4-10-2013

On-chip Structures and Techniques to Improve the Security, Trustworthiness and Reliability of Integrated Circuits

xuehui zhang
ECE Department, Uconn, xuehui.zhang@engr.uconn.edu

On-chip Structures and Techniques to improve the Security, Trustworthiness, and Reliability of Integrated Circuits

Xuehui Zhang, Ph.D.

University of Connecticut, 2013

Due to the globalization of the semiconductor design and fabrication process, integrated circuits (ICs) are becoming increasingly vulnerable to malicious activities. There are two major problems that impact the security, trustworthiness, and reliability of ICs used in military, financial, or other critical applications: (*i*) Malicious inclusions and alterations, known as hardware Trojans, could be easily inserted into intellectual properties (IPs) or ICs by an untrusted process. These hardware Trojans may leak confidential information to an adversary or potentially disable part or all of an IC at a specific target time in the field. Techniques need to be developed to identify these hardware Trojans to prevent the potential damages. (*ii*) The number of circuit-related counterfeiting incidents reported by component manufacturers increases significantly over the past few years and recycled ICs con-

tribute major percentages of the total reported counterfeiting incidents. These recycled ICs enter the market when electronic "recyclers" divert scrapped circuit boards away from their designated place of disposal for the purposes of removing and reselling the ICs on those boards. Since these recycled ICs have been used in the field before, the performance of such ICs has been degraded by aging effects and harsh recycling process. In this thesis, to address the above two problems, we developed several light-weight on-chip structures and techniques to improve the security and reliability of ICs. These structures and techniques include (*i*) a verification-based flow to detect hardware Trojans in IPs, (*ii*) an on-chip ring oscillator network (RON) acting as power monitors to detect hardware Trojans in ICs, (*iii*) a novel technique combining the improved RON with transient current to improve the sensitivity of the RON for hardware Trojan detection, (*iv*) three light-weight sensors recording the usage time to identify recycled ICs, (*v*) a path-delay fingerprinting flow with zero area overhead to identify recycled ICs, and (*vi*) two true random number generators (TRNGs) to generate sequences with high randomness, which are widely used for secure data communication and storage. The simulation results and implementation analysis demonstrate the effectiveness of our proposed techniques.

On-chip Structures and Techniques to improve the Security, Trustworthiness, and Reliability of Integrated Circuits

Xuehui Zhang

B.S., Beihang University, Beijing, China, 2006

M.S., Beihang University, Beijing, China, 2009

A Dissertation

Submitted in Partial Fullfilment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2013

APPROVAL PAGE

Doctor of Philosophy Dissertation

On-chip Structures and Techniques to improve the Security, Trustworthiness, and Reliability of Integrated Circuits

Presented by

Xuehui Zhang,

Major Advisor

Mohammad Tehranipoor

Associate Advisor

John Chandy

Associate Advisor

Lei Wang

University of Connecticut

2013

This thesis is dedicated to my family.

ACKNOWLEDGEMENTS

I would like to thank Prof. Mohammad Tehranipoor for advising me during my Ph.D. program. His support and encouragement add considerably to my graduation.

I would also like to thank Profs. John Chandy, Lei Wang, Jerry Shi, and Omer Khan for joining my Advisory Committees. They gave me lots of valuable suggestions for this dissertation.

Special thanks to all my labmates for the cooperation. The debates and exchanges of knowledge enriched my experience.

Sincere thanks to my family for their love and support throughout my entire life. They make my life beautiful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Background and Motivation

An integrated circuit (IC) is an electronic circuit on one small plate ("chip") of semiconductor material. Several billion transistors or other electronic components could be integrated into one IC, which is widely used in virtually all electronic equipment, such as computers, mobile phones, other digital home appliances in modern society, etc. According to reports from US Department of Defense [1] and documents from US Department of Commerce [2], ICs are becoming increasingly vulnerable to malicious activities. Hardware Trojan insertion and recycled ICs are major problems related to the security and reliability of ICs in the recent several years.

1.1.1 Hardware Trojan

Hardware Trojans are malicious function that can be inserted into a circuit from register transfer level (RTL) design to fabrication process by untrusted foundry

[19]. The survey presented in [19] discussed the seriousness of hardware Trojan problem. An IC design and fabrication process (shown in Figure 1.1) contains four major steps: RTL design (involving specification, IP blocks, and designers), physical design (involving CAD tools, models, and designers), fabrication (involving mask generation and lithography), and manufacturing test (involving wafer probe and packaging). In the ASIC RTL and physical design process, commercial CAD tools are generally considered to be trusted since they are commonly developed by trusted companies such as Synopsys, Cadence Design Systems, and Mentor Graphics. However, the IP blocks, models, and standard cells used by the designer during the design process and by the foundry during the postdesign processes are considered untrusted [19] due to the globalization of the semiconductor industry. For example, IP blocks provided by third party IP (3PIP) vendors oversea, who may be untrusted, could contains a well hidden Trojan that is designed to be activated under specific conditions. Moreover, the fabrication step might also be considered untrusted, because an attacker could insert a Trojan into the IC mask.

Hardware Trojans inserted by those untrusted designers and foundries feature different physical, activation, and functional characteristics. However, they are typically composed of trigger and payload, shown in Figure 1.2. The trigger inputs (T_1, T_2, ..., T_k) come from various nets in the circuit. The payload taps signals from the original (Trojan-free) circuit and the output of the trigger. Since

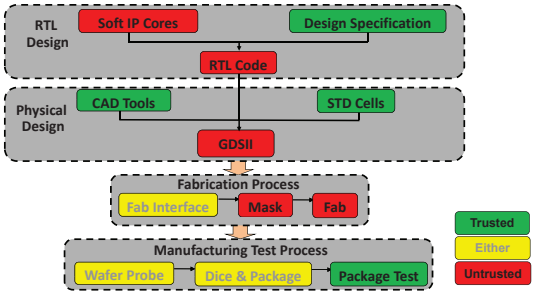


Fig. 1.1: Today's typical IC design and fabrication flow.

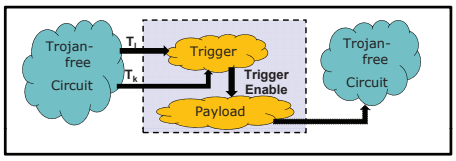


Fig. 1.2: Trojan structure.

the trigger is expected to be activated under rare conditions, the payload output maintains the same value as Trojan-free circuit most of the time. However, when the trigger is activated, the payload output will inject an erroneous value into the circuit and cause an error at the output. In addition, some Trojans may not necessarily impact the function of the circuiti, but rather execute a code that is designed to perform a specific function such as sending or receiving information to or from adversary from the outside. For instance, a hardware Trojan inserted into an microprocessor 8051 could leak the secure identification number of the design through a peripheral equipment [70].

These Hardware Trojans could destroy the fabricated chips, cause erroneous behavior in the field, or provide adversary with access to secret keys in secure

hardware. Since 2007, researchers have been studying hardware trojan detection techniques to prevent the potential damages [11]. The typical strategy is to activate hardware Trojans by applying enumerative inputs to the circuit. Then the functional behavior of the Trojan-inserted circuit will be different from that of a Trojan-free circuit. However, it is very difficult to activate most hardware Trojans since they feature different physical, activation, and functional characteristics. Moreover, exhaustive evaluation for designs with millions of gates involves lots of time and effort, which may be not economically viable. One of the alternative methods for hardware Trojan detection is to use side-channel information. With hardware Trojans, the side-channel signature of Trojan-inserted ICs, such as power consumption, path delay, and leakage current, will be beyond that of Trojan-free ICs. The problem for this type of technique is that the impact of hardware Trojans on side-channle information could be masked by that caused by process variations. Environmental variations could also make this technique less effective. Another problem with the side-channel analysis technique is that it is very difficult to analyze the impact of Trojans inserted to IPs on side-channel information since most IPs are provided in RTL code.

Due to these disadvantages of each existing technique, new techniques must be developed to secure electrics systems, especially those used in critical applications. One of our objectives in this thesis is to develop techniques to detect hardware Trojans effectively. In order to achieve our objectives, we run a case

study to further understand hardware Trojans inserted into IPs in this thesis. Then we analyze impact of hardware Trojans on a circuit. Finally, we propose a verification-based flow to detect hardware Trojans in IPs. Also, we design a novel on-chip structure to generate fingerprints, which eliminate the impact of process and environmental variations to detect hardware Trojans in ICs.

1.1.2 Recycled ICs

The counterfeiting of ICs is another major issue that impacts the security of a wide variety of electronic systems. A counterfeit component is defined as an electronic part that is not genuine because it: (*i*) is an unauthorized copy; (*ii*) does not conform to original component manufacturers design, model, and/or performance; (*iii*) is not produced by the original component manufacturers or is produced by unauthorized contractors; (*iv*) is an off-specification, defective, or used original component manufacturers product sold as "new" or working; (*v*) has incorrect or false markings and/or documentation [2].

The Office of Technology Evaluation, part of the U.S. Department of Commerce, reported over 10,000 incidents involving the re-sale of used or defective ICs from 2005 to 2008 alone which is much more than other types of counterfeits [2] (shown in Figure 1.3). From the figure, we can see that the number of reported incidents of used ICs being sold as new or remarked as higher grade is much more than other types of counterfeits. Note that the term *recycled IC* is used to denote

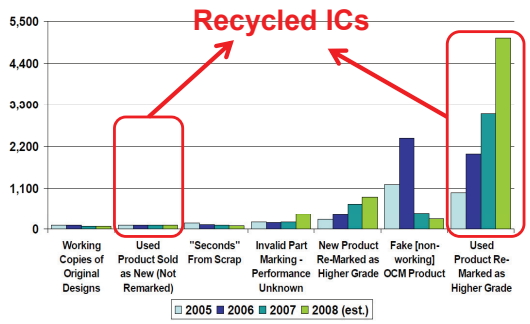


Fig. 1.3: Counterfeit incidents by type of problem for microcircuits from 2005 to 2008.

used ICs being sold as new or remarked as higher grades in this thesis and the terms *unused/new IC* represents the ICs that are brand new. In 2008, Business Week published an investigation that traced recycled ICs found in U.S. military supplies back to their sources [3]. It is reported in [4] that used or defective products account for 80 to 90% of all counterfeits being sold worldwide. With such an estimate on the percentage of used ICs being sold, and the numbers relating to semiconductor sales and counterfeiting in general presented in [5], it could be possible that the intentional sale of used or defective chips in the semiconductor market could have accounted for about $15 billion USD of all semiconductor sales in 2008 alone. Note that this number could actually be much larger since many of the counterfeit ICs go undetected and are being used in systems today. In addition, from Figure 1.3, we can see that the trends, shown in [2], suggest that this number is only going to increase over time.

These used or defective ICs enter the market when electronic "recyclers"

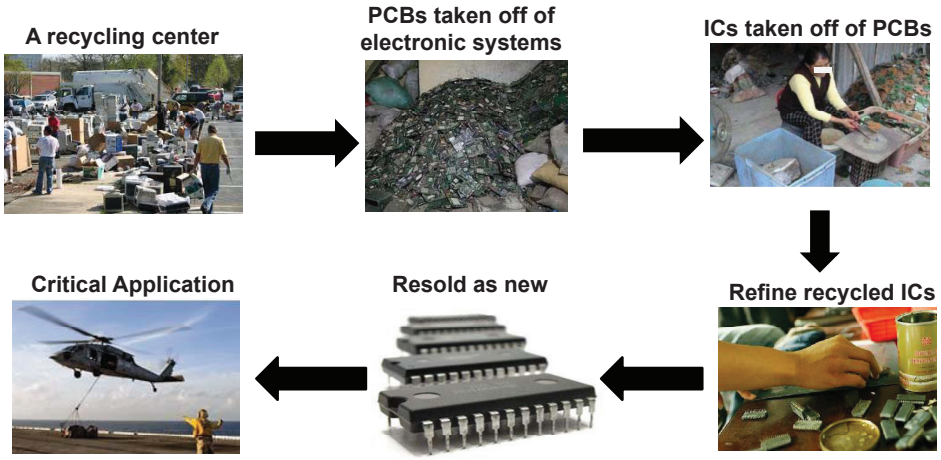


Fig. 1.4: Recycled ICs Process.

divert scrapped circuit boards away from their designated place of disposal for the purposes of removing and reselling the ICs on those boards. The detailed recycling process is shown in Figure 1.4. After carefully cleaning, those used ICs look like new and could be re-used in critical applications. It is vital to prevent recycled ICs from entering critical infrastructure, aerospace, medical, and defense supply chains since they will fail sooner and less predictably than new chips.

Since the recycling process usually involves a high temperature environment to remove ICs from boards, there are several security issues associated with these ICs: (*i*) a used IC can act as a ticking time bomb [6] since it does not meet the specification of the unused (new) ICs; (*ii*) an adversary can include additional die on top of the recycled die carrying a back-door attack, sabotaging circuit functionality under certain conditions, or causing denial of service [7]. Therefore, it is vital that we prevent these recycled ICs from entering critical infrastructures,

aerospace, medical, and defense supply chains.

1.2 Previous Works and Their Limitations

1.2.1 Hardware Trojan Detection Techniques

Hardware Trojans are extremely difficult to be detected since their impact on the functionality is not always observable. A carefully designed Trojan has small number of gates placed on different locations in the design, the change on the circuit's parameters is almost negligible. Furthermore, the adversary can easily defeat the Trojan detection strategies that target the testing of statistically unlikely circuit states because the observation points of these types of approaches will be limited to circuit states defined by only small number of nodes. The existing design for testability methods, inserting a scan chain to the circuit, are not very helpful in detecting Trojans. Automatic test pattern generation (ATPG) tools can generate structural patterns to detect certain faults, i.e., stuck-at, path delay, and transition delay. Since hardware Trojans cannot be modeled by the normal flow of ATPG, the fault locations due to these Trojans cannot be detected. Finally, hardware Trojans may be decomposed into different categories based on: structure, function, distribution, parameters, and size. Therefore, it is fairly impossible to model all possibilities and utilize these models to identify Trojans in a design by comparison.

Recently, several approaches have been proposed to identify Trojan-inserted

ICs since 2007 [19]. These detection methods can be classified into three categories: side-channel signal analysis, Trojan activation, and monitoring architectures. Side-channel signal analysis has been utilized to detect hardware Trojans by measuring circuit parameters. Examples of this type include: power-based analysis [11] [12] [19], current analysis [15], and delay-based analysis [16] [17].

The authors in [11] were the first to use power signatures to measure the power contribution of Trojans by applying random patterns, and observing the power consumption. A Trojan-free IC is supplied as a golden IC to generate a power signature, which will be used for comparison against target ICs in the paper. Path delay information is collected to build a series of fingerprints from Trojan-free circuit [16]. For all the methods using side-channel information, the basic idea is shown in Figure 1.5. Different side-channel information will be collected from the Trojan-free ICs and the circuit under authentication (CUT). If the side-channel information of the CUT is beyond the signature of those Trojan-free ICs, with a high probability that the CUT is Trojan-inserted. Otherwise, it could be Trojan-free. Side-channel analysis methods are effective for Trojans that have a significant effect on power, current, and delay. However, there are many variables which affect these parameters, such as measurement noise, process variations, and environmental variations, and may mask Trojan's contribution to the side-channel signals.

The second Trojan detection method is to use Hardware Trojan activation

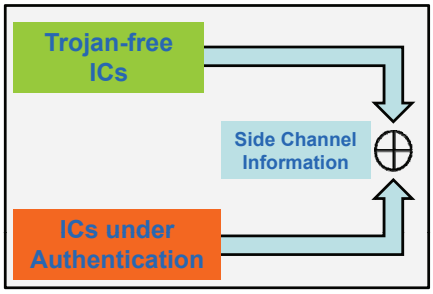


Fig. 1.5: The basic idea of methods using side-channel information.

strategies, which are proposed in [22] [23] [24] [25]. When hardware Trojans are activated, with a high probability the malicious functionality could be monitored by checking the output of the design. For example, a dummy scan flip-flop insertion procedure is aimed at decreasing the potential Trojan activation time in [25]. However, the time required to activate (i.e. launch the malicious function of) a hardware Trojan is a major concern from an authentication standpoint. In addition, a skilled adversary will design Trojans which activate under exceptionally rare conditions (e.g. a specific 32-bit instruction which would be one of $\approx 2^{32} = 4.295 \times 10^9$ possible combinations). Therefore, the disadvantage of any Trojan activation method lays in the difficulty of activating Trojans that are designed to be enabled under very specific conditions, and an inability to detect the non-functional Trojans listed in [18].

Monitoring structures have been also proposed to prevent the damages caused by Trojans. For instance, reconfigurable Design-For-Enabling-Security (DEFENSE) logic was embedded into functional designs to implement real-time

security monitors in [27]. The DEFENSE infrastructure consists of distributed instruments that can be repeatedly configured to dynamically implement different security checks to detect unexpected or illegal behavior. In [26], the Trojan-resistant SoC bus architecture tries to prevent untrusted access to the secure memory or data. Once the bus has detected unexpected behavior, it will block the attacking packet and report it to the system, which will reset and initialize necessary registers. However, with millions of nets in the circuit, it is impossible to monitor all of them.

New techniques must be developed to detect those hardware Trojans. Our main objectives in developing new techniques are: (*i*) these techniques must be effective to detect hardware Trojans composed of even a small number of gates, (*ii*) the impact of process and environmental variations on these techniques must be minimal, (*iii*) these techniques must be resilient to attacks, and (*iv*) the measurement and identification process must be done by using time efficient, low-cost, and user friendly equipment.

1.2.2 Recycled ICs Identification

In the past several years, several techniques related to recycled ICs detection have been developed. Physical unclonable functions (PUFs) implement challenge and response authentication for IC identification [48] [49] [50] [51] [52]. For each physical stimulus, the circuit will react in an unpredictable way due to the complex

interaction of the stimulus with the physical structure of the PUF and the inherent process variations. As the physical variations for each IC are unique, a distinct ID can be obtained for each IC through the PUF. Techniques to protect ICs against counterfeiting via active and passive authentication and identification (also known as hardware metering) have been proposed in [35] [53] [54]. Metering techniques attempt to ensure that over-production of ICs will be prohibited. The above approaches are effective at authenticating ICs but not at identifying recycled ICs since they are expected to have the same IDs as the unused ICs.

The computer-aided design and reliability research community has also seen extensive research on analyzing the aging of ICs. In particular, ring oscillator based reliability analysis has become a common practice. For instance, a silicon odometer has been proposed to monitor different types of aging effects [55] [56]; however, the objective was to improve the reliability of ICs, not to identify the recycled ICs. Such sensors will be ineffective if they are to be used in detecting recycled ICs due to the presence of process and environmental variations.

On the other hand, since recycled ICs have the original appearance, functionality, and markings as the devices they are meant to mimic, even the best visual inspection techniques will have difficulty identifying these ICs with certainty [8]. Physical tests, described in [9], are often used to identify recycled ICs by visual inspection, blacktop testing, scanning electron microscopy (SEM), scanning acoustic microscopy (SAM), x-ray imaging, x-ray fluorescence, fourier transform

infrared (FTIR) spectroscopy, etc. These methods can efficiently detect recycled ICs with mechanical defects, such as defects in package, lead, bond wires, die etc. However, they cannot detect recycled ICs without these physical defects. Moreover, all the ICs under physical test cannot be verified as most of these tests are based on sampling. On the other hand, electrical detection methods can be applied to all the ICs under test. SAE AS5553 [9] incorporates some electrical tests such as DC curve trace, full DC test, key (AC, switching, functional) and full functional tests at ambient temperature and over temperature in their detection procedure. However, the applicability of these tests to today's complex ICs (microprocessors, memories, programmable logic devices, ASICs, etc.) is a major concern. Detection of recycled ICs using electrical tests has not yet been verified completely and there are currently no available documents to guide recycled ICs detection using electrical tests.

Therefore, new techniques need to be developed to help measure recycled ICs' specifications and effectively detect them if they have already been used in the field even for a short period of time. The major difference between recycled ICs and unused ICs is that recycled ICs have already been used and experienced aging. We will proposed our recycled ICs identification methods based on the performance degradation caused by aging effects in this thesis. The usage time of ICs could be also reported by our techniques to identify recycled ICs.

1.3 Major Contributions

With the above motivation, this dissertation is devoted the development of on-chip structures and techniques for hardware Trojan detection and recycled ICs identification to improve the security, trustworthiness, and reliability of ICs. The major contributions of this thesis can be divided into the following three parts.

1.3.1 Hardware Trojan Detection

Hardware Trojan Detection in 3PIPs: as we mentioned, hardware Trojan detection in ICs is a difficult task. However, detection of Trojans in 3PIPs is even more difficult since IP vendors usually provide specifications and the source code, both of which may contain Trojans. Conventional side-channel techniques for IC trust are not applicable to IP trust. Identifying a few lines of RTL code in an IP core that contains a well-hidden Trojan is an extremely challenging task. Given its complexity, there is no silver bullet available.

One of the major contributions of this thesis is the development of a novel flow to detect hardware Trojans in 3PIPs, involving formal verification, coverage analysis, redundant circuit removal, sequential ATPG, and equivalence theorems. It is the first time that formal verification and code coverage are used to help identify potential Trojans. Redundant circuit removal and equivalence theorems are developed to reduce the number of potential Trojan gates.

Hardware Trojan Detection in ICs: the RON architecture is proposed

to generate a power supply fingerprint, used to identify malicious alterations in ICs. Each ring oscillator acts as a power monitor and captures the voltage drop caused by hardware Trojans close to it. In order to improve the sensitivity of the RON for hardware Trojan detection, we developed a revised RON structure. Each component of ROs in the revised RON is placed in each row of the design and each row contains at lease one component of one RO. Therefore, the voltage drop caused by hardware Trojans will be captured by at least one RO that shares power supply with them. A novel data analysis method is also proposed to separate the impact of hardware Trojans on ROs from that caused by process variations. The area overhead and power consumption of the RON is negligible compared to current designs with millions of gates.

The proposed RON structure was implemented in test chips fabricated using the IBM 90nm process by MOSIS. Hardware Trojans composed of different number of gates were also inserted into the design. With our proposed data analysis method and IC classification technique, the silicon results show that Trojan-inserted ICs could be identified even in the presence of obfuscating process variations, measurement noise, and environment variations.

1.3.2 Recycled ICs Detection

Another major contribution of this thesis is the development of techniques for recycled ICs detection. Since recycled ICs have been used before they are resold in

the market, the performance must have been degraded by aging effects. Therefore, performance degradation, such as RO's frequency degradation and path delay degradation, can be used to identify recycled ICs. Leakage current and transient current degradation could also be used for recycled ICs detection.

In this thesis, we define the recycled ICs problem and propose techniques using light-weight on-chip sensors and path delay degradation to detect recycled ICs. These light-weight on-chip sensors are the first attempts to solve recycled ICs problem. By placing the components in the sensors next to each other, the impact of aging effects on the sensors could be separated from that caused by process variations. With small area overhead, these sensors are demonstrated to be very effective. However, they only work when the designs already have those sensors in place but cannot detect ICs without such sensors. A path-delay fingerprinting flow is proposed to address this issue. With zero area overhead, the path-delay fingerprinting flow can identify recycled ICs for all digital ICs.

1.3.3 True Random Number Generator

A true random number generator (TRNG) is an important security module integrated in most ICs for secure data communication and storage. We propose two TRNGs that can generate sequences with high randomness. With small area overhead and limited power consumption, our proposed TRNGs increase the randomness of generated sequences by introducing more random noise to the circuit.

1.4 Organization

This thesis is divided into 9 chapters. The motivation, background, and contributions are provided in Chapter 1. Chapter 2 presents our case study of hardware Trojans inserted into 3PIPs and also our proposed flow to identify the potential hardware Trojan. Formal verification and coverage analysis are introduced in this chapter to help hardware Trojan detection in 3PIPs. Chapter 3 describes our on-chip structure RON to detect hardware Trojans in ICs. In order to improve the sensitivity of our hardware Trojan detection method, we propose a technique combining the revised RON and transient current to detect hardware Trojans in Chapter 4. The silicon evaluation results of our RON and IC classification algorithms are presented in Chapter 5. Chapter 6 describes our three light-weight on-chip sensors to identify recycled ICs based on the performance degradation and usage time of ICs. Chapter 7 presents the fingerprinting flow for recycled ICs identification based on path delay degradation caused by aging effects. Chapter 8 describes two novel TRNGs to generate random sequences with high randomness. We conclude this thesis in Chapter 9, with suggestions for future work.

Chapter 2

A Case Study for Hardware Trojans Detection in Third-Party Digital IP Cores

Hardware Trojans can be found in 3PIPs and ICs. We will focus on hardware Trojan detection in 3PIPs in this chapter. In general, 3PIP cores fall into one of the three categories: Soft, Firm, and Hard, depending on the format when they are supplied. Soft IP cores are described using VHDL or Verilog and are the most flexible and popular cores used in practice. Firm cores are described and synthesized for specific libraries while hard IP cores are described at the physical level and are supplied as GDSII file. Since soft IP cores are most widely used in practice, in this work, we will focus on targeting Trojans in such IPs.

Detection of such Trojans is extremely difficult since there is no known golden model for 3PIPs as IP vendors usually provide specification and source code, both of which may contain Trojans. The conventional side-channel techniques for IC trust are not applicable to IP trust. When a Trojan exists in an IP core, all the fabricated ICs will contain Trojans. The only trusted component

would be the specification from the SOC designer which defines the function, primary input and output, and other information of the 3PIP that they intend to use in their systems. For the 3PIP supplied as register transfer level (RTL) code, a Trojan can be very well hidden during the normal functional operation. A large industrial-strength IP core can include thousands of lines of code. Identifying a few lines of RTL code in a soft IP core that represent a Trojan is an extremely challenging task.

