlich, "Modeling, verification and pattern generation for reconfigurable scan networks," *2012 IEEE International Test Conference*, Anaheim, CA, 2012, pp. 1-9.

[55]. E. J. Marinissen, V. Iyengar and K. Chakrabarty, "A set of benchmarks for modular testing of SOCs," *Proceedings. International Test Conference*, 2002, pp. 519-528.

[56]. F. G. Zadegan, U. Ingelsson, G. Carlsson and E. Larsson, "Design automation for IEEE P1687," *2011 Design, Automation & Test in Europe*, Grenoble, 2011, pp. 1-6.

[57]. P. Kocher, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, “Spectre attacks: Exploiting speculative execution”, ArXiv e-prints, Jan. 2018.

[58]. M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg, “Meltdown”, ArXiv e-prints, Jan. 2018.

[59]. A. Tang, S. Sethumadhavan, and S. Stolfo, “CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management”, in USENIX Security, 2017.

[60]. G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, “Dolphin attack: Inaudible voice commands”, in Proc. of ACM Conference on Computer and Communications Security (CCS), 2017.

[61]. K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, “A2: Analog malicious hardware”, in Proc. of IEEE Symposium on Security and Privacy (S&P), May 2016, pp. 18–37.

[62]. K. P. Parker. 1998. *The Boundary Scan Handbook*. Boston, MA: Kluwer Academic Publishers.

[63]. J. Da Rolt, G. Di Natale, M. Flottes, and B. Rouzeyre, “A novel differential scan attack on advanced DFT structures”, ACM Transactions on Design Automation of Electronic Systems, 2013, vol. 18, no. 4, article no. 58.

[64]. S. S. Ali, O. Sinanoglu, S. M. Saeed, and R. Karri, “New scan attacks against state-of-the-art countermeasures and DFT”, in Proc. of IEEE International Workshop Hardware-Oriented Security Trust (HOST), pp. 142-147.

[65]. D. Hely, F. Bancel, M.-L. Flottes, and B. Rouzeyre, “Test control for secure scan designs”, in Proc. of European Test Symposium (ETS), Tallinn, Estonia. 2005, pp. 190–195.

[66]. S. S. Ali, S. M. Saeed, O. Sinanoglu, and R. Karri, “Scan attack in presence of mode-reset countermeasure”, in Proc. of IEEE 19th Int. On-Line Test. Symp. (IOLTS), Santa Clara, CA, USA, 2013, pp. 230–231.

[67]. J. Da Rolt et al., “Test versus security: Past and present”, IEEE Trans. Emerg. Topics Comput., vol. 2, no. 1, 2014, pp. 50–62.

[68]. M. G. Kuhn and O. Kommerling, "Design principles for tamper resistant smart-card processors", in USENIX Workshop on Smart-card Technology Proceedings, Chicago Illinois, 1999, pp 9-20.

[69]. D. Hély, F. Bancel, M. L. Flottes, B. Rouzeyre, M. Renovell, and N. Bérard, "Scan design and secure chip", in IEEE International On-Line Testing Symposium, Funchal, Portugal, 2004, pp. 219–226.

[70]. J. Lee, M. Tehranipoor, C. Patel and J. Plusquellic, "Securing scan design using lock and key technique", in Proceedings of the 20th IEEE International Symposium of Defect and Fault Tolerance in VLSI Systems, 2005, p. 51–62.

[71]. J. Lee, M. Tehranipoorand and J. Plusquellic, "A Low-Cost Solution for Protecting IPs Against Side-Channel Scan-Based Attacks", in Proc. of VLSI Test Symposium, 2006.

[72]. Y. Liu, K. Wu, and R. Karri, "Scan-based attacks on linear feedback shift register based stream ciphers", ACM Trans. Des. Autom. Electron. Syst., 2011, vol. 16, no. 2, pp. 115.

[73]. J. D. Rolt, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "Thwarting scan-based attacks on secure-ICs with on-chip comparison", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. PP, no. 99, p. 1.

[74]. J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "A smart test controller for scan chains in secure circuits", in Proc. of IEEE 19th IOLTS, 2013, pp. 228-229.

[75]. M. Gao, K. Lai, and G. Qu, "A highly flexible ring oscillator PUF", in Proc. of 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014.

[76]. R. Gupta, R. Gupta and M. A. Breuer, "The Ballast methodology for structured partial scan design," in IEEE Transactions on Computers, vol. 39, no. 4, pp. 538-544, Apr 1990. doi: 10.1109/12.54846.

[77]. V. Chickermane and J. H. Patel, "An optimization based approach to the partial scan design problem," Proceedings. International Test Conference 1990, Washington, DC, 1990, pp. 377-386. doi: 10.1109/TEST.1990.114045.

[78]. V. Chickermane and J. H. Patel, "A fault oriented partial scan design approach," 1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers, Santa Clara, CA, USA, 1991, pp. 400-403. doi: 10.1109/ICCAD.1991.185287.

[79]. R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable scan networks: Modeling, verification, and optimal pattern generation," ACM Trans. Design Automation Electronic System, vol. 20, no. 2, pp. 30:1–30:27, Mar. 2015. [Online]. Available: http://doi.acm.org/10.1145/2699863.

[80]. V. D. Agrawal, K. Cheng, D. Johnson, and T. Lin, "Designing Circuits with Partial Scan," IEEE Design and Test of Computers, pp. 8-15, April 1988.

[81]. G. Sengar, D. Mukhopadhyay, and D. R. Chowdhury, ''Secured flipped scan-chain model for crypto-architecture,'' IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 26, no. 11, pp. 2080–2084, Nov. 2007.

[82]. https://en.wikipedia.org/wiki/Hamiltonian_path

[83]. J. Zhang, G. Qu, Y. Lv, and Q. Zhou, "A survey on silicon PUFs and recent advances in ring oscillator PUFs," J. Comput. Sci. Technol., vol. 29, no. 4, pp. 664–678, 2014.