In this chapter, a case study is presented to detect Trojans in 3PIPs, based on identification of suspicious signals. Several concepts such as formal verification, code coverage analysis, and ATPG methods will be employed in our technique to achieve high confidence in whether the circuit is Trojan-free or Trojan-inserted. It is important to note that a 3PIP source code is largely Trojan free; only few parts may be suspicious. The challenge is to identify these suspicious parts that will most likely be part of a Trojan. Suspicious signals are identified first by coverage analysis with improved test bench. Removing redundant circuits and equivalence theorems will be applied to reduce the number of suspicious signals. Sequential ATPG is used to generate patterns to activate these suspicious signals. Our method considers both the characteristics of dormant Trojans and redundant circuits.

2.1 Formal Verification and Coverage Analysis

One of the important concepts used in our method is formal verification, which is an algorithmic-based approach to logic verification that exhaustively proves functional properties about a design [28]. It contains three types of verification methods that are not commonly used in the traditional verification namely model checking, equivalence checking, and property checking. All functions in the specification are defined as properties. The specific corner cases in the test suite as they monitor particular objects in a 3PIP could also be represented by properties, such as worry cases, inter-block interfaces, and complex RTL structures, wherever the protocols may be misused, assumptions violated, or design intent incorrectly implemented. Formal verification uses property checking to check whether the IP satisfies those properties. With property checking, we can explore every corner of the design. For example, in benchmark RS232, there are two main functionalities in the specification: (1) transmitter; (2) receiver. Figure 2.1 shows the waveform of the transmitter. Take the *start bit* as an example; with $Rst == 1'b1$, clk positive edge, and $xmitH == 1'b1$, the output signal $Uart\_xmit$ will start to transmit *start bit* "0". This functionality is described using SystemVerilog property shown in Figure 2.2 and the corresponding assertion is defined simultaneously. The remaining items in the specification are also translated to properties during formal verification. Once all the functionalities in the specification are translated to properties, coverage metrics can help identify suspicious parts in the 3PIP under

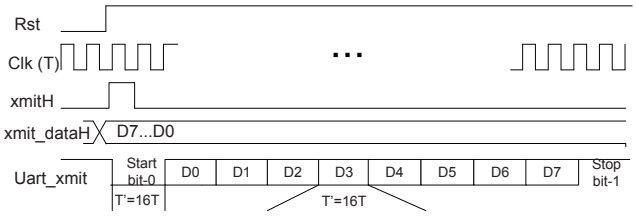


Fig. 2.1: Transmitter property in the specification.

```
01:   property e1;
02:     @(posedge uart_clk) disable iff (Rst)
03:     $rose(xmitH) |− > ##1 (uart_XMIT_dataH==0);
04:     endproperty
05:
06:     a1: assert property( e1 );
```

Fig. 2.2: One of the properties and assertions definition for RS232.

authentication. Those suspicious parts may be Trojans (or part of Trojans).

Coverage metrics include code coverage and functional coverage. Code coverage analysis is a metric that evaluates the effectiveness of a test bench in exercising the design [29] [30]. It has many different types but only a few of them are helpful for IP trust, namely line, statement, toggle, and finite state machine (FSM) coverage. Toggle coverage reports whether signals switch in gate-level netlist while the other three coverage metrics show which line(s) and statement(s) are executed, and whether states in FSM are reached in RTL code during verification. Figure 2.3 shows parts of line coverage report during our simulation with RS232. This report shows lines 72 and 74 are not executed, which help us improve the test bench by checking the source code. If the RTL code is easily readable, special patterns that can activate those lines will be added to test bench.

01: Line No	Coverage	Block Type
02: 69	1	ALWAYS
03: 70	1	CASEITEM
04: 71	1	CASEITEM
05: 72	0	CASEITEM
06: 73	1	CASEITEM
07: 74	0	CASEITEM
08: 82	1	ALWAYS
09: 82.1	1	IF
... ...	...	...

Fig. 2.3: Part of line coverage report.

Otherwise, random patterns will be added to verify the 3PIP.

Functional coverage is the determination of how much functionality of the design has been exercised by the verification environment. The functional requirements are imposed on both the inputs and outputs of the design and their interrelationships by the design specifications from SOC designer (i.e. IP buyers). All the functional requirements can be translated as different types of assertions like Figure 2.2. Functional coverage checks those assertions to see whether they are successful or not. Table 2.1 shows part of assertions coverage report (Assertion a1 is defined in Figure 2.2). The number of Attempts in the table means there are 500,003 positive edge clocks during the simulation time when tool tries to check the assertion. The Real Success represents assertion success while Failure/Incomplete denote assertion failure/incomplete. With "0" failure, this property is always

Table 2.1: Part of assertion report with RS232.

Assertion	Attempts	Real Success	Failure	Incomplete
test.uart1.uart_checker.a1	500,003	1,953	0	0
test.uart1.uart_checker.a2	1,953	1,952	0	1
...	...	...	...	...

satisfied.

If all the assertions generated from the specification of the 3PIP are successful and all the coverage metrics such as line, statement, and FSM are 100%, then with a high confidence we can assume that the 3PIP is Trojan-free. Otherwise, the uncovered lines, statements, states in FSM, and signals are considered as suspicious. All the suspicious parts constitute our suspicious list.

2.2 Techniques for Suspicious Signals Reduction

Based on formal verification and coverage metric, we proposed our flow to verify the trustworthiness of 3PIP. The basic idea of our proposed solution is that without redundant circuit and Trojans in a 3PIP, all the signals/ components are expected to change their states during verification and 3PIP should function perfectly. Thus, the signals/components that stay stable during toggle coverage analysis are considered suspicious as Trojan circuits do not change their states frequently. Each suspicious signal is then considered as the $TriggerEnable$. Figure

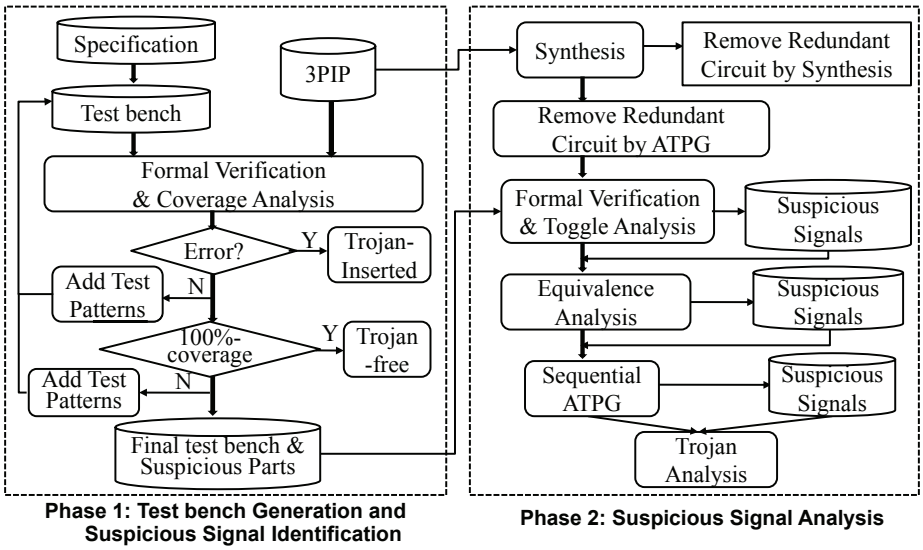


Fig. 2.4: The proposed flow for identifying and minimizing suspicious signals.

2.4 shows our flow to identify and minimize the suspicious parts, including test pattern generation, suspicious signal identification, and suspicious signal analysis. Each step in the figure will be discussed in detail in the following subsections.

2.2.1 Phase 1: Test Bench Generation and Suspicious Signal Identification

In order to verify the trustworthiness of 3PIPs, we hope the coverage of test bench could be 100%. However, it is very difficult to achieve 100% coverage for every 3PIP, especially those with tens of thousand lines of code. In our flow, the first step is to improve the test bench to obtain a higher code coverage with acceptable simulation run time. With each property in the specification and basic functional test vectors, formal verification reports line, statement, and FSM coverage for the

RTL code. If one of the assertions is failed even just once during verification, the 3PIP is considered untrusted, containing Trojans or bugs. If all the assertions are successful and the code coverage is 100%, the 3PIP is considered trusted. Otherwise, more test vectors need to be added in the test bench. The basic rule of adding new vectors is to activate the uncovered parts as much as possible. But the verification time will increase as the number of test vectors increases. With the acceptable verification time and certain coverage percentage, both defined by IP buyer, the final test bench will be generated and the RTL source code will be synthesized for further analysis.

2.2.2 Phase 2: Suspicious Signals Analysis

Redundant Circuit Removal (RCR): The redundant circuit must be removed from our suspicious list since they also tend to stay at the same logic value during the verification and input patterns cannot activate them. Removing the redundant circuit involves sequential reasoning, SAT-sweeping, conflict analysis, and data mining. The SAT method integrated in Synopsys Design Compiler (DC) is used in our flow.

We also develop another method to remove redundant circuit. Scan chain will be inserted into the gate-level netlist after synthesis for design for testability and ATPG generates patterns for all the stuck-at faults. The untestable stuck-at faults during ATPG is likely to be redundant logic. The reason is that if

the stuck-at fault is untestable, the output responses of the faulty circuit will be identical to the output of the fault-free circuit for all possible input patterns. Thus, when ATPG identifies a stuck-at- 1/0 fault as untestable, the faulty net can be replaced by logic 1/0 in the gate-level netlist without scan-chain. All the circuits driving the faulty net will be removed, as well. Figure 2.5(a) shows the circuit before redundant circuit removal. Stuck-at-0 fault of net F is untestable when generating patterns. Net F will be replaced by 0 and the gate G driving it will be removed from the original circuit as shown in Figure 2.5(b).

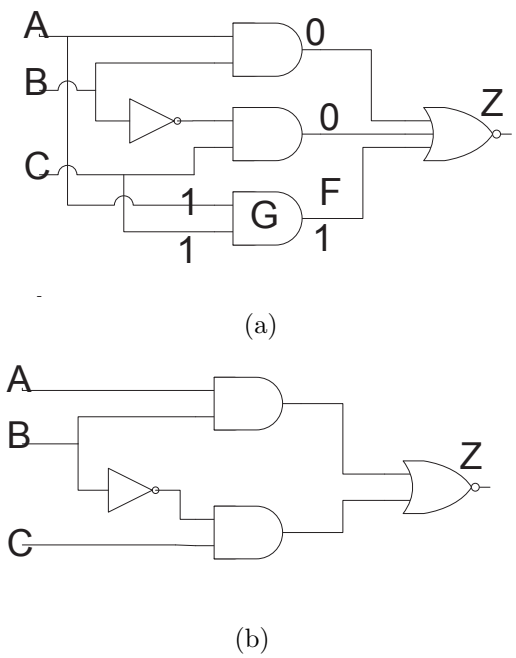


Fig. 2.5: (a) Before removing redundant circuit with untestable F stuck-at-0 fault and (b) After removing redundant circuit.

After redundant circuit removal, toggle coverage analysis for gate-level netlist

without scan chain will identify which signals do not toggle (also called quiet signal) during verification with the test bench generated in Phase 1. These signals will be suspicious and added to our suspicious list. By monitoring these suspicious signals during verification, we can obtain the logic value those signal are stuck at. We try to further reduce the number of suspicious signals in the following.

Equivalence Analysis: Fault equivalence theorems are known to reduce the number of faults during ATPG [41]. Similarly, we develop suspicious signal equivalence theorems to reduce the number of suspicious signals.

$Theorem1$: If signal A is the D pin of a flip-flop (FF) while signal B is the Q pin of the same FF, the quiet signal A makes signal B quiet. Thus signal A is considered equal to B, which means if we can find the pattern that can activate A, it will activate B as well. Then signal B will be removed from the suspicious signal list.

As the QN port of a FF is the inversion of the Q port, they will stay quiet or switch at the same time. Thus the suspicious signal B would be considered equal to A and should be removed from the suspicious list.

$Theorem2$: If signal A is the output pin of an inverter while signal B is its input, they will stay quiet or switch at the same time. Thus the suspicious signal B would be considered equal to A and should be removed from the suspicious list.

$Theorem3$: One of the input of AND gate A stuck-at-0 will cause the output B stay quiet and one of the input of OR gate C stuck-at-1 will make the output

D high all along. Thus, for AND gate, B stuck-at-0 is identical to A stuck-at-0 while for OR gate, D is identical to C stuck-at-1.

Sequential ATPG: After reducing the number of suspicious signals by applying the above equivalence theorems, we use sequential ATPG to generate special patterns to change the value of certain signals during simulation. Stuck-at faults are targeted by the sequential ATPG to generate a sequential pattern to activate the suspicious signals when applied to the 3PIP. If the 3PIP functions perfectly with this pattern, the activated suspicious signals are considered part of the original circuit. Otherwise, there must be malicious inclusion in the 3PIP.

2.3 Simulation Results

We applied the flow to RS232 circuit. 9 Trojans from our original design and 10 Trojans from [31] were inserted into the 3PIP. Totally, there are 19 RS232 benchmarks with one Trojan in each IP. In the following, we first present simulation setup and test bench analysis for the 19 Trojan-inserted benchmarks. Next, the results of redundant circuit removal and reducing the suspicious signals will be presented. Finally, Trojan coverage analysis will be discussed.

2.3.1 Benchmark Setup

Currently, there are a total of 22 benchmarks with different Trojans in Trust-Hub [31], from which 10 of them are at RT Level. Readers can visit [31] for more

Table 2.2: Analyzing impact of test bench on coverage metrics (benchmark with Trojan 1 is used).

Test Bench #	Test Bench 1	Test Bench 2	Test Bench 3	Test Bench 4	Test Bench 5
Test Patterns #	2,000	10,000	20,000	100,000	1,000,000
Verification Time	1 minutes	6 minutes	11 minutes	56 minutes	10 hours
Line Coverage	89.5%	95.2%	98.0%	98.7%	100%
FSM State Coverage	87.5%	87.5%	93.75%	93.75 %	100%
FSM Transition Coverage	86.2%	89.65%	93.1%	96.5%	100%
Path Coverage	77.94 %	80.8%	87.93%	97.34%	100%
Assertion	Successful	Successful	Successful	Successful	Failure

details about the specification, structure, and functionality of these Trojans in the 10 RTL benchmarks. However, the other 9 Trojans are briefly described in the following:

$Trojan1$: The trigger of Trojan 1 is a special input sequence $8'ha6 - 8'h75 - 8'hc0 - 8'hff$. The payload changes the FSM in the transmitter of RS232 from state *Start* to *Stop*, which means that once the Trojan is triggered, RS232 will stop transmitting data ($outputdata = 8'h0$). Since the trigger of the Trojan is a four special input sequence, the probability to detect the Trojan during verification is $1/2^{32}$. If the baud rate is 2400 and RS232 transmits 240 words in one second, it will take 207.2 days to activate the Trojan and detect the error. In other words, it would be practically impossible to detect it by conventional verification. When we insert this Trojan into RS232, an FSM is used to describe the Trojan input

sequence. A three-bit variable state represents the FSM.

$Trojan2$: This Trojan only adds four lines to the original RTL code. If the transmitting word is odd and the receiving word is $8'haa$, RS232 will stop receiving words. This Trojan is less complex compared to Trojans 1 however it provides opportunities to demonstrate the effectiveness of each step of the proposed flow.

$Trojan3$: The trigger of Trojan 3 is the same as that of Trojan 1, but the payload is different. Trojan 1 changes the state machine while Trojan 3 changes the shift process. The eighth bit of the transmitting word will be replaced by a Trojan bit during transmission. The Trojan bit could be authentication information, the special key to enable the system, or other important information.

$Trojan4$: Trojan 4 is designed to act like a time bomb. A counter is inserted into RS232 to count the number of words that have been sent out. After sending $10'h3ff$ words, the Trojan will be activated. The sixth bit of the transmitting word will be replaced by a Trojan bit.

$Trojan5$: After $24'hffffff$ positive edge clock, this Trojans enable signal will become high. The sixth bit of the transmitting word will be replaced by a Trojan bit.

$Trojan6$: If RS232 receives 0 when the system is reset, the Trojan will be activated. The eighth bit of the transmitting word will be replaced by a Trojan bit.

$Trojan7$: When the transmitter sends a word $8'h01$ and the receiver receives

a word $8'hef$ at the same time, the Trojan will be activated. A Trojan bit will replace the first bit of the transmitting word.

$Trojans8$ & $9:$ They do not tamper the original function of RS232 but add extra one stage (Trojan 8) and three stage (Trojan 9) ring oscillator to the RTL, which will increase the temperature of the chip quickly in the field if they get activated.

2.3.2 Impact of Test Bench on Coverage Analysis

All the items in the specification are translated into properties and defined as assertions in test bench. Assertion checkers is used to verify the correctness of assertions by SystemVerilog. Another most important feature of a test bench is the input patterns. Some test corners need special input patterns. The more input patterns in test bench, the more, for example, lines will be covered during verification. Table 2.2 shows five test benches with different test patterns and verification time for various coverage metric reports for the RS232 benchmark with Trojan 1. Generally, the verification time increases with more test patterns and the code coverage is higher, as well. For Test Bench 1 to Test Bench 4, all the coverage reports are less than 100% and all the assertions are successful, which states that the Trojan is dormant during the entire verification. But if we add special test patterns in Test Bench 5 and increase the pattern count significantly, which can activate the Trojans inserted in the benchmark, the code coverage

could achieve 100% and one of the assertion experiences a failure, which means the Trojan is triggered and RS232 gives an erroneous output. A conclusion that the IP is Trojan-inserted would be made. However, it is not easy to generate a test bench with 100% code coverage for large IPs and the verification time will be extremely long. This phase of the flow can help improve quality of the test bench. Given the time-coverage trade off, a test bench is selected for further analysis. Thus, here, we selected Test Bench 4 to verify this and the remaining benchmarks.

2.3.3 Reducing the Suspicious Signals

All the 19 benchmarks with different Trojans were synthesized to generate the gate-level netlist. Removing redundant circuit was done during the synthesis process with special constrains using Design Compiler. The simulation results are shown in Table 2.3. The second column in the table shows the area overhead of each Trojan after generating the final layout. From this table, we can see that Trojans are composed of different sizes, gates, and structures as well as different triggers and payloads as mentioned earlier. The smallest Trojan has only 1.15% area overhead. The percentage of Trojan area covered by suspicious signals *SS-Overlap-Trojan* is obtained by *SS-Overlap-Trojan*$=\frac{N_{SS}}{N_{TS}}$ where N_{SS} is the number of suspicious signals and N_{TS} is the number of Trojan signals. The results in Table 2.3 show that *SS-Overlap-Trojan* is between 67.7% and 100% as shown in seventh

Table 2.3: Suspicious signal analysis.

Benchmark (RS232)	Trojan Area Overhead	Step 1: Number of SS after RCR with Synthesis	Step 2:Number of SS after RCR with with ATPG	Step 3: Number of SS after Equivalence Analysis	Step 4: Number of SS after Sequential ATPG	SS-Overlap - Trojan
With Trojan 1	11.18%	22	20	17	12	100%
With Trojan 2	20.35%	17	16	3	Trojan is identified	100%
With Trojan 3	10.48%	20	15	15	10	97.3%
With Trojan 4	20.35%	3	3	3	2	87.6%
With Trojan 5	4.59%	9	8	8	7	100%
With Trojan 6	1.15 %	1	1	1	Trojan is identified	100 %
With Trojan 7	3.79 %	3	3	3	2	100%
With Trojan 8	1.15 %	1	Trojan is removed	-	-	100%
With Trojan 9	3.79 %	3	Trojan is removed	-	-	100%
TR04C13PI0	1.6%	8	3	3	3	100%
TR06C13PI0	1.8%	9	3	3	3	100%
TR0AS10PI0	2.09%	8	1	1	1	100%
TR0CS02PI0	25.3%	59	55	39	39	67.7%
TR0ES12PI0	2.09%	8	1	1	1	100%
TR0FS02PI0	25.0%	30	28	20	20	73.3%
TR2AS0API0	11.9%	19	18	11	11	100%
TR2ES0API0	12.0%	20	18	11	11	100%
TR30S0API0	12.4%	22	20	13	13	93.6%
TR30S0API1	12.3%	25	22	14	14	87.3%

column. If all the suspicious signals are part of Trojan, the *SS-Overlap-Trojan* would be 100%. This indicates that the number of signals in the final suspicious list fully overlapped with those from Trojan. This is an indicator of how successful the flow is in identifying Trojan signals. In addition, if the Trojan is removed or detected by sequential ATPG, the *SS-Overlap-Trojan* would be 100%, as well.

Test Bench 4 is used to verify the gate-level netlist and toggle coverage analysis reports which signals in each Trojan-inserted circuit are not covered by the simulation with all the successful assertions. Those quiet signals are identified as suspicious. The number of suspicious signals of each benchmark is shown in the third column in Table 2.3. Different benchmarks have different number of suspicious signals based on size of Trojans. The larger the Trojan is, the more suspicious signals it has. On the other hand, the suspicious signals stuck-at values are monitored by verification. All stuck-at-faults are simulated by ATPG tool with scan chain in the netlist. If the fault is untestable, the suspicious circuit is a redundant circuit and will be removed from the original gate level netlist, in addition to the gates that drive the net. The number of suspicious nets after redundant circuit removal is shown in the fourth column in Table 2.3. From the table, we can see that the suspicious nets of benchmarks with Trojan 8 and Trojan 9 are zero, which means if the redundant circuit are removed in the two benchmarks, the benchmarks will be Trojan-free. The reason that redundant circuit removal can distinguish Trojans is that some Trojans are designed without

payload and have no impact on circuit functionality. Thus, we can conclude that such Trojans can be removed by redundant circuit removal.

The remaining suspicious nets of each benchmark were processed by equivalence analysis and sequential ATPG. The fifth and sixth columns in Table 2.3 show the number of suspicious signals after the two steps. We can see that equivalence analysis can reduce a large number of suspicious signals and sequential ATPG can be effective as well. For benchmarks with Trojan 2 and Trojan 6, the sequential ATPG can generate sequential pattens for the stuck-at faults in the suspicious signal. The sequential test patterns improve the test bench and increase its coverage percentage. Even though the coverage percentage is not 100%, some assertions experience failure during simulation. Thus, we conclude that the benchmark with Trojan 2 and Trojan 6 is Trojan-inserted.

We have implemented our flow on 10 of trust benchmarks from Trust-Hub [31] and the results are reported in rows 11 to 20 in Table 2.3 show that the presented flow can effectively reduce the total number of suspicious signals. In addition, as shown in seventh column, there is a good overlap between the number of suspicious signals and actual Trojan signals inserted into each benchmark. However, we experience low *SS-Overlap-Trojan* with a couple of benchmarks, such as RS232-TR0CS02PI0, since part of this Trojan was activated during simulation.

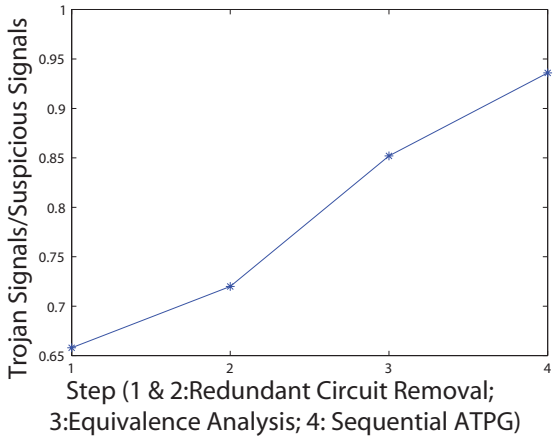


Fig. 2.6: Average Trojan signals/Suspicious signals in 19 benchmarks.

2.3.4 Trojan Coverage Analysis

In the suspicious list, not all of signals are part of Trojans. However, $TriggerEnable$ must be in the suspicious list if the IP contains Trojan. Once one net is identified as part of Trojans, we conclude that the 3PIP is Trojan-inserted. All the gates driving this net are considered to be Trojan gates. Figure 2.6 shows that the percentage of Trojan signals in the suspicious list increases significantly with our flow. As we apply different steps (step 1 through 4) to the benchmarks, on average 72% of the suspicious signals are part of Trojan after redundant circuit removal with synthesis and ATPG in the 19 benchmarks. However, the percentage increases to 85.2% when equivalence analysis is done and 93.6% of signals in the suspicious signal list are part of Trojans after sequential ATPG is applied to these benchmarks.

2.4 Conclusions

In this chapter, we have presented a study to verify trustworthiness of 3PIPs, involving formal verification, coverage analysis, redundant circuit removal, sequential ATPG, and equivalence theorems. The code coverage generates the suspicious signals list. Redundant circuits are removed to reduce the number of suspicious signals. Equivalence theorems are developed for the same purpose. Sequential ATPG is used to activate these suspicious signals and some Trojans will be detected. However, more work are needed if we want to get 100% hardware Trojan detection rates in 3PIPs.

Chapter 3

Hardware Trojans Detection in ICs Using Basic Ring Oscillator Network

In this chapter, we will introduce our proposed techniques for hardware Trojan detection in ICs. A new structure, namely RON, is developed with the ability to detect Trojans that cause power fluctuations, thereby uncovering the malicious inclusion. A number of ring oscillators (ROs) acting as *power monitors*, distributed across the entire IC, constitute the RON, which takes into account the noise caused by the Trojan gates and those caused by both inter-die and intra-die process variations. The output of each ring oscillator represents one part of the power signature of the entire IC. With N_{RO} ring oscillators in the IC, a series of power signatures can be generated by the RON. An off-chip test equipment would be able to select which ring oscillator should be used to generate the signature and could disable the RON when IC operates in functional mode. The number of ring oscillators, N_{RO}, could be adjusted according to the size of the IC and sensitivity to Trojans, thereby scaling the network and optimizing Trojan detection. Statisti-

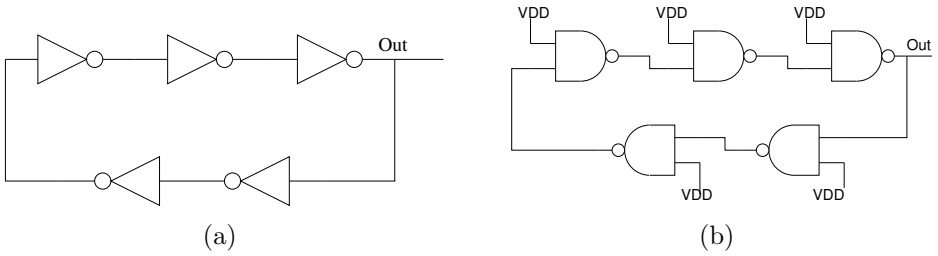


Fig. 3.1: Five-stage ring oscillators.

cal data analysis is used to effectively distinguish the power differences caused by Trojans from those of process variations, and identifies hardware Trojans inserted into the IC.

3.1 Analyzing Impact of Power Supply Noise on Ring Oscillators

Two simple five-stage ring oscillators are shown in Figure 3.1: the ring oscillator in Figure 3.1(a) consists of inverters and the ring oscillator in Figure 3.1(b) is composed of NAND gate. The second ring oscillator has a higher sensitivity to supply noise since one of its inputs is connected to power supply but offers larger area overhead. In this thesis, we only use the first ring oscillator as power monitor due to its easier analytical analysis. The frequency of this ring oscillator is determined by the total delay of all the inverters, in the presence of supply voltage and process variations. Assume that each stage in the ring oscillator provides a delay of t_d. The delay of the n-stage ring oscillator is approximately $2 * n * t_d$ and the oscillation frequency will be:

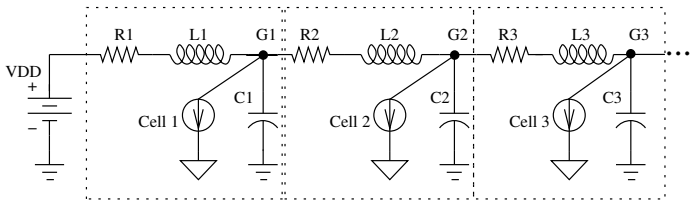


Fig. 3.2: The RLC model of a simple power line in a power distribution network.

$$f = \frac{1}{2 * n * t_d} \tag{3.1}$$

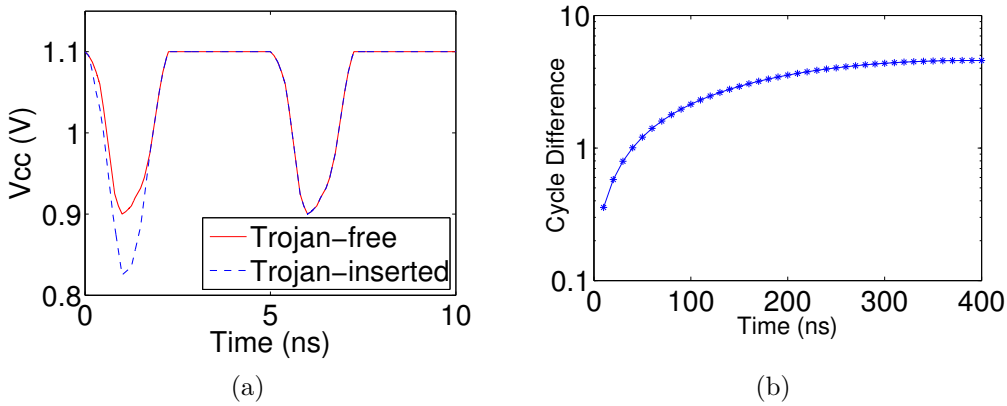
The delay of each inverter varies according to parameters such as temperature, supply voltage (VDD), load capacitance (CL), threshold voltage (Vth), channel length (L), oxide thickness (Tox), and transistor channel width (W). Since all ICs can be tested under the same temperature, the environmental variation will not be considered in this work. All the remaining parameters are susceptible to process variations and power supply noise.

Power supply noise (also known as voltage drop) impacts the delay of the logic gates. When the voltage drops, the delay of the gates increases. Thus, a change in the supply voltage of any inverter in a ring oscillator impacts the delay of all associated gates, and therefore impacts the oscillation frequency. Concerning today's tightly designed power supply distribution networks, transitions in some gates can impact the power supply of other gates within close proximity [42]. Figure 3.2 shows a simple power line model in which VDD supplies one row in standard cell design. The indicated VDD represents the point where a via connects

the power rail to the upper metal layer in a power distribution network. Nodes G1, G2, and G3 connect to adjacent cells represented as current source for Cell 1, Cell 2, and Cell 3. Here, for sake of simplicity, the power via is assumed to have zero impedance and each interconnect is modeled by a resistance, inductance, and capacitance (RLC) network. The contribution of each current source to the overall noise is described in Equation 3.2 where $V1$, $V2$, and $V3$ (voltage at nodes G1, G2, and G3) are the power supply noise spectrum, $Vii = Z_{ii} * I_{ii} (i = 1, 2, 3)$ (Z_{ii} is the impedance of node i and I_{ii} is the current) is the power noise, $\rho_{ij}(i, j = 1, 2, 3)$ is voltage division coefficient, and ω is the frequency of the circuit. From the equation, we can see that $V1$, $V2$, and $V3$ are related to the neighboring gates, demonstrating that a gate's transition has effect on neighboring gates connected to the same VDD line.

$$\begin{aligned} V1 &= V11 + \rho_{21}(\omega) * V22 + \rho_{31}(\omega) * V33 \\ V2 &= \rho_{12}(\omega) * V11 + V22 + \rho_{32}(\omega) * V33 \\ V3 &= \rho_{13}(\omega) * V11 + \rho_{23}(\omega) * V22 + V33 \end{aligned} \tag{3.2}$$

For Trojan-inserted ICs, the switching gates in the Trojan would cause small voltage drop on the VDD line and ground bounce on the VSS line. Thus, with the same input patterns, the power supply noise affecting the Trojan-free IC and Trojan-inserted IC will differ. In order to verify the impact of the Trojan on the frequency of the ring oscillator, we implemented a 5-stage ring oscillator (shown in Figure 3.1(a)) in 90nm technology for simulation.



(a) (b)

Fig. 3.3: (a) Power supply variations for Trojan-free and Trojan-inserted circuits; (b) Cycle difference caused by Trojan gates' switching.

In Figure 3.3(a), assume that the dashed line denotes the dynamic power in the presence of a Trojan and the solid line denotes the Trojan-free power (assuming $VDD = 1.1V$). As can be observed, the two supply voltages only differ during the first 2ns. These two power waveforms are applied to the ring oscillator for 400ns. Figure 3.3(b) shows the cycle count difference due to the extra noise caused by the Trojan. At time 0, the two ring oscillators denoting *with* and *without* an inserted Trojan have the same period. However, with the presence of power supply noise, the difference will grow steadily as the measurement duration increases.

3.2 Ring Oscillator Network

As mentioned earlier, Trojan gates switching impacts the frequency of a ring oscillator due to injected power supply noise. Process variations can impact the

threshold voltage, channel length, and oxide thickness in circuit gates which, in turn, impacts power supply noise distribution in an IC. Since these effects may be localized, one ring oscillator may not have enough sensitivity to distinguish the effect of Trojans and process variations. A ring oscillator placed in one corner of an IC may not be able to capture noise effects which occur due to a Trojan placed in another corner of the IC. A ring oscillator network however can improve the sensitivity to Trojan noise, and increase the accuracy in determining Trojan's contributions using relative values.

Our RON is composed of N_{RO} ring oscillators distributed across the entire IC. For different ICs, the number of ring oscillators can be adjusted accordingly depending on the sensitivity of the ring oscillators to the gate switching in a predetermined proximity. The output of RON in Trojan-free ICs generates a power signature. In this thesis, we assume that a number of golden ICs can be identified via a thorough test process. If the output of an IC under authentication is not compatible with the expected signature, the IC may contain a Trojan.

The oscillation cycle count generated from the ring oscillators in the RON is used to generate the IC's signature. For i_{th} ring oscillator, the total accumulated cycles, C_i, in the measurement time T is:

$$C_i = \int_0^T \frac{1}{2 * n * t_{di}(t)} \, dt \tag{3.3}$$

where $t_{di}(t)$ is the inverter delay which will vary with time as the input patterns change. Let $\Delta t_{dti}(t)$ represent the change in inverter delay of i_{th} ring oscillator

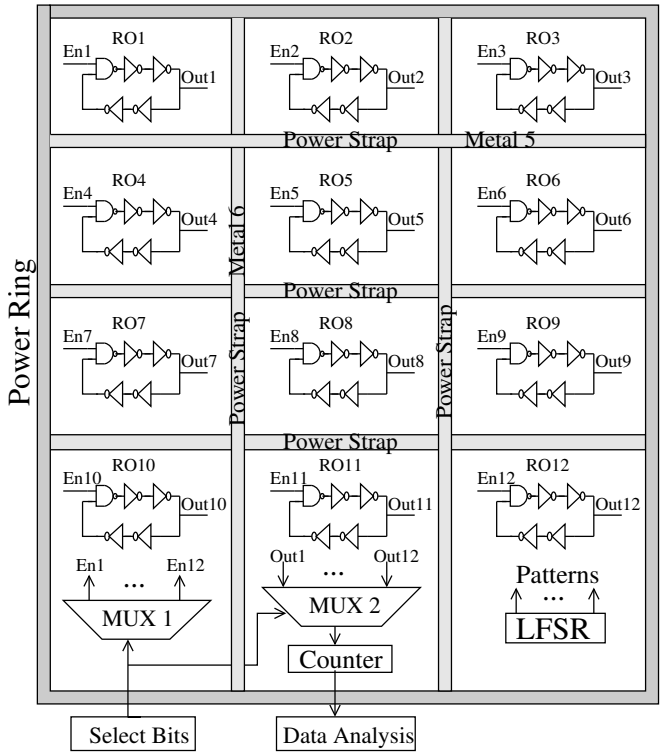


Fig. 3.4: A RON with N_{RO} ring oscillators distributed in the circuit layout.

caused by Trojan effects and C_{TFi} and C_{TIi} denote the total cycle count for Trojan-free and Trojan-inserted ICs, respectively. The effect a Trojan has on i_{th} ring oscillator (ΔC_i) is presented by Equation 3.4. The value of ΔC_i is related to the number of stages in a ring oscillator (n), the measurement time (T), and the Trojan's impact on inverter delay ($\Delta t_{dti}(t)$). The Trojan's impact on a ring oscillator is determined by the size of the Trojan, switching activity of the Trojan, and the distance between the Trojan and the ring oscillator.

$$\Delta C_i = C_{TIi} - C_{TFi} = -\int_0^T \frac{\Delta t_{dti}(t)}{2 * n * t_{di}(t) * (t_{di}(t) + \Delta t_{dti}(t))}\, dt \tag{3.4}$$

Figure 3.4 shows the proposed ring oscillator network with N_{RO}=12 oscillators inserted into the ISCAS'89 s9234 benchmark circuit according to power straps in the layout. One RO is inserted into each grid surrounded by power straps. The on-chip structure also includes a linear feedback shift register (LFSR), one decoder, one multiplexer, and one counter. The LFSR will supply random functional patterns for the entire IC during the signature generation and authentication processes; the same seed must be used for each golden IC and each IC under authentication. A decoder and multiplexer are used to select which ring oscillator is measured. When a ring oscillator is selected, the decoder enables that particular RO and the multiplexer transmits the output of that RO to the counter. The counter measures the cycle count of the selected ring oscillator over a specified duration. The number of stages in a ring oscillator is limited by the operating speed of the counter, which is determined by the technology node. For

example, using our 90nm technology, a 16-bit counter can operate at a maximum frequency of $1GHz$ according to HSPICE simulation.

The RON architecture has a small area overhead, mainly caused by the counter and LFSR. For instance, the overhead is 10.8% for the smaller benchmark circuit, s9234 (two vertical power straps and three horizontal power straps, $N_{RO} = 12$), 3.6% for s35932 benchmark circuit (three vertical power straps and three horizontal power straps, $N_{RO} = 16$), and 0.9% for DES circuit (five vertical power straps and five horizontal power straps, $N_{RO} = 30$). We believe that the area overhead will be negligible for larger circuits even if N_{RO} increases considerably based on power planning, since the counter size does not increase linearly with N_{RO}. Also, LFSR is commonly used for built-in self-test (BIST) in modern designs.

The RON is resilient to removal and tampering attacks. It is inherently difficult for an attacker to remove the ring oscillator network, due to (i) its distributed placement throughout the entire IC and (ii) the expected measurement results from each ring oscillator, i.e., the designer relies on the ability to capture RON data from each embedded ring oscillator. If a specific ring oscillator is not reporting data, the designer should assume the design has been attacked. On the other hand, ring oscillator is sensitive to its stage count and inverter type. For the RON inserted by the designer, the frequency falls in a certain range considering variations. If one of them is not within the range, it must be tampered with.

In addition, similar to ring oscillator based physical unclonable functions (PUFs), the RON architecture is also resilient to modeling and reverse engineering attacks.

3.3 Statistical Analysis

When the Trojan is small or widely distributed, distinguishing between noise generated by Trojan gates and process variations may be exceedingly difficult. Therefore, as an extension, a signature must be generated by recording all ring oscillators cycle count from a large number of ICs of the same design. Since the ICs will all be subject to different process variations, this signature can be statistically more tolerant to errors. In order to separate the effect of process variations and Trojans, a data analysis flow is suggested in this work, which includes three methods namely: (i) Simple Outlier Analysis, (ii) Principal Component Analysis (PCA), and (iii) Advanced Outlier Analysis. Simple outlier analysis offers least complexity compared with the other two data analysis methods.

Simple outlier analysis is based on the oscillation cycle distribution of each ring oscillator in the RON. For each ring oscillator, the oscillation cycle is within a certain range for Trojan-free ICs. If the oscillation cycle of one ring oscillator in the IC under authentication is outside of the range, this IC is considered suspicious and might contain a Trojan. This method uses the information from individual ring oscillators but not the relationship between them in the RON. Usually, this method can identify a small number of Trojan-inserted ICs but not most based

on our results. If oscillation cycle count of all ring oscillators in an IC under authentication is within each Trojan-free IC's signature, the data collected from this IC will be processed by PCA and advanced outlier analysis.

The concept of principal component analysis [43] is used to account for the N_{RO} variables (one variable represents one ring oscillator). The relationship between the data from the N_{RO} ring oscillators is considered by PCA when it transforms the N_{RO} variables into uncorrelated variables. For example, noting similarities in oscillation readings between two adjacent ring oscillators, would imply a correlation in the data. The oscillation cycle count of N_{RO} ring oscillators in the Trojan-free ICs will be analyzed by PCA and convex hull [44] is constructed with the first three components. If the output of RON is beyond the convex, a Trojan must exist in the IC under authentication. However, if the output is inside the convex, advanced outlier is used for futher analysis and validation.

Advanced outlier analysis is developed to identify the ICs with Trojan that cannot be detected by simple outlier analysis and PCA. It considers the relationship between ring oscillators in the RON. The pseudo-code is shown in Figure 3.5, which consists of two steps. The first step, Measurement, generates N_{RO}*(N_{RO}-1) power signatures from N_{TF} Trojan-free ICs. For each Trojan-free IC, the total oscillation cycle count from the RON is $C_{RON} = \sum_{m=1}^{N_{RO}} C_m$. Then, the data from the RO_i (C_i) and RO_j ($j \neq i$) (C_j) are selected to calculate $x_i = (C_{RON} - C_i)/C_i$ and $y_j = (C_{RON} - C_i)/C_j$. Finally, ($x_i$, y_j) from all the Trojan-free ICs would be

Step 1: Measurement

01: **Collect** data from N_{TF} Trojan-free ICs with N_{RO} ring oscillators

02: **for** $(i = 1, i <= N_{RO}, i++)$ { //select $i^{th} RO$

03: **for** $(j = 1, j <= N_{RO}, j++)$ $(j \neq i)$ { //select $j^{th} RO$

04: **for** $(k = 1, k <= N_{TF}, k++)$ { //select k^{th} Trojan-free IC

05: x_{ki}=$(\sum_{m=1}^{N_{RO}} C_{km} - C_{ki})/C_{ki}$;

06: y_{kj}=$(\sum_{m=1}^{N_{RO}} C_{km} - C_{ki})/C_{kj}$;

07: **plot**(x_{ki}, y_{kj});

08: }

09: The power signature, PS_{ij}, is created from all N_{TF} ICs.

10: } //x_{ki} is named as the first vector

11: } //y_{kj} is named as the second vector

Note: C_{km}, C_{ki}, C_{kj}: Oscillation cycle count of RO_m, RO_i, RO_j in k^{th} IC

Step 2: Authentication

For each IC under authentication:

01: **Collect** data from N_{RO} ring oscillators ($C_{RON} = \sum_{i=1}^{N_{RO}} C_i$).

02: **for** $(i = 1, i <= N_{RO}, i++)$

03: { $x = (C_{RON} - C_i)/C_i$;

04: **for** $(j = 1, j <= N_{RO}, j++)$ $(j \neq i)$;

05: { $y = (C_{RON} - C_i)/C_j$;

06: **plot**(x,y);

07: **if** ((x,y) is outside of the power signature PS_{ij}

08: {The IC is Trojan-inserted; Break; }

09: **else** go on;

10: }

11: }

Note: C_i, C_j: Oscillation cycle count of RO_i, RO_j

Fig. 3.5: Advanced outlier analysis procedure.

plotted to generate PS_{ij} power signature. Thus, N_{RO}*(N_{RO}-1) power signatures can be generated. The second step, Authentication, deals with the IC under authentication using the same process. If one of the IC's signatures is beyond the N_{RO}*(N_{RO}-1) power signatures, then it is assumed to contain a Trojan.

3.4 Results and Analysis

In order to verify the effectiveness of the RON architecture, we implemented $N_{RO} = 12$ ring oscillators with 5-stage inverters in (i) s9234 benchmark using 90nm technology, including 2 vertical and 3 horizontal power straps, for IC simulation and (ii) AES circuit on Xilinx Spartan-3E FPGA for hardware validation. For IC simulation, six Trojans (T_1 through T_6) with different sizes, distributions, and switching activities are inserted into s9234 benchmark. s9234, which is a small benchmark with 145 flip-flops and 420 gates, is selected for simulation rather than AES (6,089 flip-flops and 18,103 gates) to be able to run the very slow process of Monte Carlo simulations. Few of the Trojans can change the output of the original circuits when they are enabled. The location of the ring oscillators and Trojans are shown in Figure 3.6. The dark-colored circles in the figure represent the corresponding regions used in the actual layout by the Trojans. Four of the Trojans (T_1, T_2, T_4, and T_5) are placed around the ring oscillator RO8. Gates in Trojans (T_3 and T_6) are distributed at different regions within close proximity to RO5, RO7, RO8, and RO9. All Trojans have passed our validation test suite including

Table 3.1: Oscillation cycle count of ring oscillators in presence of Trojan gates switching without process variations

RO	T1			T2			T3			T4		
	C_{TI}	C_{TF}	ΔC	C_{TI}	C_{TF}	ΔC	C_{TI}	C_{TF}	ΔC	C_{TI}	C_{TF}	ΔC
RO1	4933	4939	-6	4985	4989	-4	4944	4965	-21	4999	4999	0
RO5	4735	4744	-9	4740	4749	-9	4908	4948	-40	4906	4925	-19
RO8	4714	4545	-31	4932	4974	-42	4796	4855	-59	4604	4635	-31
RO12	5279	5282	-3	4999	4999	0	4943	4966	-23	5027	5031	-4

RO	T5			T6		
	C_{TI}	C_{TF}	ΔC	C_{TI}	C_{TF}	ΔC
RO1	4976	4985	-9	5000	5000	0
RO5	4989	4994	-5	4792	4819	-27
RO8	4936	4981	-45	4925	4974	-49
RO12	5054	5062	-8	5242	5250	-8

100,000 random functional patterns as well as structural patterns generated using automatic test pattern generation (ATPG) tool. During simulation, the same input patterns generated by LFSR are applied to all ICs, including those which are Trojan-free. The fast Spice simulation tool Nanosim from Synopsys is used to conduct the power analysis and collect the oscillation cycle count in presence of process variations.

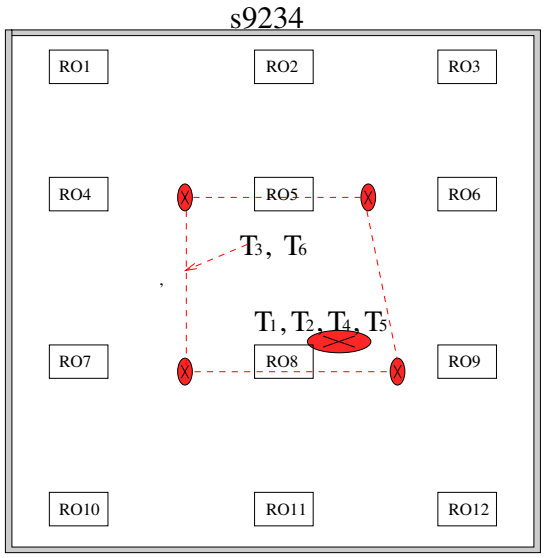


Fig. 3.6: s9234 with 12 ROs and 6 Trojans. One Trojan at a time is inserted into the circuit.

3.4.1 Trojan Size Analysis

The six inserted Trojans are designed with varying sizes to analyze the impact they would have on the RON architecture. T1, T2, and T3, are composed of 8 inverters, 12 inverters, and 25 inverters, respectively. 8 combinational gates consisting of AND, INV, and OR constitute T4, while T5 and T6 are comprised of 25 and 22 combinational gates, respectively. We observed that in T1, T2, and T3, the oscillation cycle count difference of RO8 increased with Trojan size from -31 (for T1) to -59 (for T3). This occurred due to the greater power supply noise imparted from the Trojan gates. As the power supply voltage is lowered, the speed of the ring oscillator is dropped. For T4, T5, and T6, we observed similar results. In general, the greater the size of the Trojan, the larger impact it can

have on the power supply network and consequently the greater impact on the ring oscillators.

3.4.2 Trojan Switching Activity Analysis

Trojan size is not the only parameter impacting the frequency of the ring oscillators.The Trojan switching activity plays an important role as well. In the interest of simulation running time, we designed few Trojans featuring frequent switching activities; e.g., T1, T2, and T3 switch 760 times, 1140 times, and 2375 times respectively during the pattern application period. T4, T5, and T6 switch 665 times, 2090 times and 1850 times during the simulation. From the Table 3.1, one can notice the trend: the more frequently the Trojan switches, the greater the voltage drop imparted on the ring oscillator gates, which in turn, impacts oscillation cycle count reported by the ring oscillator.

3.4.3 Process Variations Analysis

Random process variations, consisting of 10% voltage threshold (8% inter-die and 2% intra-die), 3% oxide thickness (2% inter-die and 1% intra-die), and 10% channel length (8% inter-die and 2% intra-die) in 90nm technology library, are used in the following simulations. All the simulations are done under temperature 25 °C. 100 Trojan-free ICs and 600 Trojan-inserted ICs (100 per Trojan) are generated by Monte Carlo simulations. The statistical data analysis flow proposed

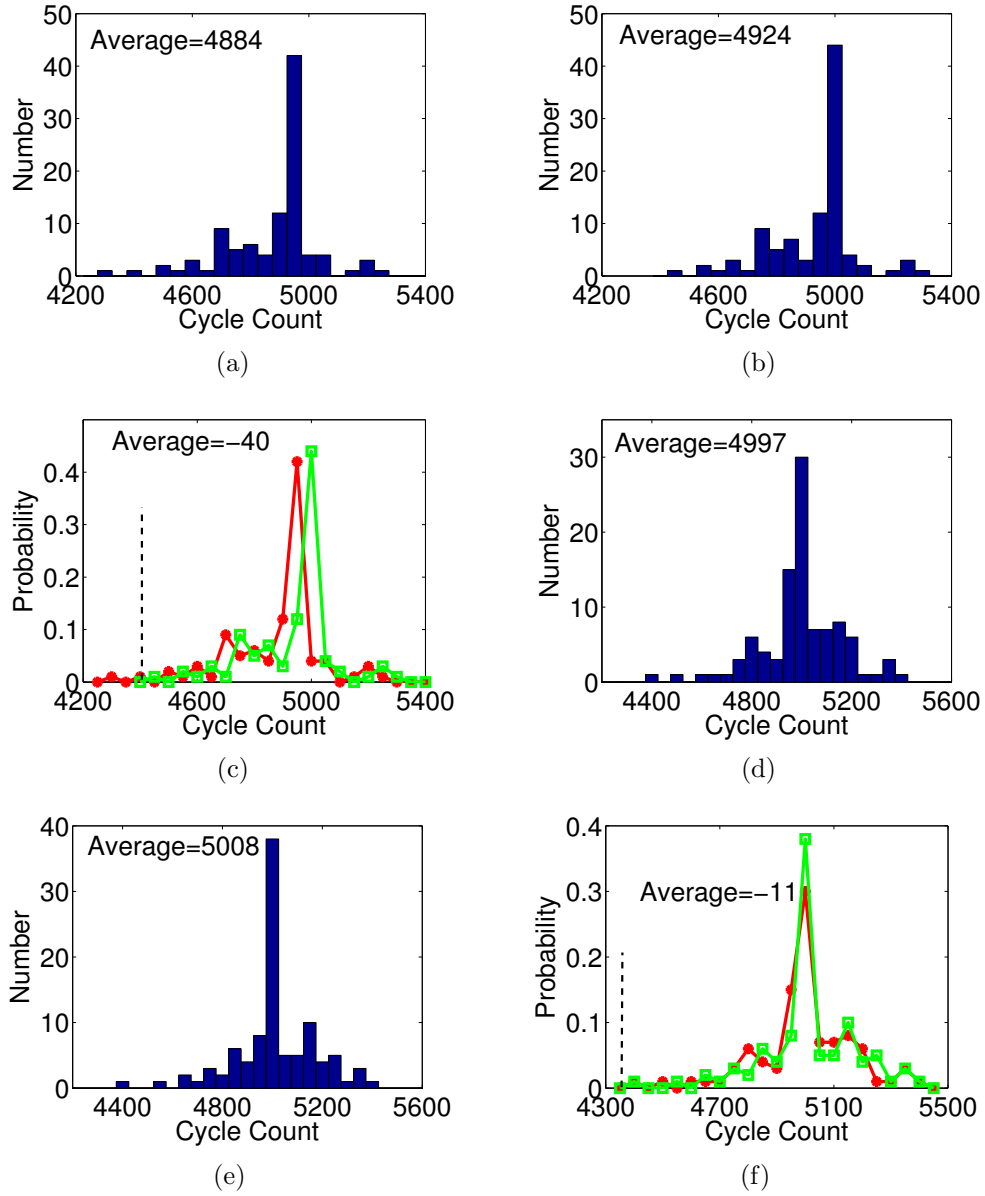


Fig. 3.7: Oscillation cycle distribution of RON with 100 Monte Carlo simulations when T5 is inserted in s9234. (a) RO8 with Trojan; (b) RO8 w/o Trojan; (c) Cycle count distribution of RO8; (d) RO5 with Trojan; (e) RO5 w/o Trojan; (f) Cycle count distribution of RO5.

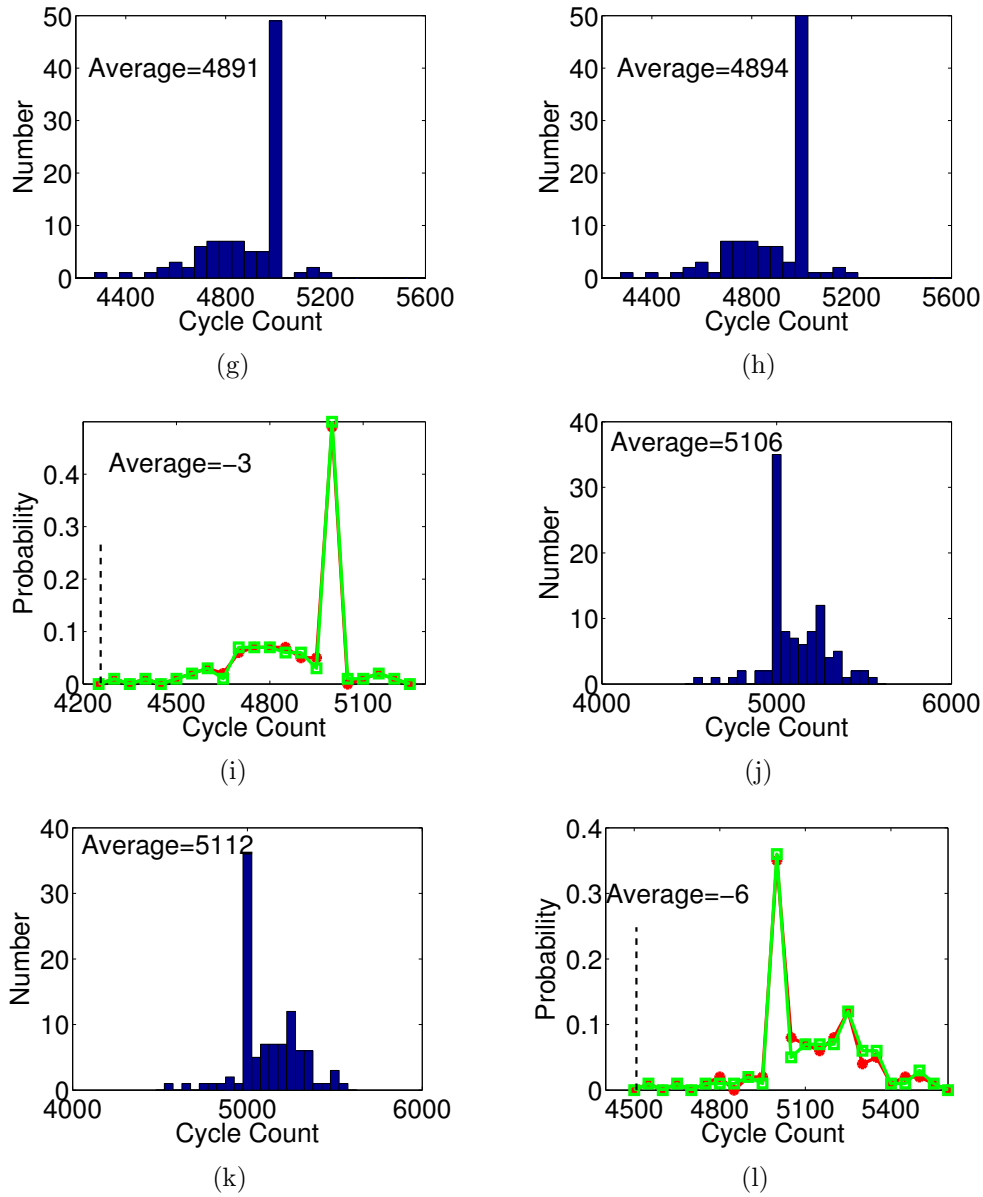


Fig. 3.7: Oscillation cycle distribution of RON with 100 Monte Carlo simulations when T5 is inserted in s9234. (g) RO1 with Trojan; (h) RO1 w/o Trojan; (i) Cycle count distribution of RO1; (j) RO12 with Trojan; (k) RO12 w/o Trojan; (l) Cycle count distribution of RO12.

in the previous section processed the data collected from these ICs. T5 is used to show the detailed results of the data analysis flow.

Simple outlier analysis is first applied to distinguish the effect of Trojan and process variations. Histograms obtained from RO1, RO5, RO8, and RO12 are shown in Figure 3.7, each showing the distribution of oscillation cycle count plotted from the data obtained in the presence of process variations with T5. Figure 3.7(a) displays the histogram of the cycle count of oscillations reported by RO8 with the Trojan inserted and Figure 3.7(b) shows the same result without (w/o) the Trojan. The distribution of the two sets of oscillation cycle count are plotted in Figure 3.7(c). The remaining figures (3.7(d)-3.7(l)) show the data distribution collected from RO5, RO1, and RO12, respectively. We do not notice a significant change in RO5, RO1, and RO12. However, due to the presence of T5, RO8's distribution shifts toward left considerably. For RO8, the oscillation cycle range is $4400 - 5350$ in Trojan-free ICs and the boundary is marked by the black dashed line in Figure 3.7(c). 3 ICs out of the 100 ICs under authentication fell outside of the range, which are identified to contain Trojan.

For the remaining 97 ICs, PCA is done to analyze the data. Figure 3.8 shows the power signature comparison using PCA for Trojan detection. The convex is drawn from the first three principal components with Trojan-free ICs. The asterisks denote data obtained from ICs with the inserted Trojan, which are shown to be separate from the convex hull. Thus, with the RON architecture and

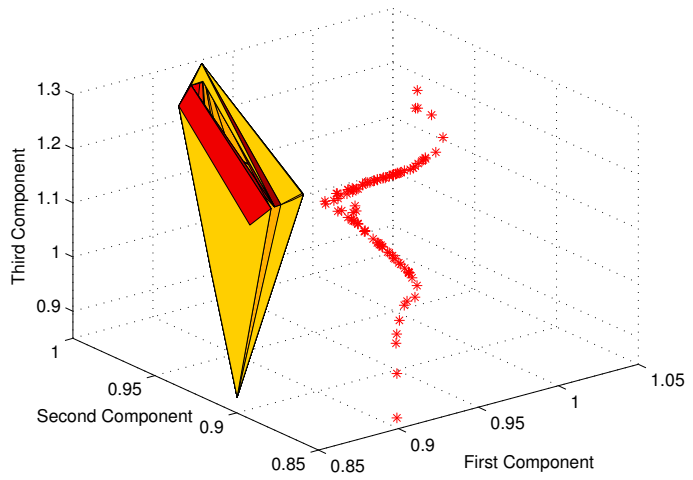


Fig. 3.8: Power signature using PCA for Trojan-free ICs and Trojan-inserted ICs with T5.

statistical analysis, T5 can be detected with 100% accuracy. However, limited by the statistical methods and the increasingly larger process variations of nano-scale technologies, *smaller Trojans* may not necessarily be detected with such accuracy.

Thus, advanced outlier analysis shown in Figure 3.5 is also used to identify Trojan-inserted ICs. There are a total of 12*11=132 power signatures generated by the Trojan-free ICs. In the following, for advanced outlier analysis results, only the power signature that can detect the most Trojan-inserted ICs is shown. As an example, for T5, Figure 3.9(e) shows the advanced outlier analysis result. The blue dots represent Trojan-free ICs and the red asterisks denote Trojan-inserted ICs. We can see that all of the Trojan-inserted ICs are outside of the Trojan-free ICs. Thus, the detection rate with T5 using advanced outlier analysis is 100%.

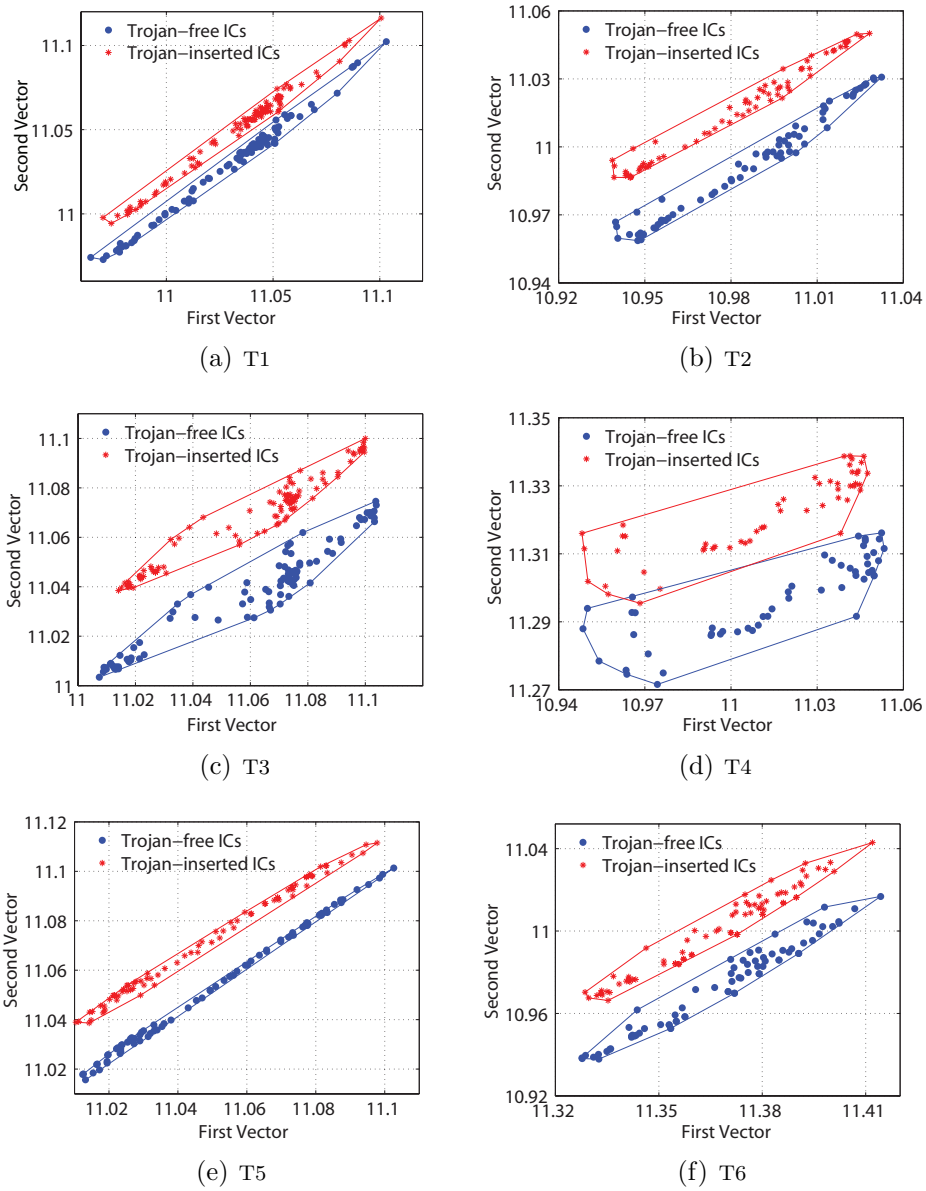
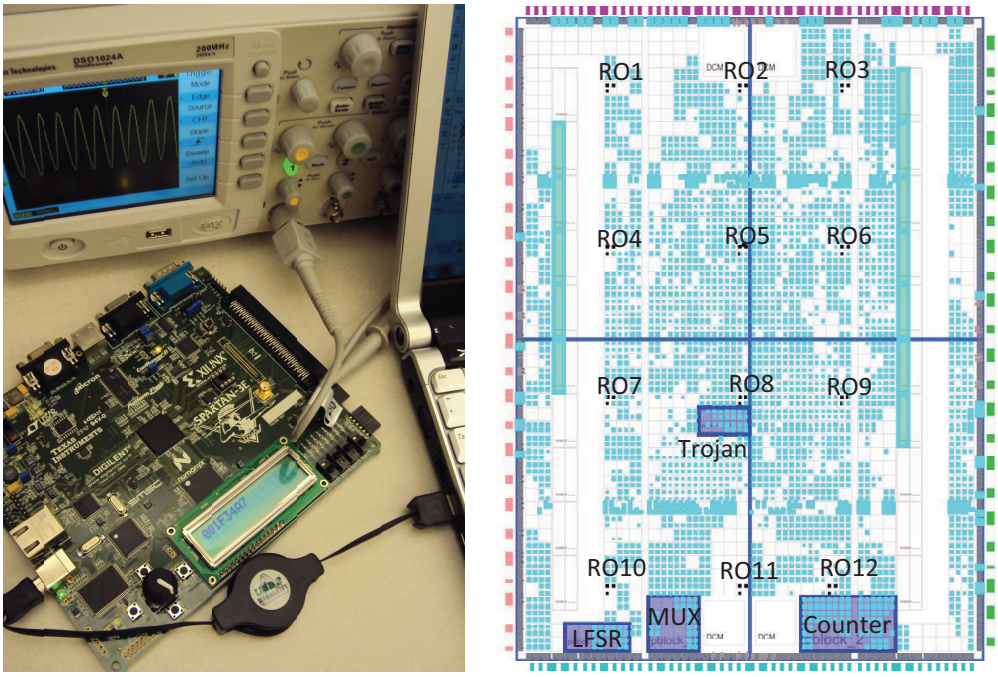


Fig. 3.9: Power signatures with advanced outlier data analysis from IC simulation.

Similarly, the remaining 5 Trojans (T1, T2, T3, T4, and T6) with 100 Trojan-free ICs and 100 Trojan-inserted ICs are also simulated and the data analysis flow is applied for every Trojan. By simple outlier analysis, one Trojan-inserted IC is detected with T1, T2, and T4 and two Trojan-inserted ICs are identified with T3 and T6. Using PCA, Trojan-inserted ICs detected with T1, T2, T3, T4, and T6 are 16, 17, 8, 10, and 29, respectively. The remaining Trojan-inserted ICs are analyzed by advanced outlier analysis, shown in figures 3.9(a) - 3.9(f). In order to show the effectiveness of RON when using our advanced outlier analysis, the Trojan-inserted ICs detected by simple outlier analysis and PCA are also plotted in these figures. Combined simple outlier analysis, PCA, and advanced outlier analysis, the Trojan detection rates for T2, T3, and T6 are 100%. For smaller Trojan T1, the detection rate is 100%, even though the Trojan-inserted ICs are so close to the Trojan-free ICs. For T4, 98% Trojan-inserted ICs are detected. Note that the detection rates presented above are all only from the best distributions selected from 132 power signatures. When we analyze all power signatures, the detection rate for all Trojans including T4 is 100%.

3.4.4 Validation on Spartan-3E FPGA

The same RON architecture is applied to AES implemented on Xilinx Spartan-3E FPGA (shown in Figure 3.10(a)). Three Trojans (T7, T8, and T9) are inserted into the benchmark. T9 consists of 80 gates. The area overhead of T7 is 0.17%,



(a) (b)

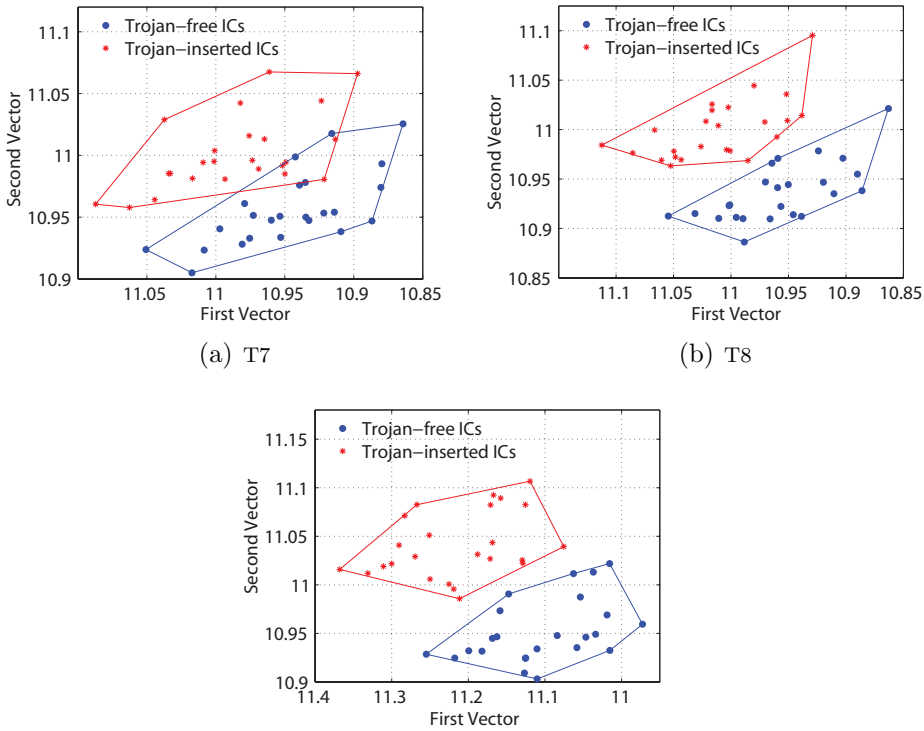
Fig. 3.10: (a) Xilinx Spartan-3E FPGA board and (b) AES layout after placement.

T8 is 0.25%, and T9 is 0.33%. 24 Trojan-free FPGAs and 24 Trojan-inserted FPGAs are used. The oscillation cycle count from different FPGAs represent inter-die process variations and the oscillation cycle count from the same FPGA but different ring oscillators denote intra-die process variations.

The layout of FPGA after the placement and routing is shown in Figure 3.10(b). 12 ring oscillators with five inverters constitute RON while Trojans are placed near RO8. LFSR module generates patterns during authentication process. Multiplexer module selects which ring oscillator would be enabled and recorded. The implementation temperature is 25 °C. Several measurements are done for each ring oscillator in every FPGA in order to eliminate the measurement noise, and the average value is used to perform data analysis.

One Trojan-inserted FPGA is detected by simple outlier analysis for each Trojan. PCA detects 9 Trojan-inserted FPGAs with T7, 10 Trojan-inserted FPGAs with T8, and 16 Trojan-inserted FPGAs with T9. The remaining Trojan-inserted FPGAs are analyzed by advanced outlier analysis (shown in Figure 3.11). In order to show all the detected Trojan-inserted FPGAs, the FPGAs detected by simple outlier analysis and PCA are also plotted in these figures. From combined simple outlier analysis, PCA, and advanced outlier analysis, 100% Trojan-inserted FPGAs are detected for T8 and T9 but 80% for T7 from the best selected power signature. Performing similar analysis for all power signatures, we are able to increase the Trojan detection rate to 92% for T7. In addition, we also imple-

mented Trojans of smaller size (T10=30 gates and T11=20 gates) to verify the sensitivity of RON. Trojan detection rate is 92% for T10 but 100% for T11 since it experiences more switching activity.



(a) T7 (b) T8

(c) T9

Fig. 3.11: Advanced outlier analysis results from FPGA implementation.

3.5 Conclusions

In this chapter, we presented an effective structure to detect hardware Trojans inserted into an IC. The RON architecture generates a power supply fingerprint, used to identify malicious alterations. Statistical analysis distinguishes the effects of hardware Trojans from process variations. The experimental results demon-

strate that this approach is very effective in identifying Trojan-inserted ICs. However, for Trojans with very small number of gates, more work needs to be done. We will present our improved technique in Chapter 4.

Chapter 4

Detection of Trojans using Combined Ring Oscillator Network and Off-chip Transient-Power Analysis

Based on the experimental results and analysis in Chapter 3, we can see that the sensitivity of using RON only is limited. It is very difficult to be detect Trojans with a very small number of gates. In order to improve the effectiveness of our method, we propose a novel hardware Trojan detection method performed by combining measurements from RON with external dynamic current measurements. It monitors power fluctuations and differentiates fluctuations due to hardware Trojans from fluctuations due to measurement noise and process variations. This method considers Trojans' impact on the power consumption of neighboring cells and the entire IC. Each row of the circuit under authentication contains at least one inverter of an RO in the RON. Thus any malicious inclusions in each row would be captured by one of these ring oscillators. Off-chip test equipment will measure the transient current of the IC, which will be combined with the ROs' cycle counts to generate a power signature for the entire IC. The signature of the

CUT is then compared against the Trojan-free signatures.

Comparing the new combined on- and off-chip measurement method with the method we proposed in Chapter 3, we have improved our method in the following ways: (*i*) the new Trojan detection method takes into account a Trojan's impact on neighboring cells; (*ii*) the new proposed architecture ensures the placement of ring oscillator components in every row in a standard-cell design, the most widely used design style in practice; (*iii*) high threshold voltage gates are used in the new on-chip sensors to improve their sensitivity to the noise induced by Trojans; (*iv*) the circuit's total transient current is taken into account.

The ring oscillators will be disabled when the IC operates in functional mode to reduce the power consumption. The number of ring oscillators, N_{ro}, can be adjusted according to the size of the IC and the desired sensitivity to Trojans. Simulation and FPGA implementation results demonstrate that the proposed method effectively distinguishes the power differences caused by Trojans from those caused by process variations. We will present this technique in detail in this chapter.

4.1 The Relationship between RO Frequency and Localized and Total Dynamic Current

The delay of each inverter in the ring oscillator can be expressed as $t_d = k_g/I_g$ where k_g is a gate-dependent constant and I_g is the dynamic current of the inverter

[45]. Based on the Alpha-Power Model mentioned in [46], the dynamic current of a switching gate is

$$I = \mu_g * (V_{dd} - V_{th})^{\alpha} \tag{4.1}$$

where α is the velocity saturation index. Thus the frequency of the n-stage ring oscillator can be expressed as:

$$f = \frac{\mu_g * (V_{dd} - V_{th})^{\alpha}}{2n * k_g} \tag{4.2}$$

In the presence of a Trojan, the ring oscillator frequency is modeled by Equation 4.3 rather than Equation 4.2 where the voltage-drop ΔV_t represents the Trojan's contribution to the ring oscillator frequency. From the equation, we can see that the frequency of the ring oscillator f_t is more sensitive to the voltage-drop ΔV_t when the stage of the ring oscillator n is smaller. However, if n is too small, the frequency of the ring oscillator will be too high to be measured by on-chip counter in practice. Using (i) an operating frequency of f=1GHz, (ii) V_{dd} = 1.2V, and (iii) Synopsys 90nm technology in a Nanosim simulation, we have determined that a 5-stage RO would be the smallest allowable RO. Thus, 5-stage ring oscillators will be used throughout this thesis.

$$f_t = \frac{\mu_g * (V_{dd} - \Delta V_t - V_{th})^{\alpha}}{2n * k_g} \tag{4.3}$$

The dynamic current of the entire Trojan-free chip is:

$$I_{total} = \sum_{i=0}^{i=N} \lambda_i * N * \mu_g * (V_{dd} - V_{th})^{\alpha} \tag{4.4}$$

where N is the total number of switching gates in the IC and λ_i denotes the gate-dependent constant of the i_{th} gate. The constant, λ_i, depends only on the type of gate specified, not the particular instance of such a gate. The relationship between the frequency of the n-stage ring oscillator embedded into the chip and the dynamic current of entire chip will be:

$$\frac{I_{total}}{f} = \sum_{i=0}^{i=N} \lambda_i * N * 2n * k_g \tag{4.5}$$

For ICs with n_t Trojan gates inserted, Equation 4.5 becomes:

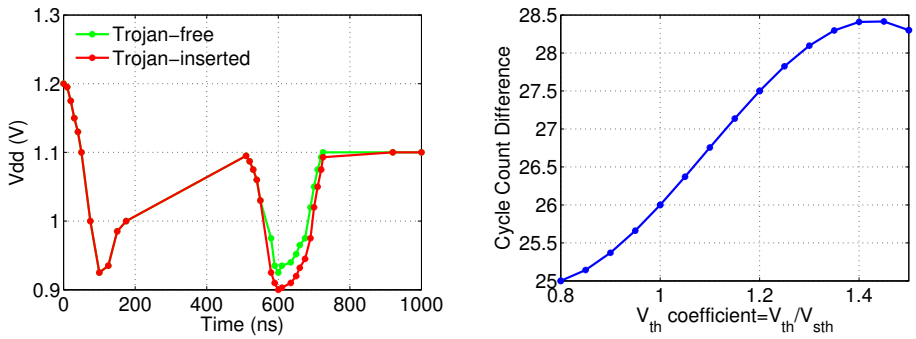
$$\frac{I_{total,t}}{f_t} = \sum_{i=0}^{i=N+n_t} \lambda_i * (N + n_t) * 2n * k_g(1 + \frac{\Delta V_t}{V_{dd} - \Delta V_t - V_{th}})^{\alpha} \tag{4.6}$$

Since $\Delta V_t << V_{dd} - \Delta V_t - V_{th}$, the above equation can be approximated as Equation 4.7 based on Taylor's expansion theorem.

$$\frac{I_{total,t}}{f_t} \approx \sum_{i=0}^{i=N+n_t} \lambda_i * (N + n_t) * 2n * k_g(1 + \alpha * \frac{\Delta V_t}{V_{dd} - \Delta V_t - V_{th}}) \tag{4.7}$$

Comparing Equation 4.7 with Equation 4.3, we can see that combining ring oscillator frequency measurements with current measurements will achieve greater sensitivity to Trojans than either measurement alone.

All the above analysis are based on ring oscillators made with standard threshold voltage (SVT) transistors. However, ring oscillators with high threshold voltage (HVT) transistors are more sensitive to power supply noise as shown by the simulation results in Figure 4.1, performed using the same technology with a 5-stage ring oscillator. The green line in Figure 4.1(a) denotes the power supply



(a) (b)

Fig. 4.1: (a) Power supply variations for Trojan-free and Trojan-inserted circuits; (b) Cycle count difference increases as threshold voltage increases.

voltage of Trojan-free ICs during the 1000ns simulation period while the red line represents the power supply voltage of Trojan-inserted ICs. Figure 4.1(b) shows that for a particular ring oscillator, the cycle count difference between Trojan-free ICs and Trojan-inserted ICs will increase as the threshold voltage of the transistors increases until a maximum is reached. Once this maximum has been reached, increasing the threshold adversely affects the cycle-count difference (and thus the sensitivity to inserted Trojans). The X axis in Figure 4.1(b) represents the threshold voltage coefficient, V_{th}/V_{sth}, where V_{sth} is the SVT of the MOS transistors. In the Synopsys 90nm technology library, the threshold voltage coefficient of the HVT transistors is 1.2. With HVT ring oscillators, Equation 4.7 becomes:

$$\frac{I_{total,t}}{f_t} = \sum_{i=0}^{i=N+n_t} \lambda_i * (N + n_t) * 2n * k_g(1 + \alpha * \frac{\Delta V_t + V_{hth} - V_{sth}}{V_{dd} - \Delta V_t - V_{hth}}) \tag{4.8}$$

where V_{hth} is the high threshold voltage of the transistors in the ring oscillators.

From Equation 4.8, we can tell that the relationship between the IC's dynamic current and the frequency of a ring oscillator in the circuit will be more sensitive using HVT transistors. In addition, we can conclude that Trojans with larger size (n_t) and more IR-drop (ΔV_t) are easier to be detected.

Some of the parameters in Equation 4.8 will change with process and environmental variations. In this thesis, we assume that ICs are tested under the same temperature condition in a production test environment, thus, only small environmental variations will be considered in this thesis. All remaining parameters are susceptible to process variations and we will use statistical analysis to separate the contributions of process variations and Trojans to the transient power.

4.2 Improved Ring Oscillator Network Structure

As aforementioned, Trojan gates' switching impacts both the frequency of nearby ring oscillators and the IC's dynamic current. Since a Trojan's effects may be localized (i.e. tightly distributed), and the impact of a Trojan on a ring oscillator is dependent upon the distance between them, one ring oscillator may not be sensitive enough to distinguish the effects of Trojans from process variations throughout the entire IC. A improved RON, however, can improve the sensitivity to Trojan noise.

Figure 4.2 shows a circuit into which the proposed on-chip structure with N_{RO} n-stage ring oscillators is inserted. These n-stage ring oscillators are each

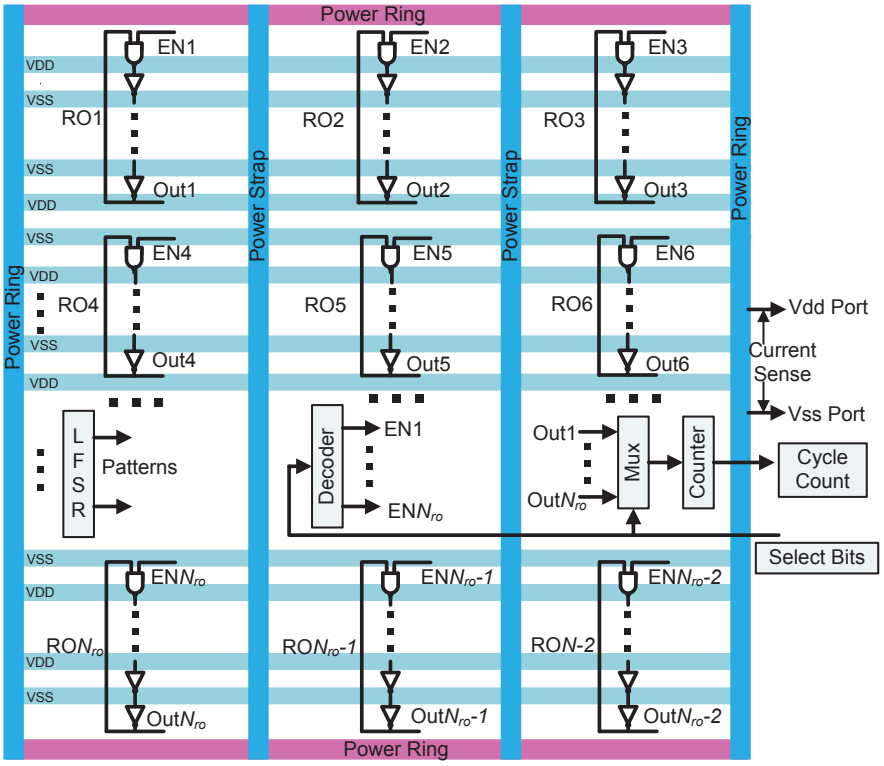


Fig. 4.2: Our improved on-chip structure with each gate of the ring oscillators placed in a standard cell row.

composed of one NAND gate and $n-1$ inverters with one component located in each of the n rows of the standard cell design. The ring oscillators are more sensitive to the voltage drop caused by a Trojan if they share the same power strap. Therefore, it is highly advantageous to ensure complete coverage of the power distribution network by placing at least one ring oscillator component in each row of the standard cell (and thus near each power strap). One set of n-stage ring oscillators will be inserted between two vertical straps. If there are M vertical power straps and R rows in the design, $N_{RO} = (M+1)\lceil R/n \rceil$. However, the number of ring oscillators could be adjusted according to the required Trojan detection sensitivity and the minimum sensitivity to Trojan activity. The linear feedback shift register (LFSR), decoder, multiplexer, and counter are the same as those in Chapter 3.

Since the ring oscillators are only enabled during the production test and the authentication phase, their power overhead in the field is negligible. The proposed architecture has a small area overhead, due mainly to the ring oscillators. For larger circuits, assuming there is one vertical power strap for every 20 FFs or 80 gates, the area overhead of the ring oscillators will be approximately $1/(20*4) = 1.25\%$. The total area overhead will be approximately 2.5% if there is one vertical strap for only every 10 FFs or 40 gates in the design. For a small circuit, the counter may play a significant part in the area overhead, but the counter size does not increase linearly with the size of the circuit. Since LFSRs are commonly

used for built-in self-tests in modern designs, it can be ignored when analyzing the area overhead. However, even with LFSRs, the area overhead of a RON in large designs is still quiet small since the area of the LFSRs does not increase significantly with the size of the circuit, either. Transient current will be measured externally (i.e. with no area cost). In summary, the area overhead will be less than 3% for a large circuit and would be slightly larger for a smaller circuit. For instance, the overhead of RONs with LFSRs is 5.58% for the ISCAS'89 benchmark circuit s38584 (which contains four vertical power straps), 2.47% for a AES circuit (with six vertical straps), and 1.99% for a DES circuit (with six vertical power straps). The AES and DES circuits are provided in [31].

Since the ring oscillators are distributed across the entire IC, it is inherently difficult for an adversary to remove or tamper with one. If one of the ring oscillators reports data outside of a certain range or does not report data, we will assume it has been attacked. In addition, this proposed on-chip structure is resilient to modeling. Some attackers may build up a lookup table to repeatedly generate the same cycle count for each ring oscillator, which would attempt to replace the Trojan-effected counter values with known good values. However, the current consumed by the lookup tables may be captured by the external current measurement and the power signature generated by our outlier analysis would also be changed. On the other hand, if the ROs are replaced with lookup tables embedded in the design, the frequency of the same ring oscillator at the same

location, but on a different chip would stay at the same value in different ICs. However, unlike the value stored in a lookup table, the measured frequency of an RO in different ICs should be slightly different due to different process variations in Trojan-free circuits. If one ring oscillator in all CUTs has the exactly same frequency, designers would easily know that the IC was tampered with embedded lookup tables.

4.3 Measurement Flow and Statistical Analysis

The measurement flow for each IC is shown in Figure 4.3. To measure the frequency of N_{RO} ring oscillators, we apply the LSFR patterns with the same seed N_{RO} times. The transient current is measured externally. A signature is generated by recording the cycle count of each ring oscillator and the transient current from a large number of ICs of the same design. Since the ICs will all be subject to different process variations, this signature can be statistically more tolerant to similar variations in chips under authentication. In order to separate the effect of process variations and Trojans, a data analysis flow is suggested in this work which includes three methods namely: (i) Simple Outlier Analysis, (ii) Principal Component Analysis (PCA), and (iii) Advanced Outlier Analysis. This data analysis flow is very similar with the data analysis flow we proposed in Chapter 3.

Principal component analysis method is slightly different from the PCA we

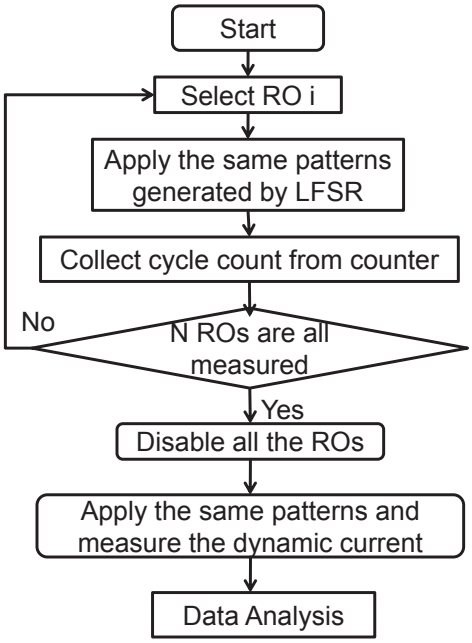


Fig. 4.3: Measurement flow of our proposed method.

used in Chapter 3. With one variable representing one ring oscillator, there are N_{RO} variables and the $N_{RO} + 1^{th}$ variable represents the dynamic current. The relationship between the data from the N_{RO} ring oscillators and the dynamic current is considered by PCA when it transforms the $N_{RO} + 1$ variables into uncorrelated variables. The $N_{RO} + 1$ variables are transformed by PCA and the first three of the resulting components in Trojan-free ICs are used to construct a convex hull [44]. If the output of the CUT is beyond the convex, a Trojan must exist in the IC under authentication. However, if the output is inside the convex, advanced outlier is used for further analysis and validation.

Advanced outlier analysis considers the relationships among ROs in the RON and the dynamic current of the entire chip. The pseudo-code is shown in

Power Signature Generation

01: **Collect** data from N_{TF} Trojan-free ICs with N_{RO} ring oscillators

02: **for** $(i = 1, i <= N_{RO}, i++)$ { //select $i^{th} RO$

03: **for** $(j = 1, j <= N_{RO}, j++)$ $(j \neq i)$ { //select $j^{th} RO$

04: **for** $(k = 1, k <= N_{TF}, k++)$ { //select k^{th} Trojan-free IC

05: x_{ki}=$(\sum_{m=1}^{N_{RO}} CC_{km} - CC_{ki})/CC_{ki}$;

06: y_{kj}=$(\sum_{m=1}^{N_{RO}} CC_{km} - CC_{ki})/CC_{kj}$;

07: **plot**(x_{ki}, y_{kj}, I_K);

08: }

09: The power signature, PS_{ij}, is created from all N_{TF} ICs.

10: } //x_{ki} is named as the first vector

11: } //y_{kj} is named as the second vector

Note: CC_{km}, CC_{ki}, CC_{kj}: Oscillation cycle count of RO_m, RO_i, RO_j in k^{th} IC

I_k: The dynamic current of kth IC

(a)

Authentication

For each IC under authentication:

01: **Collect** data from N_{RO} ring oscillators ($CC_{RON} = \sum_{i=1}^{N_{RO}} CC_i$).

02: **for** $(i = 1, i <= N_{RO}, i++)$ {

03: **for** $(j = 1, j <= N_{RO}, j++)$ $(j \neq i)$ {

04: $x = (CC_{RON} - CC_i)/CC_i$;

05: $y = (CC_{RON} - CC_i)/CC_j$;

06: **plot**(x,y,I);

07: **if** ((x,y) is outside of the power signature PS_{ij}

08: {The IC is Trojan-inserted; Break; }

09: **else** go on;

10: }

11: }

Note: CC_i, CC_j: Oscillation cycle count of RO_i, RO_j

I: Dynamic current of the under test IC

(b)

Fig. 4.4: Advanced outlier analysis procedure.

Figure 4.4. For each Trojan-free IC, two out of N_{RO} ring oscillators will be selected along with the dynamic current information to generate a power signature (shown in Figure 4.4(a)). For a particular Trojan-free IC, the total oscillation cycle count from the RON is $CC_{RON} = \sum_{m=1}^{N_{RO}} CC_m$. Then, the data from the RO_i (CC_i) and RO_j ($j \neq i$) (CC_j) are selected to calculate $x_i = (CC_{RON} - CC_i)/C_i$ and $y_j = (CC_{RON} - CC_i)/CC_j$. Finally, ($x_i$, y_j, I) from all the Trojan-free ICs would be used to generate the power signature, PS_{ij}. There will be $N_{RO} \times (N_{RO} - 1)$ unique power signatures in total. The same process will be applied to the CUT (shown in Figure 4.4(b)). If the CUT lies within the signature it may be assumed that the circuit is Trojan-free. Otherwise, if one of the N_{RO}*$(N_{RO} - 1)$ signatures does not match the Trojan-free signature, it will be treated as a suspicious part, i.e., Trojan-inserted.

4.4 Experimental Results and Analysis

We implemented our proposed approach on a small s9234 benchmark using Synopsys 90nm technology, and a larger circuit, AES benchmark, on Xilinx Spartan-6 FPGAs (45nm technology). For IC simulation, the s9234 benchmark was designed with two vertical power straps and 35 rows, with $N_{RO} = 15$ ring oscillators constituting the on-chip structure. Twenty Trojans (T_1 to T_{20}) with different sizes, gates types, and physical distributions were inserted into s9234. Table 4.1 shows these twenty Trojans where FF represents a flip-flop, $Cen.$ indicates that the Trojan

Table 4.1: Twenty Trojans inserted in s9234 circuit.

	Combinational Trojans									
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Sizes	2 gates	3 gates	4 gates	5 gates	7 gates	8 gates	10 gates	12 gates	16 gates	20 gates
Area Overhead	0.09%	0.16%	0.2%	0.25%	0.37%	0.43%	0.5%	0.66%	0.82%	0.92%
Distribution	Cen.	Cen.	Dis.	Dis.	Cen.	Dis.	Dis.	Cen.	Dis.	Dis.

	Sequential Trojans									
	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}
Sizes	2 FFs	3 FFs	4 FFs	5 FFs	6 FFs	7 FFs	8 FFs	10 FFs	12 FFs	16 FFs
Area Overhead	0.41%	0.62%	0.81%	0.98%	1.18%	1.4%	1.61%	1.83%	2%	2.21%
Distribution	Cen.	Cen.	Dis.	Dis.	Cen.	Dis.	Dis.	Dis.	Dis.	Dis.

is centrally located, and *Dis.* indicates that the Trojan is physically distributed (shown in Figure 4.5). Ten combinational Trojans (T_1-T_{10}) tap internal signals working as comparators and the sequential Trojans (T_{11}-T_{20}) act as shift registers. None of these Trojans were detected by a test suite made up of 80,000 random functional patterns and 206 structural patterns (created by ATPG tools) for detecting stuck-at and transition delay faults. StarRC was used to extract parasitic parameters from the layout of benchmarks and generate SPICE files. A Monte Carlo simulation (performed with Synopsys Nanosim) was used to emulate the effects of process variations which impact the frequencies of the ring oscillators and the dynamic current. The simulation temperature was 25°C with $\pm 5°C$ vari-

Table 4.2: Oscillation cycle count of some of the ring oscillators and circuit dynamic current in presence of hardware Trojans without process variations.

		T_1			T_3			T_6			T_{10}		
		TF	TI	ΔT	TF	TI	ΔT	TF	TI	ΔT	TF	TI	ΔT
Average Dynamic Current (μA)		29.8	29.84	0.04	25.56	25.65	0.09	24.94	25.08	0.14	24.94	26.1	1.16
RO (CC)	RO8	2790	2787	-3	3396	3392	-5	3064	3054	-10	3064	3024	-40
	RO7	3021	3021	0	3528	3528	-2	3008	3005	-3	3008	2998	-10
	RO1	2952	2952	0	3377	3377	0	2985	2984	-1	2985	2982	-3
	RO15	3103	3103	0	3406	3406	0	2803	2803	0	2803	2801	-2

		T_{11}			T_{16}			T_{20}		
		TF	TI	ΔT	TF	TI	ΔT	TF	TI	ΔT
Average Dynamic Current (μA)		27.48	27.6	0.12	23.14	23.77	0.63	26.85	29.05	2.25
RO (CC)	RO8	3150	3141	-9	3120	3084	-36	3031	2972	-59
	RO7	3117	3117	-0	3158	3150	-8	2925	2914	-11
	RO1	3042	3042	0	3198	3198	0	2980	2977	-3
	RO15	3132	3132	0	3210	3210	0	3012	3011	-1

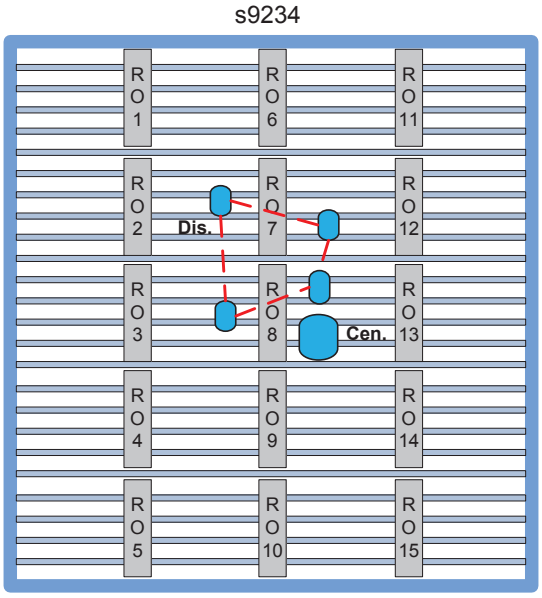


Fig. 4.5: s9234 with 15 ROs and 20 Trojans. One Trojan at a time is inserted into the circuit.

ations. For hardware validation, eight Trojans (T_{21}-T_{28}) with different gates and distributions were inserted into an AES benchmark. Trojan-inserted and Trojan-free versions of the AES benchmark were both implemented on multiple FPGAs under room temperature; the use of multiple FPGAs allowed us to analyze the effects of both inter-die and intra-die process variations.

4.4.1 Effectiveness Demonstration

Trojan Size and Distribution Analysis: Using a simulation without variations, the detailed cycle count and dynamic current results of T_1-T_3, T_6, T_7, and T_{12} with four ring oscillators (RO1, RO7, RO8, and RO15) during a 1000-clock cycle

LFSR test are shown in Table 4.2. Since the IC's dynamic current varies with the test pattern applied, the waveform of the dynamic current is recorded during the simulation. Average dynamic current in the measurement time window is shown in the table as well. In Table 4.2, TF indicates that the data in this column was collected from Trojan-free ICs while TI denotes data from Trojan-inserted ICs. ΔT represents the difference between the Trojan-inserted ICs and the Trojan-free ICs. From Table 4.2, we can see that the Trojans consume extra power, increase the dynamic current, and decrease the cycle count of the ring oscillators.

Table 4.2 shows that T_1, T_3, and T_{11} have a larger impact on the oscillation frequency of RO8 than the other ring oscillators. Similarly, for T_6, T_{10}, T_{16} and T_{20}, there is a larger impact on RO8 and RO7 than RO1 and RO15. This phenomenon is explained by the power supply voltage's dependence on the voltage division coefficient which is partially determined by the distance (resistive path) between two gates; smaller distance implies greater Trojan impact on ring oscillators. The remaining Trojans not shown in Table 4.2 exhibit similar behavior for ring oscillators. However, the total dynamic current current does vary with the distributions of Trojans.

From Table 4.2, we observed that in these seven Trojans, the oscillation cycle count difference ΔCC_{ft} of RO8 increased with Trojan size from -3 (for T_1) to -59 (for T_{20}). This occurred due to the greater power supply noise imparted from the Trojan with more gates. The dynamic current difference between Trojan-

free IC and Trojan-inserted IC varies from 0.04 μA to 2.25 μA. Larger Trojans consume more power. For the Trojans that are not shown in the table, we observed similar results. In general, larger Trojans have a greater impact on the power supply network and consequently have a greater impact on the ring oscillators and dynamic current measurements.

Process Variations Analysis: random process variations, consisting of $3\sigma = 10\%$ voltage threshold (5% inter-die and 5% intra-die), $3\sigma = 3\%$ oxide thickness (2% inter-die and 1% intra-die), and $3\sigma = 10\%$ channel length (5% inter-die and 5% intra-die) are used in the following simulations to analyze their impact on our method. 200 Trojan-free ICs and 100 Trojan-inserted ICs for each Trojan are generated by Monte Carlo simulations. The statistical data analysis flow proposed in the previous section was used to process the data collected from these ICs. T_{10}, composed of 20 combinational gates, is used to show the detailed results of the data analysis flow.

Simple outlier analysis is first applied to distinguish the effects of Trojans and process variations. Histograms obtained from RO1, RO7, RO8, and RO15 in Figure 4.6 show the distribution of oscillation cycle counts in the presence of process variations with T_{10}. Figure 4.6(a) displays the histogram of the cycle counts reported by RO8 with the Trojan inserted and Figure 4.6(b) shows the same result without (w/o) the Trojan. The distributions of the two sets of oscillation cycle counts are plotted in Figure 4.6(c). The remaining figures (4.6(d)-4.6(l))

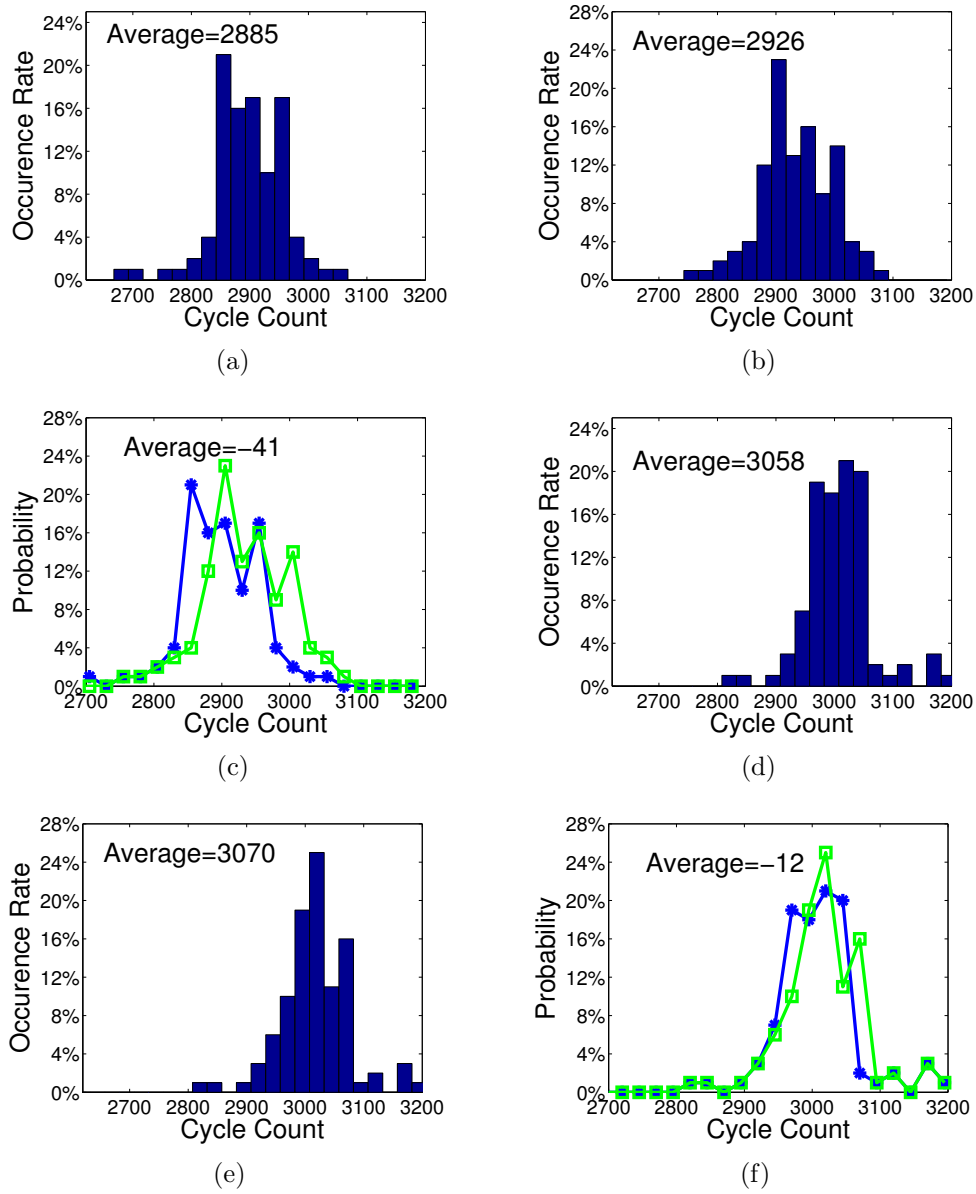


Fig. 4.6: Oscillation cycle distribution of RON with Monte Carlo simulations when T_{10} is inserted in s9234. (a) RO8 with Trojan; (b) RO8 w/o Trojan; (c) Cycle count distribution of RO8; (d) RO7 with Trojan; (e) RO7 w/o Trojan; (f) Cycle count distribution of RO7.

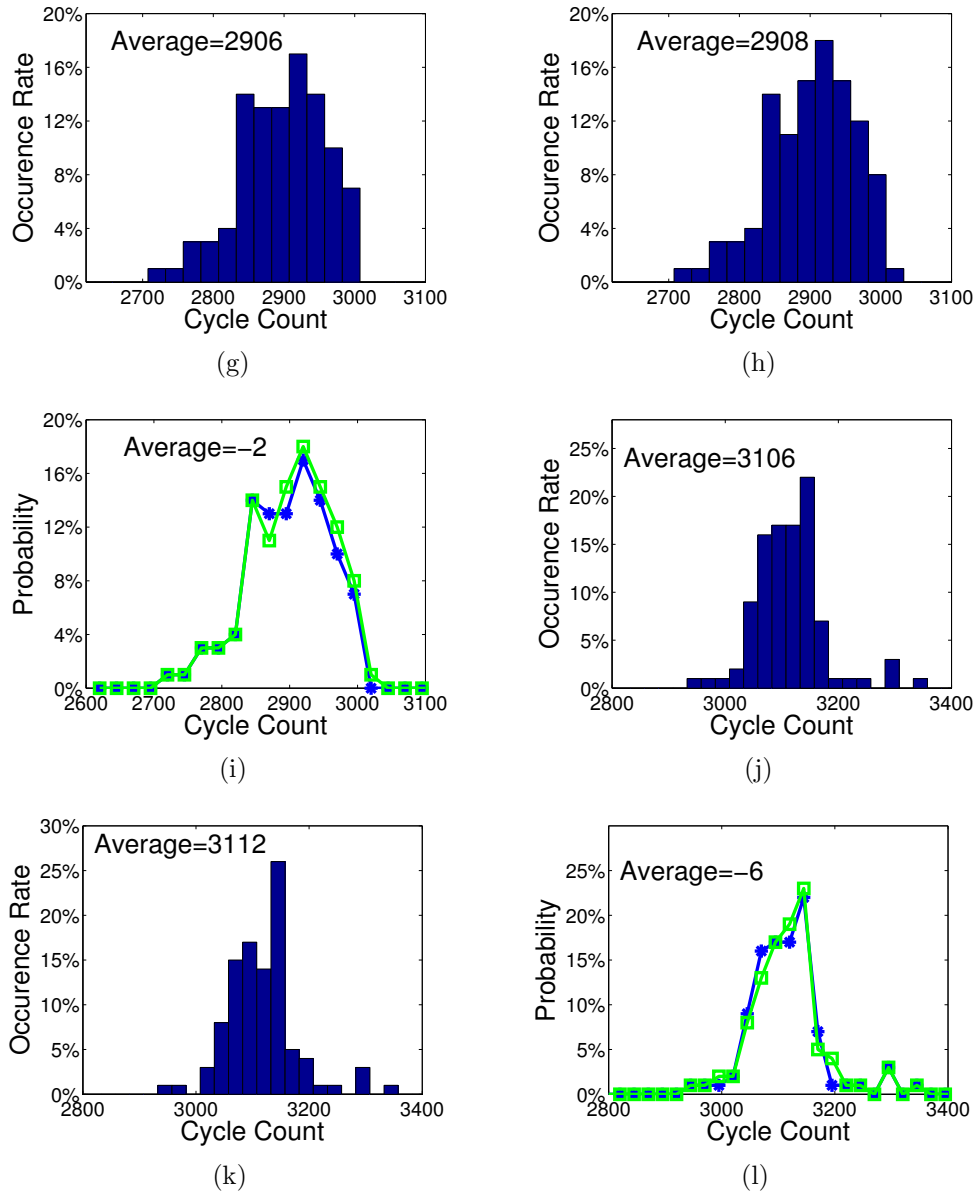


Fig. 4.6: Oscillation cycle distribution of RON with Monte Carlo simulations when T_{10} is inserted in s9234. (g) RO1 with Trojan; (h) RO1 w/o Trojan; (i) Cycle count distribution of RO1; (j) RO15 with Trojan; (k) RO15 w/o Trojan; (l) Cycle count distribution of RO15.

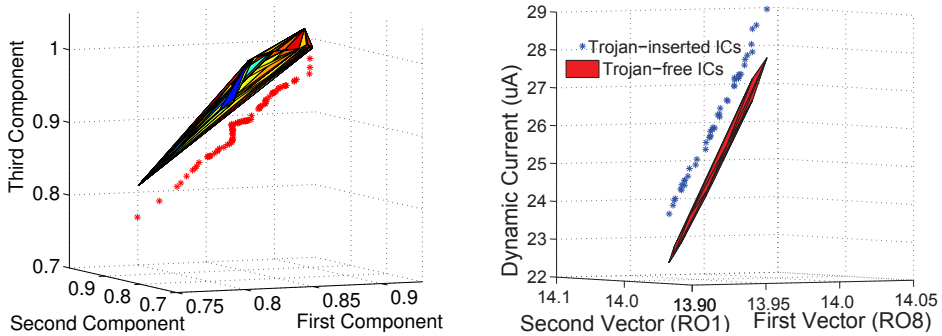
Table 4.3: Trojan detection rates with process variations.

	Combinational Trojans									
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Trojan Detection Rate	75%	80%	86%	100%	100%	100%	100%	100%	100%	100%

	Sequential Trojans									
	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}
Trojan Detection Rate	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

show the data distributions collected from RO7, RO1, and RO15, respectively. We do not notice a significant change in RO7, RO1, and RO15. However, due to the presence of T_{10} which is proximal to RO8, RO8's distribution shifts leftward considerably. For RO8, the oscillation cycle range is $2756 - 3090$ in Trojan-free ICs. 3 ICs out of the 100 ICs under authentication fell outside of the range, which are identified to contain a Trojan.

For the remaining 97 ICs, PCA is done to analyze the data. Figure 4.7(a) shows the power signature comparison using PCA with N_{RO} ring oscillators and dynamic current for Trojan detection. The convex is drawn from the first three principal components with 200 Trojan-free ICs. The asterisks denote data obtained from ICs with Trojans which are shown to be separate from the convex hull. Thus, with the RON architecture and statistical analysis, T_{10} can be detected with 100% accuracy. However, with limited statistical analysis, or if the



(a) (b)

Fig. 4.7: Power signature for Trojan-free ICs and Trojan-inserted ICs with T_{10} using (a) PCA and (b) advanced outlier analysis.

RON is subjected to the increasingly large variations of nano-scale technologies, *smaller Trojans* may not necessarily be detected with such accuracy, which was the case for T_1 to T_8 and T_{11} to T_{17}.

Thus, advanced outlier analysis shown in Figure 4.4 is also used to identify Trojan-inserted ICs. There are a total of 15*14=210 power signatures generated by the Trojan-free ICs. Some power signatures could identify more Trojan-inserted ICs than the others. In the following advanced outlier analysis results, only the power signature that can detect the most Trojan-inserted ICs is shown. Figure 4.7(b) shows the advanced outlier analysis results with Trojan T_{10}. The ring oscillator that was selected as x in Figure 4.4 is defined as the first vector and y as the second vector. The blue dots represent Trojan-free ICs and the red asterisks denote Trojan-inserted ICs. We can see that all of the Trojan-inserted

ICs are outside of the Trojan-free signature. Thus, the detection rate with T_{10} using advanced outlier analysis is 100%.

Similarly, the remaining nineteen Trojans with 200 Trojan-free ICs and 100 Trojan-inserted ICs are also simulated and the data analysis flow is applied for every Trojan. The Trojan-inserted ICs with T_1, T_4, T_{11}, and T_{20} are selected to present detailed results using advanced outlier analysis shown in Figure 4.8(a)-4.8(d). The Trojan detection rates of Trojans T_{11} and T_{20} shown in Figure 4.8 are 100% with only one signature. For T_4, 98% of the Trojan-inserted ICs are detected using one signature shown in Figure 4.8(b). When all 210 power signatures are used, the detection rate for Trojan T_4 is 100%. Complete results for all Trojans using all of the power signatures are shown in Table 4.3. From Table 4.3 and Figure 4.8, we can see that for Trojan T_4-T_{20}, the detection rates are all 100%. The power signatures of the Trojan-free ICs are completely separate from the Trojan-inserted ICs. However, the Trojan-inserted ICs with T_4 are close to the Trojan-free ICs. For the very small Trojans T_1-T_3, the detection rates are less than 100% because of their diminished impact on the power supply lines.

4.4.2 Sensitivity Analysis

Ring Oscillator Number Analysis: Trojans T_1, T_2, and T_3 were chosen for ring oscillator number analysis since their detection rates are less than 100% with N_{RO}=15 ring oscillators. RONs with N_{RO}=10, 20, and 25 ring oscillators

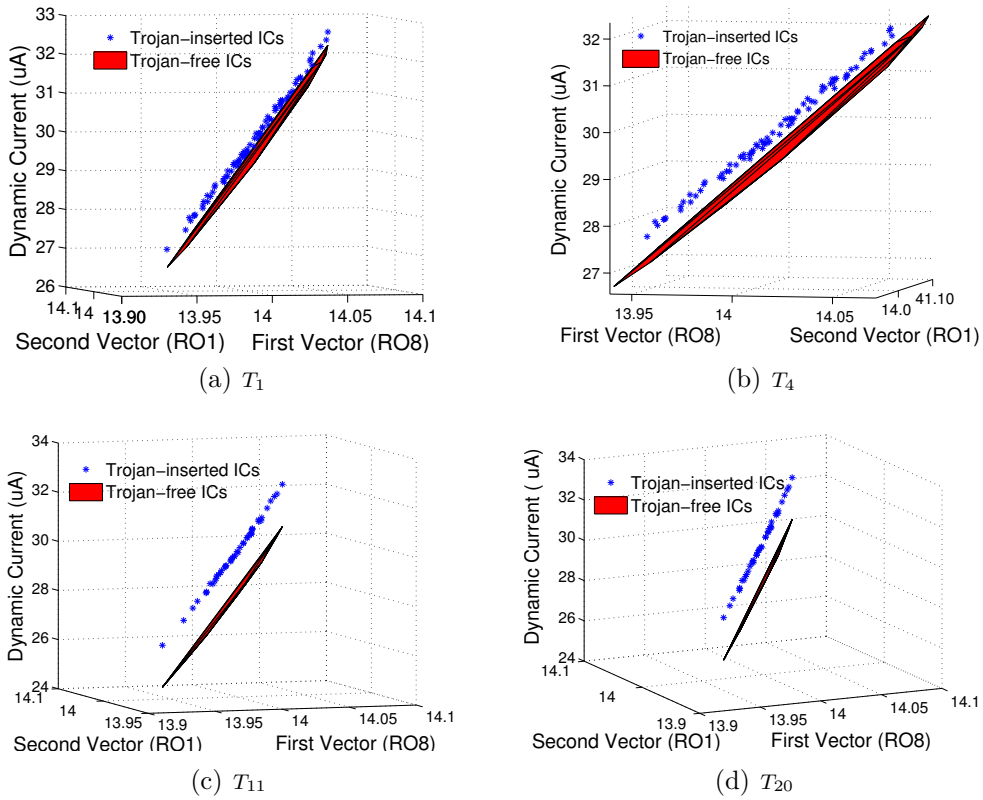


Fig. 4.8: Signatures with outlier data analysis from IC simulation.

were implemented with Monte Carlo simulation. The location of the inserted Trojans is fixed throughout this analysis. For RONs with different quantities of ring oscillators, the layout is similar to Figure 4.5. The three columns of ring oscillators were replaced by 2, 4, and 5 columns of ring oscillators respectively.

Figure 4.9 shows Trojan detection rates using advanced outlier analysis with different number of ring oscillators in RON for Trojans T_1, T_2, and T_3. With 10 ring oscillators, the detection rates for T_1, T_2, and T_3 are 40%, 48%, and 53%, respectively. With 25 ring oscillators, the detection rates increase to 95%, 100%, and 100%. These results imply that increasing the number of ring oscillators in the circuit improves detection rates. This is because a Trojan will likely be closer to a ring oscillator (or perhaps several) with more of them embedded in a design.

When the number of ring oscillators in the RON is increased, the power consumption will be unchanged while the circuit is under normal operation; the RON is only on for a short time during testing and remain off during functional operation. The area overhead would increase slowly with the number of ring oscillators. For our simulation, the area overheads are 2.5%, 3.75%, 5.0%, and 6.25% with 10, 15, 20, and 25 ring oscillators in the RON inserted in s9234. However, the increase in area overhead is small in comparison to the increase in Trojan detection rates. Thus, the RON structure may be adjusted to meet desired area overhead and detection resolution values.

Trojan Location Analysis: in order to verify the impact of a Trojan's

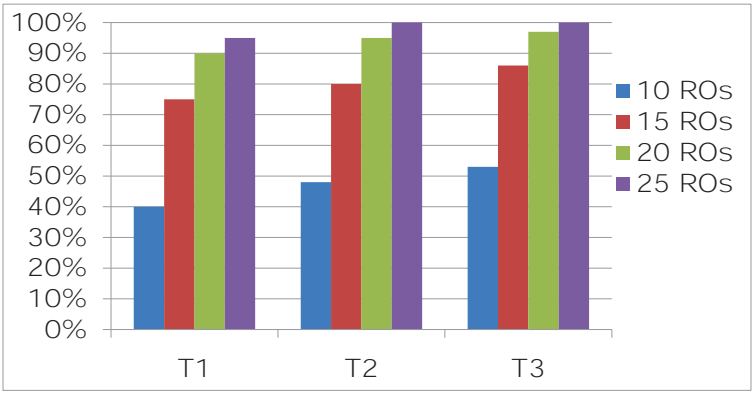


Fig. 4.9: Ring oscillator number (N_{RO}) analysis with Trojans T_1, T_2, and T_3.

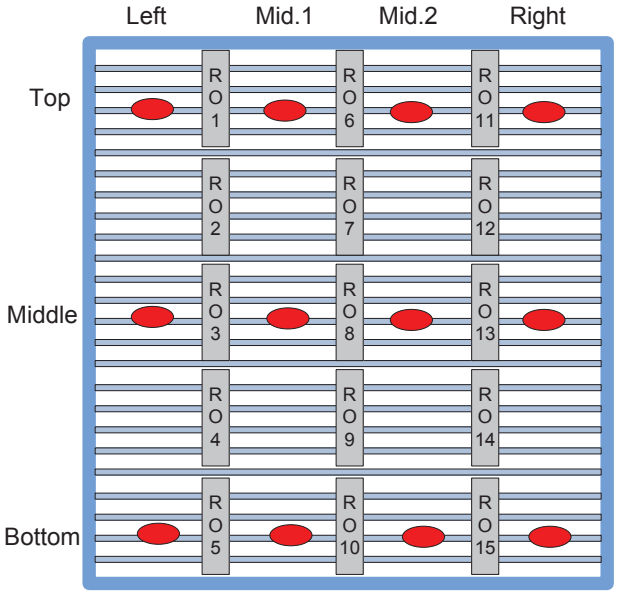


Fig. 4.10: Placing T_2 at different location in the s9234 circuit.

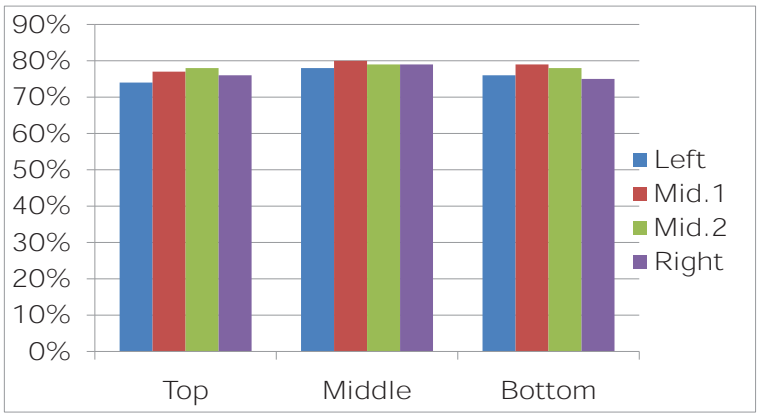


Fig. 4.11: Trojan location analysis with T_2.

location on its detection rate, Trojan T_2 was placed in twelve locations (shown in Figure 4.10). For each location, 200 Trojan-free ICs and 100 Trojan-inserted ICs were generated by a Monte Carlo simulation. A RON of 15 ROs was embedded into the benchmark. The detection rates using advanced outlier analysis are shown in Figure 4.11. From the figure, we can see that when Trojan T_2 was placed around boundary corners, fewer Trojan-inserted ICs would be detected than if it was placed centrally. This occurs because the Trojan is closer to a greater number of ring oscillators when placed towards the center. However, the Trojan detection rate varies by less than 8% for the twelve locations. This can likely be alleviated with greater design coverage; placing columns of ring oscillators adjacent to the outermost edges of the IC will limit the maximum distance between a Trojan and an RO.

Pattern Analysis: since different inputs could cause different switching activities in ICs, the pattern generated by the LFSR during testing can impact

the Trojan detection resolution in two ways: (1) Trojan switching activity (and thus the Trojan contribution to changes in dynamic power) depends on circuit inputs and thus the pattern selected and (2) the total switching activity in the circuit may be altered by the patterns. Increased switching among Trojan gates implies a greater Trojan contribution to side-channel information. Decreased total switching in the circuit under authentication implies reduced background noise and a greater chance that Trojan activity will not be obfuscated. It is crucial to note that Trojan switching activity does not refer to the event of actually activating a Trojan to launch its malicious function, but rather refers to any amount of switching in the gates which comprise the Trojan. For example, for Trojan T_3, which is composed of four gates, if only one gate transitions we will say that there was switching activity in the Trojan, regardless of whether or not the Trojan was completely activated. The LFSR was simulated to verify the impact of pattern selection on our combined ring oscillator network and dynamic current method. We use different seeds in the LFSR to generate different patterns; 1000 patterns are generated by one seed.

Figure 4.12 shows the detection rates with four different seeds ($S1$, $S2$, $S3$, and $S4$) in the LFSR. The four different seeds were randomly generated by MATLAB. Trojans T_1, T_2, and T_3 were selected to show the results. All these Trojans were fixed at locations shown in Figure 4.5. From Figure 4.12, we can see that our Trojan detection method gives different detection rates using different

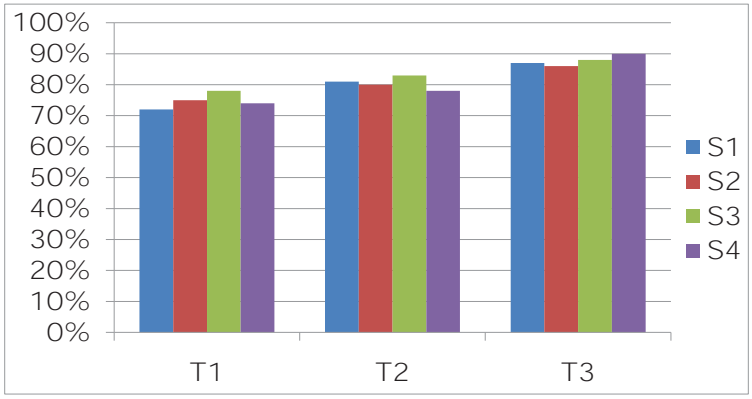


Fig. 4.12: Pattern analysis with Trojans T_1, T_2, and T_3.

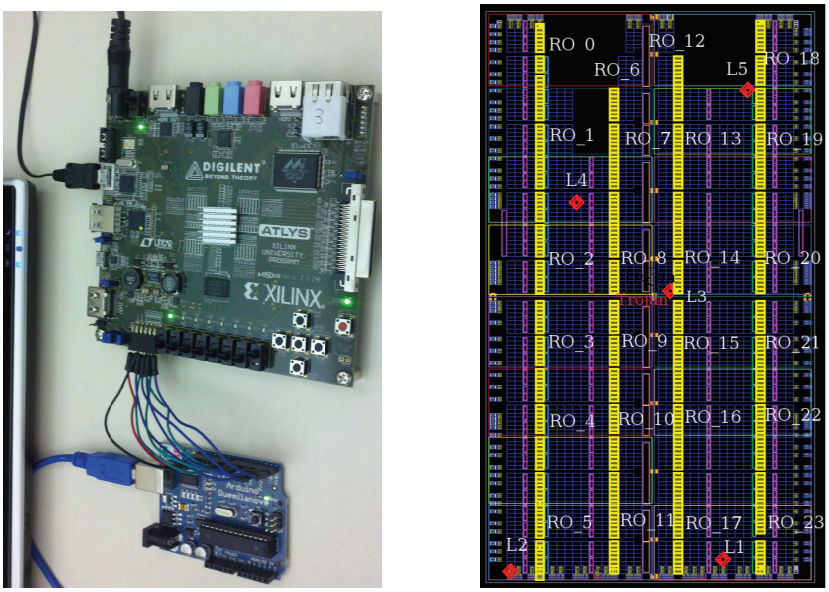
Table 4.4: Trojans Inserted in FPGAs and their detection rate when $N_{RO} = 24$.

	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}	T_{26}	T_{27}	T_{28}
Area Overhead	0.0016%	0.012%	0.025%	0.05%	0.08%	0.1%	0.15%	0.2%
Trojan Detection Rate	73%	86%	100%	100%	100%	100%	100%	100%

random patterns. In generally, the detection rate will be higher if the Trojan switching activity is greater. However, with random patterns the Trojan detection rates do not vary significantly. If we can generate special patterns, such as ones that could cause more switching at nets which activate rarely in the design, our Trojan detection method would be more effective.

4.4.3 Experimental Results from Spartan-6 FPGA

Xilinx Spartan-6 FPGA boards (shown in Figure 4.13(a)) were used for the hardware validation of our proposed method and 24 ring oscillators were inserted into



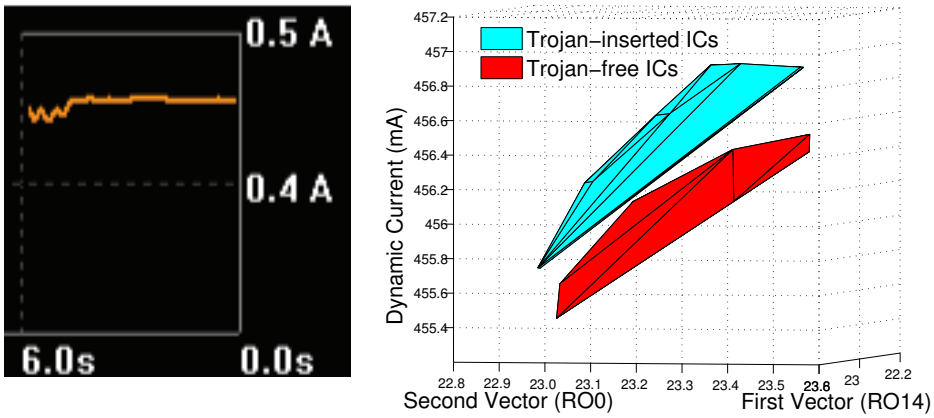
(a) (b)

Fig. 4.13: (a) Xilinx Spartan-6 FPGA board (45nm technology) and (b) AES layout after placement.

an AES benchmark circuit (shown in Figure 4.13(b)). An Atmel Atmega328P microcontroller is connected to the FPGA to facilitate in the collection of ring oscillator cycle count data from the counter. Transient current waveforms (shown in Figure 4.14(a)) are collected using Digilent Adept software [47]. 28 Trojan-inserted FPGAs and 60 Trojan-free FPGAs were used to verify the impact of process variations. Several measurements were done for each ring oscillator in each FPGA in order to eliminate measurement noise, and the average oscillation count was used to perform data analysis. The Trojans implemented in the following analysis are composed of arbitrary combinational gates of varied sizes. The malicious function to be carried out by the Trojans will not be important since this analysis is intended to demonstrate the ability of our method to detect arbitrarily added yet difficult to detect malicious gates.

Eight different Trojans T_{21}-T_{28} with different sizes were inserted into the AES benchmark. As seen in Table 4.4, some Trojans are extremely small and switch rarely during functional operation. For example, the switching probability of Trojan T_{21} is 0.0016%. These Trojans were located at location L_3 (shown in Figure 4.13(b)). The area overhead and detection rates of these Trojans are shown in Table 4.4. T_{26} was used to show the detailed results of our advanced outlier analysis in Figure 4.14(b). From the figure, we can see that the Trojan detection rate for T_{26} is 100%. With all the Trojan detection rates (shown in Table 4.4), we can also see that most of Trojans were detected with a 100% detection rate

however for very small Trojans, the detection rates were lower.



(a) T_{26} (b) T_{26}

Fig. 4.14: (a) Transient current waveform and (b)Outlier analysis results with Trojan T_{26} from FPGA implementation.

The impact of the number of ring oscillators on detection rates was analyzed on Xilinx Spartan-6 FPGAs in addition to the simulation results presented earlier. Here, the number of oscillators in the network is varied and Trojans of varied sizes are inserted into the circuit. The Trojans are placed in the same location, and the same LFSR seed is applied for each part of this experiment. RONs, composed of 8, 16, and 24 ring oscillators, were implemented in the AES benchmark circuit. Figure 4.13(b) shows the RON with 24 ring oscillators and RONs with 8 ring oscillators and 16 ring oscillators were implemented similarly. With 60 Trojan-free FPGAs and 28 Trojan-inserted FPGAs, Figure 4.15 shows detection rates with different RONs for Trojans T_{21}, T_{22}, T_{23}, and T_{24}. From the figure, we can

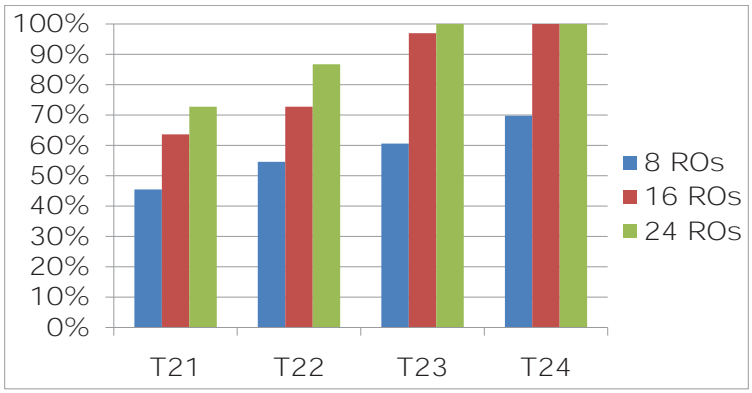


Fig. 4.15: Ring oscillator number analysis with Trojans T_{21}, T_{22}, T_{23} and T_{24} in FPGAs.

see that the number of ring oscillators in the RON plays a considerable role in the effectiveness of our method. For T_{23} and T_{24}, a detection rate of 100% is achieved by increasing the size of the network from 8 to 24 ring oscillators.

Also, a significant improvement is achieved by increasing the number of ROs from 8 to 16, but a smaller improvement is seen when the number of ROs is increased from 16 to 24. This suggests that detection resolution is not linear with the number of ring oscillators in RON.

To analyze the sensitivity of our method to location of Trojans, T_{22} was placed in different locations, from L_1 to L_5. Figure 4.16 shows results using our data analysis flow. The detection rate varies between 88.3% and 79.3% by changing the Trojan's location. When the Trojan was placed in locations L_4 and L_3, the detection rate is relatively higher since it impacted more ring oscillators. When the Trojan was located in L_2, at a corner of the FPGA, the Trojan detection

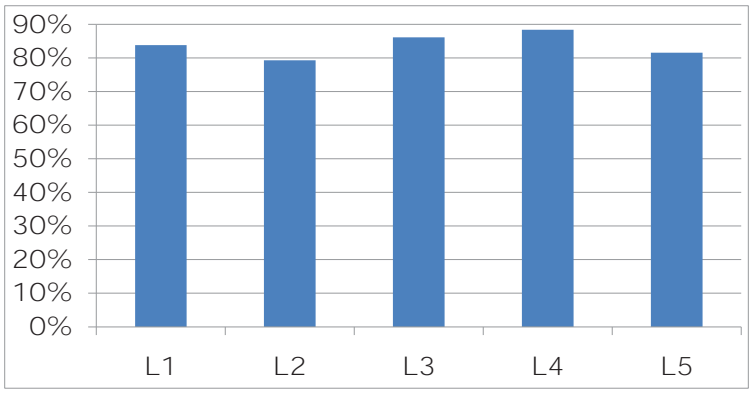


Fig. 4.16: Trojan location analysis with Trojans T_{22} in FPGAs.

rate is at its lowest.

To analyze the impact of patterns, Trojan T_{22} was located in L_3. Six randomly selected seeds were applied to the LFSR. The ring oscillator cycle counts and transient current waveforms were collected and analyzed. Figure 4.17 shows the data analysis results. From the figure, we can see that random patterns do not have a significant impact on the Trojan detection rate. We acknowledge, however, that if a designer were to intelligently select a set of patterns which control background noise and net coverage, more substantial improvements in detection resolution are possible.

4.5 Conclusions

In this chapter, we proposed an effective Trojan detection framework which combines an on-chip structure with off-chip current measurements. We have shown that our technique has the capability of detecting very small Trojans with very

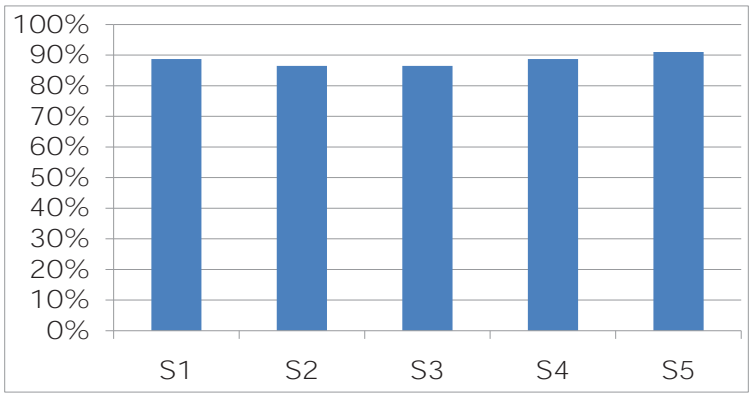


Fig. 4.17: Patterns analysis with Trojans T_{22} in FPGAs.

little contribution to circuit transient current. Statistical analysis distinguishes the effects of hardware Trojans from process variations. The experimental results on 45nm FPGAs demonstrated that this approach is very effective at identifying Trojan-inserted ICs.

Chapter 5

Experimental Analysis of Ring Oscillator Network for Hardware Trojan Detection in a 90nm ASIC

In Chapter 3 and Chapter 4, we presented our ring oscillator network that serves as a power supply monitor by detecting fluctuations in characteristic frequencies due to malicious modifications (i.e. hardware Trojans) in the circuit under authentication. In order to further verify the effectiveness of our method, we implemented the ring oscillator network on a design with controlled hardware Trojans and fabricated the design using IBM 90nm ASICs technology by MOSIS. With silicon results for 40 test chips, this chapter will analyze the impact of Trojans with varied partial activity, area, and location on the proposed ring oscillator structure and demonstrate that stealthy Trojans can be efficiently detected with our technique even while obfuscated by process variations, background noise, and environment noise.

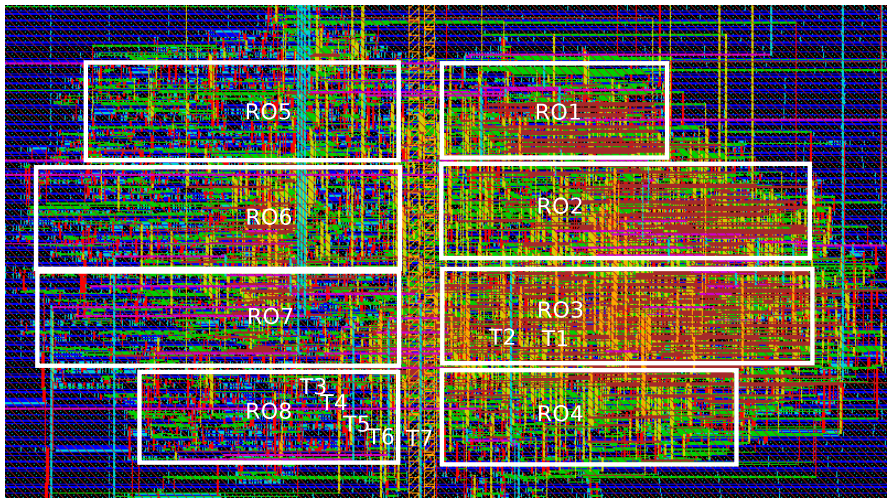


Fig. 5.1: Layout for the test chip design.

5.1 IC Design and Implementation

5.1.1 Test Chip Design

The RON architecture shown in Figure 4.2 is inserted into the ISCAS s9234 benchmark which represents the design to be protected in the test chip. Figure 5.1 shows the layout of the test chips with the RON structure composed of $N_{ro} = 8$ $n = 61$-stage ROs (RO_j where $1 \leq j \leq 8$) with one NAND gate and 60 inverters each distributed across the chip. It is important to note that the areas labeled RO_1 to RO_8 show the broad area in which that RO is confined rather than the total area occupied by that RO. Ring oscillator stages are placed in each standard cell row in an intentionally, loosely distributed fashion that improves its coverage of the power distribution network. Therefore, these areas are also occupied by background circuit and control structure components and the area overhead

of the oscillators is substantially lower than the labeled areas. The approximate locations of the seven Trojan stages (T_i where $1 \leq i \leq 7$) are labeled as well. The number of RO stages was selected so that the maximum observed frequency would not exceed the $400MHz$ operating frequency of the 90nm counters used in this design. The distance between the two adjacent RO components is limited to 10 times of the width of the flip-flops. Based on this design rule and the area of the chip, 8 ROs were used.

The feedback polynomial of the LFSR used in our test chip is

$$X^7 + X^3 + 1 \tag{5.1}$$

To conserve area, this design uses an LFSR with only 8-bits to generate patterns for the 36 input s9234 benchmark. A broadcasting technique is used to assign this 8-bit output to the 36 inputs. An 8-bit decoder and 8-bit multiplexer are used for RO selection. A 16-bit counter is used to measure the number of oscillations observed in the test duration which is controlled by a timer. In this design, the test duration of 500 clock cycles was selected based on the technology node and test area overhead.

5.1.2 Hardware Trojan Design

Each IC contains seven combinational hardware Trojan designs which may be completely deactivated. Since this design is implemented in 90nm CMOS technology, the static power dissipation and side-channel contribution are negligible

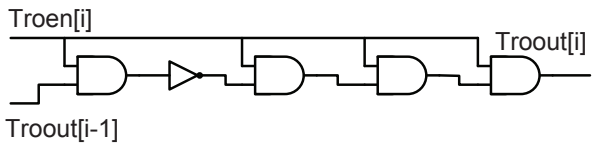


Fig. 5.2: Design of a hardware Trojan stage T_i.

when the Trojans are deactivated. By using a single-IC multiple-Trojan design we are able to not only carry out a more extensive set of Trojan impact tests, but we are also able to isolate the effect of process variations from the effect of inserted Trojans on RO characteristic frequencies. Further, since the static power is present in the Trojan-free case, it is neglected in comparisons to Trojan-inserted cases, and the detection results provide a lower-bound.

The gate-level implementation of a Trojan stage is shown in Figure 5.2 where $troout[i]$ is the output of the ith Trojan stage, $troout[i-1]$ is the output of the previous Trojan stage, and $troen[i]$ is the enable signal for the ith stage which also asserts all prior enable signals when enabled. Trojan T_i contains i stages consisting of $i \times (4AND + 1INV)$ gates where each stage $i-1$ is also enabled if stage i is enabled. The first Trojan, T_1 is driven by the $200MHz$ clock signal at the location of signal $troout[0]$. Note that the Trojan, T_i, is not derived of the trigger-payload Trojan design used in [11] [23] [24]. Here, each Trojan gate transitions once per clock cycle, therefore, the partial activity of each of these Trojans is simply $5i$ partial activations per clock cycle. The average ratio of Trojan partial activation to background circuit activity is estimated in the fourth column of Table 5.2.

Table 5.1: Estimation of area occupied by s9234 in terms of the number of transistors.

Component	Quantity	Total Transistors
D Flip-Flops	211	7174
Inverters	3570	7140
Gates	2027	8108
Total	5808	22422

The s9234 benchmark consists of 211 D flip-flops, 3570 inverters, and 2027 other gates. The number of transistors used in the s9234 benchmark is estimated in Table 5.1 by assuming each flip-flop consists of 8 NAND or NOR gates and 2 inverters. As mentioned earlier, there are a total of seven Trojans (T_1 to T_7) in this design. The area overhead of each Trojan is summarized in Table 5.2.

5.2 Experimental Setup

During data collection, the IC is mounted on and wired to a prototyping board which includes a high-density serial connector. The serial connector allows the prototyping board to interface with a Xilinx Spartan-6 FPGA on a Digilent Nexys 3 board. The FPGA is programmed to control the test sequence supplied to the IC and transmit the outputs of the IC to a computer using an on-board USB-UART module. The complete setup is shown in Figure 5.3.

The nominal supply voltage of the pins of the IC is 2.5V. This is converted

Table 5.2: Estimation of Trojan area overheads and noise.

Trojan Number	Transistors	Percent Area	Trojan to Background Circuit Switching Ratio
T1	26	0.12%	0.11%
T2	52	0.23%	0.22%
T3	78	0.35%	0.33%
T4	104	0.47%	0.45%
T5	130	0.58%	0.56%
T6	156	0.70%	0.67%
T7	182	0.81%	0.78%

internally to the nominal core voltage of 1.2V using a voltage divider. Since the s9234 benchmark circuit used in this design is small compared to a modern IC, in order to emulate the tight power design of a modern circuit, an external voltage divider is used to supply the IC with 1.875V and the core with 0.9V which is greater than the 0.80V minimum core voltage. In practice, reducing the power supply voltage will reduce the background circuit switching activity and improve Trojan detection rates. Therefore, it is desirable to reduce the supply voltage during measurement.

The FPGA includes a state machine which sequences through each ring oscillator, begins a data collection trial, selects each 4-bit window of the counter output for the current ring oscillator, and transmits each 4-bit window as a hex digit over the USB-UART connection. The process is repeated for 10 trials on

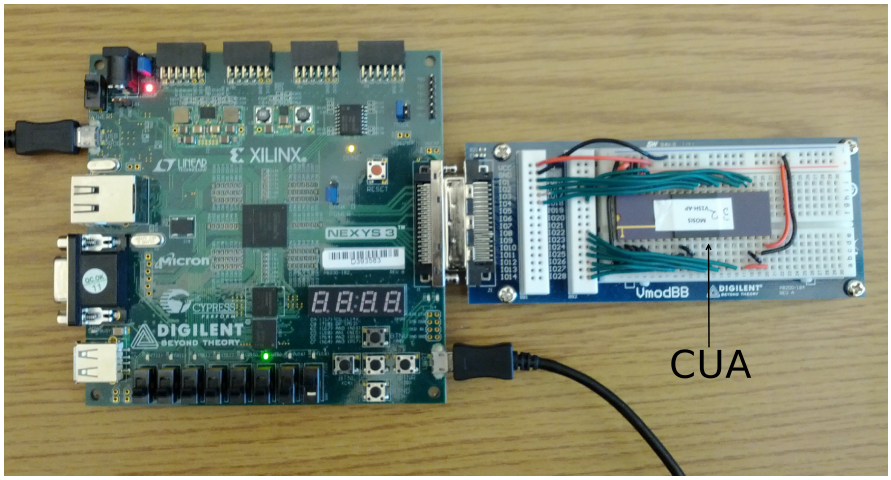


Fig. 5.3: Data collection setup including a Spartan 6 FPGA connected to a prototyping board through a serial connector.

each ring oscillator of each IC. The IC is supplied with 1.875V using a voltage divider and the board's 2.5V peripheral power supply over the serial connection along with a 200MHz clock signal. Each trial lasts 500 clock cycles.

As shown in Figure 5.1, each of the 40 ICs contains $N_T = 7$ pre-inserted hardware Trojan designs. During Trojan-free data collection each hardware Trojan circuit is disabled, as is any Trojan not being analyzed. Since the designs are implemented with CMOS circuits, the static dissipation is negligibly low. Furthermore, since all Trojan measurements are compared to the Trojan-free results (which include static dissipation) the presented detection results provide a conservative lower bound.

5.3 Experimental Results and Analysis

The frequency of a single ring oscillator on a single IC was measured 10 times. The measurement noise is then calculated with

$$\frac{Max\{f_{Trial1}, ..., f_{Trial10}\} - Min\{f_{Trial1}, ..., f_{Trial10}\}}{0.1 \sum_{m=1}^{10} f_{Trialm}} \tag{5.2}$$

for a single IC and a single ring oscillator where f_{Trialm} is the mth repeated measurement of frequency for that RO. This is repeated for all ICs and all ROs and averaged resulting in a measurement noise of 0.23%.

The impact of intra-die variation on an RO's frequencies was analyzed by comparing a single RO on an IC with other ROs on that same IC. For a single IC, intra-die variation is calculated with

$$\frac{Max\{f_{RO_1}, ..., f_{RO_8}\} - Min\{f_{RO_1}, ..., f_{RO_8}\}}{0.125 \sum_{j=1}^{8} f_{RO_j}} \tag{5.3}$$

where f_{ROj} is the frequency of the jth RO. This calculation is repeated for all ICs and averaged resulting in a mean intra-die variation impact on frequency of 8.05%.

Of the 40 fabricated ICs, 38 functioned correctly and the remaining faulty ICs are omitted. The impact of inter-die variation on the frequency of a ring oscillator was determined by selecting a single RO and comparing the frequency of this RO across each IC. For a single RO the inter-die variation is calculated with

$$\frac{Max\{f_{IC1}, ..., f_{IC38}\} - Min\{IC_1, ..., f_{IC38}\}}{(1/38) \sum_{k=1}^{38} f_{ICk}} \tag{5.4}$$

Table 5.3: Summary of validation data

Measurement Noise	0.23%
Intra-die Variation	8.05%
Inter-die Variation	16.67%
Mean RO Frequency	291MHz

where f_{ICk} is the frequency of the individual RO of interest on the kth integrated circuit. This calculation is repeated for all ROs and averaged resulting in a mean inter-die variation impact on frequency of 16.67%. The average RO frequency of all ROs on all ICs was $291MHz$. The maximum recorded frequency was $315MHz$ which was less than the $400MHz$ frequency the counter was timing closed at. These results are summarized in Table 5.3.

5.3.1 Trojan Impact Analysis

The direct impact of hardware Trojan induced power supply noise on ring oscillator frequencies is analyzed by measuring the frequency of each RO on each IC for the Trojan-free case as well as for each Trojan. The mean impact of a particular Trojan on a particular RO is then computed by comparing the frequency of that RO on a particular IC with the frequency of that RO on the same IC with the Trojan disabled. The computation is thus

$$TROI_{ROj,Ti} = (1/38) \sum_{k=1}^{k=38} \frac{|RO_{j,k,Tfree} - RO_{j,k,Ti}|}{RO_{j,k,Tfree}} \times 100\% \tag{5.5}$$

where $TROI_{ROj,Ti}$ is the mean impact of the ith Trojan on the jth RO across all ICs compared to the Trojan-free case. $RO_{j,k,Tfree}$ is the Trojan-free frequency for the jth RO on the kth IC, and similarly, $RO_{j,k,Tj}$ is the frequency of the jth RO on the kth IC with the ith Trojan activated.

It is with this calculation that the value of the single-IC multiple-Trojan design is best demonstrated. By comparing measurements made with a Trojan enabled against measurements made on the same IC with the Trojan disabled inter-die variation is eliminated from the analysis. Had separate ICs been fabricated with Trojans inserted and Trojans removed, only comparisons between different ICs would be possible and the computation would include inter-die process variation. By restricting comparisons to the same RO intra-die process variations are eliminated from the computation as well.

The results for Trojan impact are presented in Figure 5.4. It is immediately clear that Trojans of greater area and those which partially activate more frequently induce a greater change in the frequencies of nearby ROs since they consume more power. The maximum induced change for the largest Trojans in this experiment is representative of one of the core issues in the IC trust problem. The Trojan induces at most a change of 2.5% to frequencies, yet as Table 5.3 reports, intra-die variation and inter-die variation induce far greater changes suggesting these Trojans would be completely obfuscated in a test where these variations are not isolated. However, Trojan detection is still possible with this technique. The

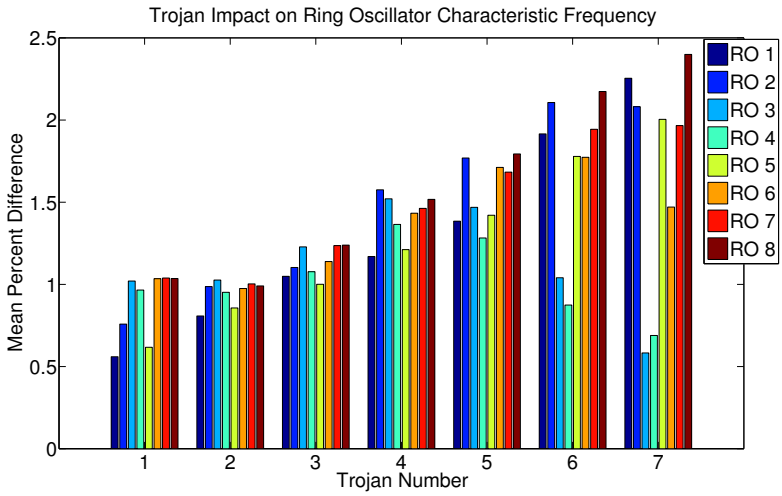


Fig. 5.4: The impact of inserted hardware Trojans on RO frequencies isolated from process variations.

manner in which Trojan impact is distributed across ROs, including the decrease in impact on RO_3 and RO_4 for larger Trojans.

5.3.2 Spatial Locality Analysis

To analyze the effect of Trojan location, the ring oscillator which experiences the greatest Trojan impact calculated with Equation 5.5 is determined for each IC with a particular Trojan. A histogram showing the frequency with which each ring oscillator was the most impacted on an IC is shown in Figure 5.5. The location of Trojan gates relative to the gates of the ROs and the vertical power line is shown in Figure 5.1

Notably, RO_8 is impacted most frequently for all Trojans since several of its gates are closest to the vertical power strap thereby causing a portion of the

overall power supply noise to affect this RO. For T_1 and T_2 a substantial portion of the Trojan impact is distributed on RO_2 and RO_3 since these Trojans are located close to these ROs and likely share power lines.

Since the majority of the gates in subsequent Trojans are closest to RO_8, more of the Trojan impact is distributed on this RO. Perhaps counter-intuitively, the distribution becomes more focused on a single RO as the Trojan expands in size. Had the Trojan expanded vertically and towards multiple ROs it is likely the distribution would become less focused. However, for these Trojans which extend primarily horizontally, the increase in area and activity further increases the Trojan impact without expanding into other regions of the power network.

For T_7 the Trojan becomes less localized on RO_8 since T_7 is particularly close to the vertical power strap. For this reason, the Trojan impact is more evenly distributed across ROs since the vertical power strap supplies power to the entire circuit. Finally, the reduced impact on RO_3 and RO_4 for T_6 and T_7 shown in Figure 5.4 is due to the loosely distributed nature of these ROs away from the vertical power line and the placement of these Trojans close to the vertical power line.

5.3.3 IC Classification and False-Positive Analysis

In previous section, it was shown that all Trojans used in this study impacted the RO frequencies substantially less than inter-die and intra-die process variations.

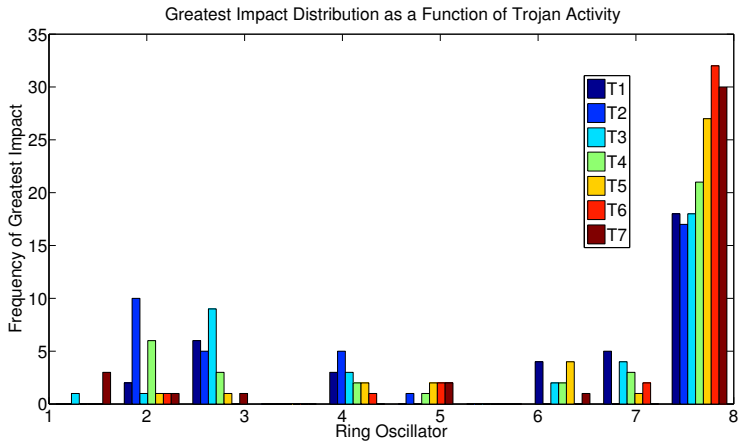


Fig. 5.5: Number of instances of each RO being most impacted by a Trojan.

However, using the principal component analysis (PCA) [43] based classification scheme presented below, it is still possible to detect these Trojans. In order to verify that this data is adequately represented in fewer than 8 principal components, the percent of the total variance in each PCA representation is computed by dividing the cumulative sum of the latent of the PCA representation by the total sum. The percent variance for each representation is shown in Table 5.4. The results imply that any representation of at least 2 components should adequately represent this data.

To succeed, a classification scheme must perform two functions: (1) it must correctly label Trojan-inserted circuits as tampered and (2) it must correctly label Trojan-free circuits as un-compromised. The steps for the presented classification scheme are:

1. Form a matrix from golden (Trojan-free) data in which each row is a verified

Table 5.4: Percent variation contained in a representation of h principal components.

Components	Percent Variation
1	89.4%
2	99.39%
3	99.59%
4	99.79%
5	99.87%
6	99.93%
7	99.97%
8	100%

Trojan-free IC and each column is a ring oscillator. Append a similar row containing the data from the chip under authentication (CUA) to the matrix.

2. Obtain a representation of this matrix using the first h principal components

3. Render an h-dimensional convex hull [44] with all data except that of the CUA.

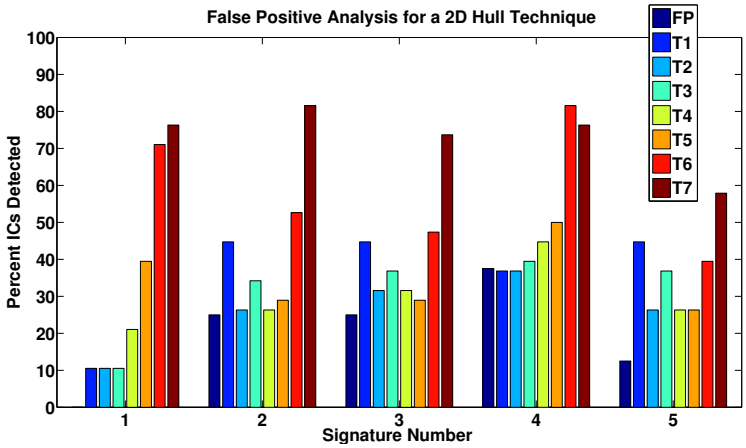
4. Determine if the CUA point falls within the hull. If it is within the boundaries of the hull it is considered Trojan-free.

To examine the performance of this classification scheme, the data are organized into five cases in which 8 of the 38 functioning ICs are randomly selected

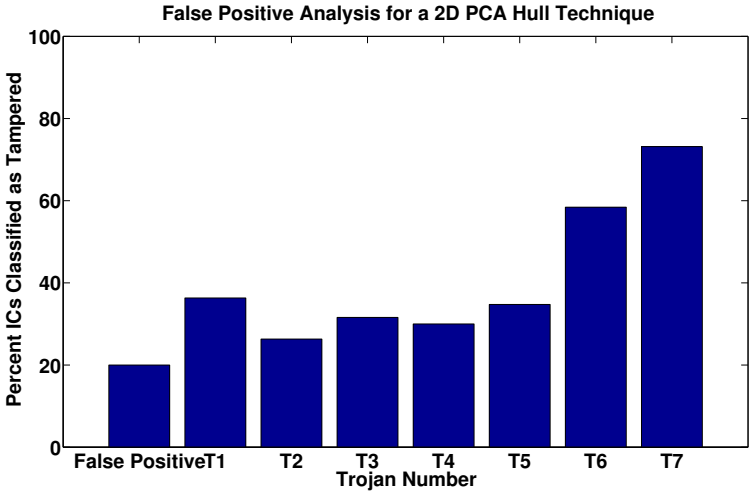
to represent Trojan-free chips to be authenticated and the remaining ICs are used to build the golden signature. All 38 ICs are used as Trojan-inserted chips under authentication.

The classification scheme was tested using both 2 and 3 dimensional hulls using the same subset cases for both hull types. The percent chips labeled as Trojan-inserted are shown for each case using both 2 and 3 dimensions are shown in Figure 5.6(a) and Figure 5.7(a) respectively. "FP" indicates the number of Trojan-free chips which were incorrectly classified. For both 2 and 3 dimensions, the behavior varies among the randomly selected cases. Thus for clarity, the average rates among all cases are shown in Figure 5.6(b) and Figure 5.7(b). For both the 2 and 3 dimensional schemes, the false positive rates are lower than the detection rates for even the smallest Trojans in the experiment. For Trojans T1-T5 the detection rates are under 50%. This is unsurprising since these Trojans consisting of fewer than 130 transistors were intentionally designed to determine and emphasize the limitations of this technique.

For the larger Trojans, the detection rates are as high as 60-70% for the 2 dimensional case and 80-90% for the 3 dimensional case. Notably, the percent ICs labeled Trojan-inserted tends to be higher for the 3 dimensional case indicating sensitivity is related to the number of dimensions used. However, the three-dimensional case also achieves a higher ratio of detection rate to false positive rate for some cases.

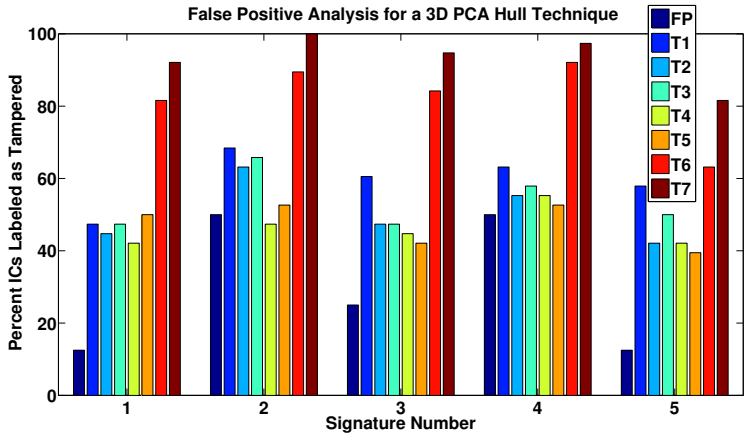


(a) All cases using 2 dimensions

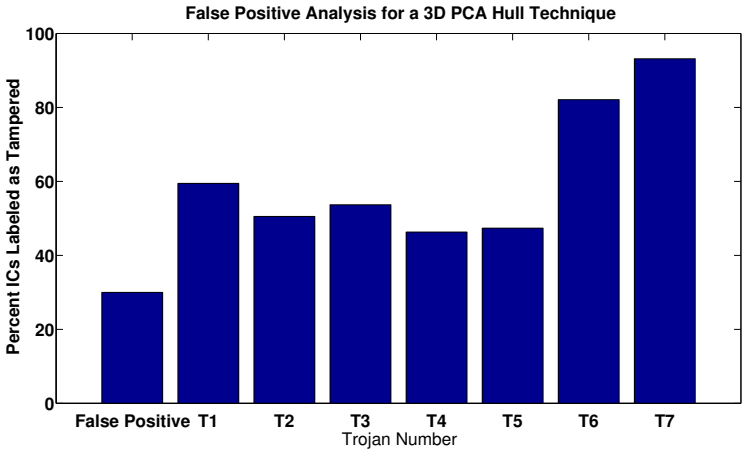


(b) Mean rates using 2 dimensions

Fig. 5.6: Classification using the presented scheme and 2 dimensions.



(a) All cases using 3 dimensions



(b) Mean rates using 3 dimensions

Fig. 5.7: Classification using the presented scheme and 3 dimensions.

These results demonstrate that the ring oscillator network scheme and the presented classification scheme can adequately separate Trojan-inserted designs from the Trojan-free designs despite the presence of obfuscating process variations. Although intra-die and inter-die variations introduce roughly 8% and 17% variations in RO frequencies respectively compared to the 1-3% change induced by the inserted Trojans, this technique successfully classifies ICs by exploiting the spatially correlated nature of process variations.

5.4 Conclusions

In this chapter, our proposed RON structure for detecting hardware Trojans was analyzed using 38 ICs containing the ISCAS s9234 benchmark circuit fabricated using the IBM 90nm process. Each IC contains 7 different hardware Trojans. By using a single-IC multiple-Trojan design we are able to not only carry out a the extensive set of Trojan impact tests, but we are also able to isolate the effect of process variations from the effect of inserted Trojans on RO characteristic frequencies. We have shown that ring oscillator frequencies increase with increasing Trojan partial activity and that ring oscillators which share power lines with nearby Trojans will be most impacted. The presented results reveal that it is possible for Trojan impact to counter-intuitively become more localized as it expands in size provided it remains within the region most closely aligned with a single ring oscillator. Lastly, this chapter has demonstrated that the proposed IC

classification method is very effective to detect Trojan-inserted ICs even in the presence of obfuscating process variations, measurement noise, and environment variation.

Chapter 6

Design of On-chip Light-Weight Sensors for Effective Detection of Recycled ICs

In Chapter 2, 3, 4, and 5, several hardware Trojans detection techniques in 3PIPs and ICs are presented, which could help improve the security and trustworthiness of circuits. The other topic we will focus on in this thesis is recycled ICs detection. As we mentioned in Chapter 1, the recycled ICs have the original appearance, functionality, and markings as the devices they are meant to mimic, but they have been used for a period of time before they are re-sold. It is vital to develop new techniques to help measure these ICs' specifications and effectively detect them if they have already been used in the field even for a short period of time.

The major difference between recycled ICs and unused ICs is that recycled ICs have already been used and experienced aging, as they were removed from their original boards and re-sold in the market. Aging effects, such as negative bias temperature instability and hot carrier injection, would have had an impact on the performance of the recycled ICs due to the change in threshold voltage.

In this chapter, we propose two techniques using light-weight sensors (RO-based and AF-based) to help with the detection of recycled ICs.

The RO-based sensor is composed of a reference ring oscillator (Reference RO) and a stressed ring oscillator (Stressed RO). The Stressed RO is designed to age at a very high rate by using high threshold voltage gates to expedite aging so that ICs used for a period of time can be identified. The Reference RO is gated off from the power supply during chip operation, so that it experiences less stress. The frequency difference between the two ROs could denote the usage time of the chip under test (CUT); the larger the difference is, the longer the CUT has been used, and with a higher probability the CUT could be a recycled IC. With close placement of the two ROs in the RO-based sensor, the impact of intra-die process variations could be minimized. Data analysis can effectively distinguish the frequency differences caused by aging from those caused by temperature and inter-die process variations, to identify recycled ICs, which is demonstrated by our simulation and silicon results. The RO-based sensor presents a negligible area overhead, imposes no constraint on circuit layout, and is resilient to removal and tampering attacks. The three working modes of the RO-based sensor proposed in the chapter ensure that the Reference RO cannot be gated on alone, thus the frequency difference between the two ring oscillators cannot be changed to mask detection.

The second half of this chapter, we propose the AF-based sensor, composed

of counters and an embedded antifuse memory block, to identify recycled ICs. The counters are used to record the usage time of ICs and the value is dynamically stored in the antifuse memory block by controlling the programming signal. Since the antifuse memory block is one time programmable, "recyclers" could not erase the context during recycling process. Therefore, our AF-based sensor is resilient to removal and tampering attacks. Two different structures of AF-based sensor are proposed to measure the usage time of ICs in this chapter: (*i*) AF-based sensor using clock (CAF-based) records the cycle count of the system clock during the chip operation. The usage time of recycled ICs can be reported by this sensor and the measurement scale and total measurement time could be adjusted according to the application of ICs. (*ii*) AF-based sensor using signal transition (SAF-based) selects a certain number of signals with low switching probability and records their switching activities to calculate usage time to detect recycled ICs with less area overhead compared to CAF-based sensor.

6.1 Background

In this section, we will briefly describe aging phenomenon in ICs and present their impact on different circuit components, which will be used in our RO-based sensor. The antifuse OTP memory used in the AF-based sensor will also be briefly introduced in this section.

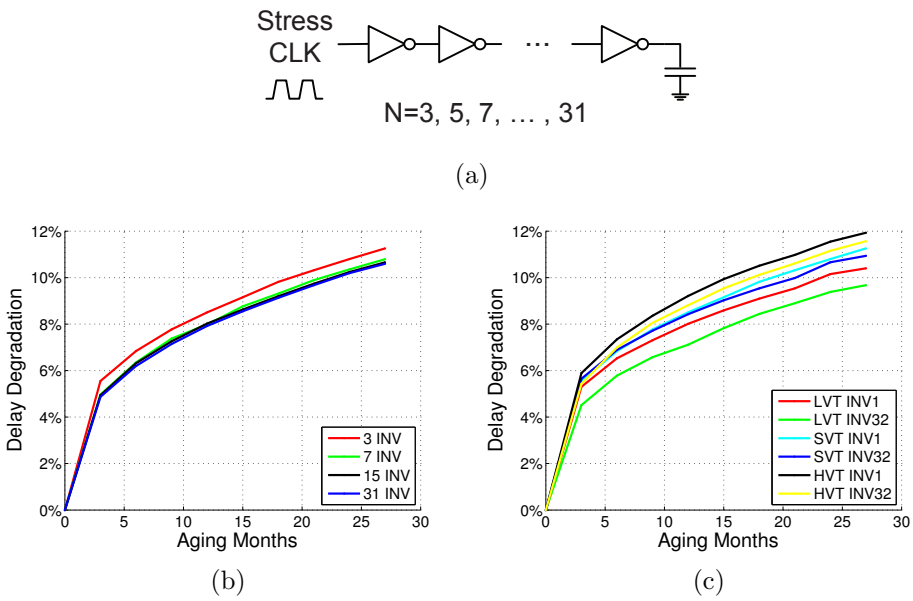


Fig. 6.1: (a) Inverter chain structure, (b) Degradation of inverter chains with different lengths (stage count), and (c) Degradation of a 3-inverter chain with different inverter types.

6.1.1 Aging Analysis

When the chip operates in functional mode, the transistors age mainly due to NBTI and HCI. The aging effects of NBTI and HCI could cause parametric shifts and circuit failures, as demonstrated by reliability models [57] [59] [60]. NBTI occurs when a negative gate-to-source voltage is applied at the PMOS transistors, which breaks Si-H bonds generating the interface traps. These interface traps can increase the absolute value of the PMOS threshold voltage (V_{th}), resulting in reduced transistor current and increased gate delay. Equation 6.1 shows the shift

of V_{th} caused by NBTI [61].

$$\Delta V_{th} = \frac{qN_{it,NBTI}(t)}{C_{ox}} \tag{6.1}$$

where C_{ox} represents the gate oxide capacitance, q is the electronic charge, and $N_{it,NBTI}(t)$ is the number of interface traps, which will increase as the transistors continue to operate in the field. HCI occurs when the electron or hole in transistors gains sufficient energy to overcome silicon dioxide barrier in order to break an interface state. The silicon substrate/gate dielectric interface and dielectric bulk traps caused by HCI can impact device parameters including threshold voltage, shown in Equation 6.2.

$$\Delta V_{th} = \frac{qN_{it,HCI}(t)}{C_{ox}} \tag{6.2}$$

where $N_{it,HCI}(t)$ is the number of interface traps caused by HCI.

Since recycled ICs have been impacted by these aging effects when used in the field, the circuit parameters of recycled ICs would be different from those of new ICs. If a *fast-aging sensor* was embedded into the circuit to help detect its usage, then recycled ICs could be identified.

In order to verify the effects of aging on a circuit's performance, several different inverter chains were simulated using Synopsys 90nm technology [62]. The delay of these inverter chains will represent the circuit's performance. The simulation was conducted using HSPICE MOSRA (Synopsys' reliability analysis tool) with combined NBTI and HCI aging effects at $25°C$. Figure 6.1(a) shows the basic structure of the inverter chains with the same capacitive load and the

same stress coming from a 500MHz clock. These chains are composed of 3, 7, 15, and 31 standard, high, and low threshold voltage (SVT, HVT, and LVT) inverters. Figure 6.1(b) presents the delay degradation of inverter chains under clock stress for up to 27 months with no interrupt. From the figure, we can see that the number of inverters does not have a significant impact on the degradation of these chains since they receive the same stress, and each inverter's speed degrades at the same rate. Aging effects are also dependent on device's threshold voltage. The 3-inverter chains were simulated using SVT, HVT, and LVT and two different size inverters (INVX1 and INVX32). Figure 6.1(c) shows that the chain with the HVT inverters experiences more degradation than the chains with SVT or LVT inverters. The INVX1 inverter chain has a larger degradation than the INVX32 inverter chain.

NAND and buffer (BUF) gate chains with HVT were also simulated at $25°C$ with a 500MHz clock stress. The basic structure of these chains is the same as the inverter chains. A NAND gate will function as an inverter when its two inputs are connected together. Figure 6.2 shows the simulation results. From the figure, we can see that the gate type does not impact the aging speed significantly. However, the inverter chain ages slightly faster than the others, while the NAND gate chain and the BUF chain age at almost the same speed. The difference in the amount of aging depends on the structure of gates. Therefore, inverters (INVX1) with HVT will be used to create the ring oscillators used to detect recycled ICs in our

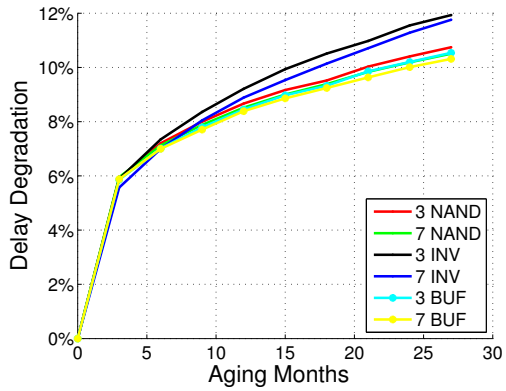
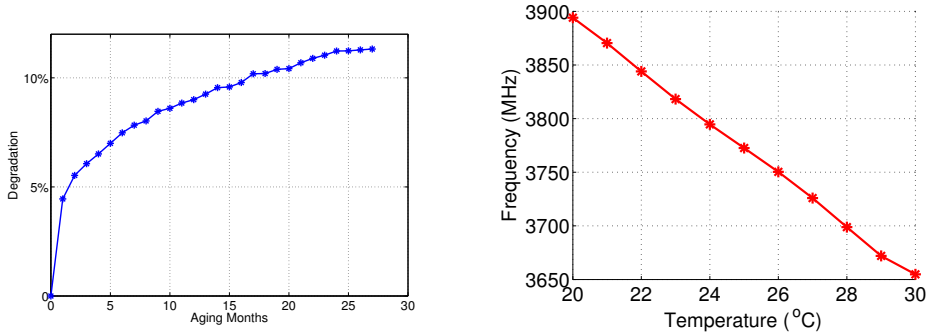


Fig. 6.2: Delay degradation of NAND, BUF, and INV chains.



(a) (b)

Fig. 6.3: (a) Frequency degradation of a 5-stage RO, and (b) Frequency of a 5-stage RO decreases with increasing temperature.

simulation.

Figure 6.3(a) shows the frequency degradation of a 5-stage ring oscillator with HVT inverters after aging for 27 months. The frequency of the RO in a recycled IC will be smaller than in a new IC. If there are no environmental or process variations, we could easily identify recycled ICs by measuring the frequency of the RO embedded in the circuit. However, variations have a significant impact on the

frequency of ROs. Figure 6.3(b) shows that the frequency of the 5-stage RO will decrease as we increase the temperature, and that the frequency variation could be very large. Note that increasing temperature can also increase the degradation of the circuit.

The 1000 Monte Carlo (MC) simulation results of the 5-stage RO are shown in Figure 6.4(a), at a temperature of $25°C$ with 3σ: 2% Tox, 5% Vth, and 5% L inter-die variations and 1% Tox, 5% Vth, and 5% L intra-die variations. We can see that the frequency of the RO can vary as much as 20% under process variations. In addition, process variations impact the aging rate of the RO, as shown in Figure 6.4(b). The frequency degradation of the 1000 chips varies around 8% (7.4%-8.6%) for one year of aging. This frequency shift caused by the aging effects in recycled ICs can help separate them from those caused by process variations in new ICs if we try to use ROs to detect recycled ICs.

With a fixed stress, the number of inverters does not have a significant impact on an inverter chains' delay degradation. However, the frequency of an RO is related to the number of inverters, $f = \frac{1}{2*n*t_d}$, where n is number of stages in the RO and t_d is the delay of an inverter. Figure 6.4(c) shows the frequency shift of a 21-stage RO with HVT inverters. The frequency degradation is shown in Figures 6.4(d). Comparing the frequency degradation of the 5-stage and 21-stage ROs, we can see that the 5-stage RO experiences slightly more degradation since its oscillation frequency is higher than the 21-stage RO. However, a 5-stage

RO may require a very fast counter which might be difficult to design for timing closure.

6.1.2 Antifuse Memory

An antifuse is an electronic device that changes state from non-conducting/high resistance to low resistance in response to electrical stress. With sufficiently high voltage/current, a large power dissipation in a small area will melt a thin insulating dielectric between polysilicon and diffusion electrodes and form a thin, permanent, and resistive silicon link. The programming performed after manufacturing is irreversible and permanent in antifuse cells, which will be used in our AF-based sensor to store the usage time of ICs.

The AF-based sensor is composed of counters with usage time of ICs when power-on stored in an embedded antifuse OTP memory block during the chip operation. Otherwise, the data may be erased or altered in power-off mode by attackers. The reasons for using an antifuse block in the AF-based sensor are [63]: (*i*) it consumes less power to program or read compared with other types of OTP structures, such as electrical fuse or CMOS floating gate, (*ii*) the area of an antifuse is much smaller than an efuse, and (*iii*) it does not require additional mask or manufacturing handing steps during fabrication.

However, most antifuse memories are programmed in a programming environment with relatively high voltage/current. Therefore, integrated charge pumps

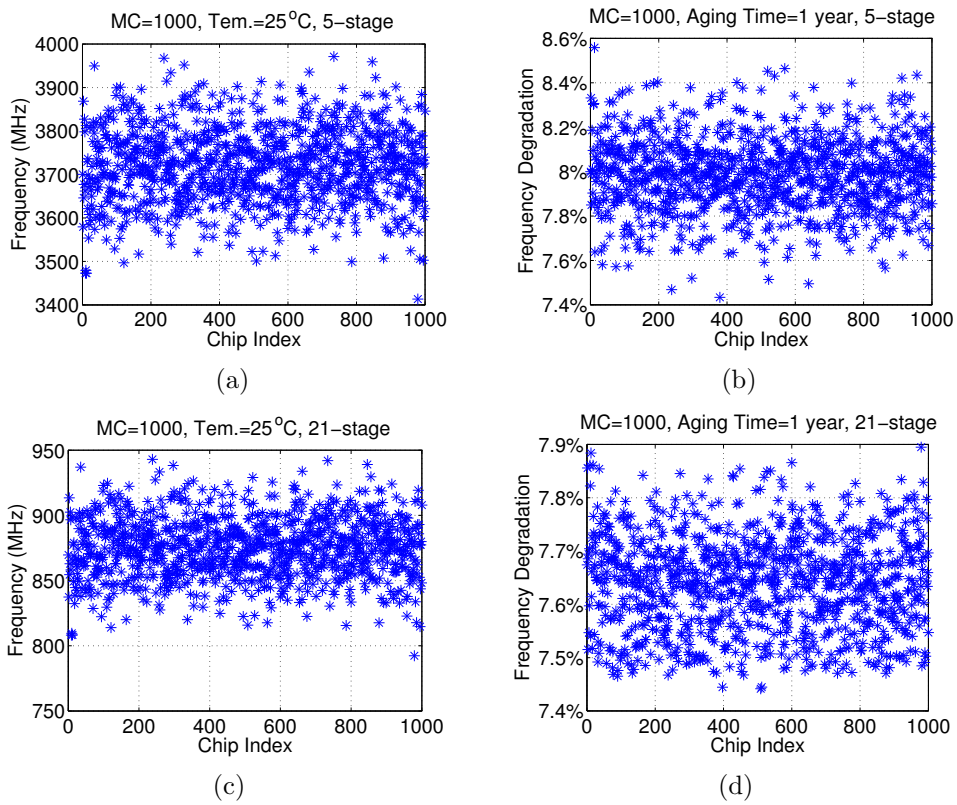


Fig. 6.4: (a) Frequency of a 5-stage RO varying with process variations, (b) Frequency degradation of a 5-stage RO aging for one year varying with process variations, (c) Frequency of a 21-stage RO varying with process variations, and (d) Frequency degradation of a 21-stage RO varying with process variations.

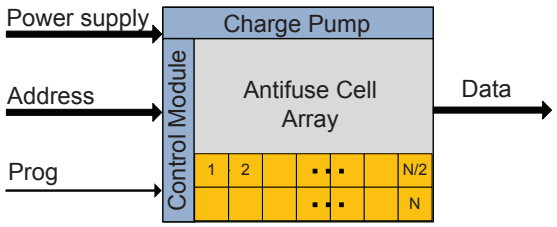


Fig. 6.5: Typical interface of antifuse memory.

or voltage multipliers are used to provide sufficiently high voltage/current [64] [65] in embedded antifuse OTP memories. With those charge pumps or voltage multipliers, no additional power supply is required during programming. The typical interface of the embedded antifuse memory is shown in Figure 6.5 [64] [65], including *Power supply, Address, Prog*, and *Data* signals. We will use existing antifuse blocks with the interface shown in Figure 6.5 instead of designing a new embedded antifuse structure in our AF-based sensor since embedded antifuse memory is only a small part of the sensor.

6.2 Recycled-IC Detection Sensors

Two different sensors to identify recycled ICs are proposed in this chapter. RO-based sensor is based on the aging differences between two ring oscillators to record the usage time of ICs. RO-based sensor does not require any memory element to store the usage time since it is hidden in the degraded RO frequency because of aging. AF-based sensors count the system clock or the switching activity of signals in the design and store the usage time in an antifuse OTP block. The two

sensors will be discussed in detail in the following.

6.2.1 RO-based Sensor

Our main objectives in designing the RO-based sensor are: (i) the sensor must age at a very high rate to help detect ICs used for a short period of time, (ii) the sensor must experience no aging during manufacturing test, (iii) the impact of process variations and temperature on RO-based sensor must be minimal, (iv) the sensor must be resilient to attacks, and finally (v) the measurement process must be done using low-cost equipment and be very fast and easy.

As mentioned earlier, aging effects could slow down the frequencies of ROs embedded into ICs. With an embedded RO, these recycled ICs could be identified based on its frequency, which will be lower than that of a new IC. However, there are many parameters impacting the frequency of an RO, such as temperature and process variations. Our RO-based sensor uses a Reference RO and a Stressed RO to separate the aging effects from process/environmental variations.

Figure 6.6 shows the structure of our RO-based sensor, which is composed of a control module, a Reference RO, a Stressed RO, a MUX, a timer, and a counter. The counter measures the cycle count of the two ROs during a pre-specified time period, which is controlled by the timer. System clock is used in the timer to minimize the measurement period variations due to circuit aging. The MUX selects which RO is going to be measured, and is controlled by the $ROSEL$

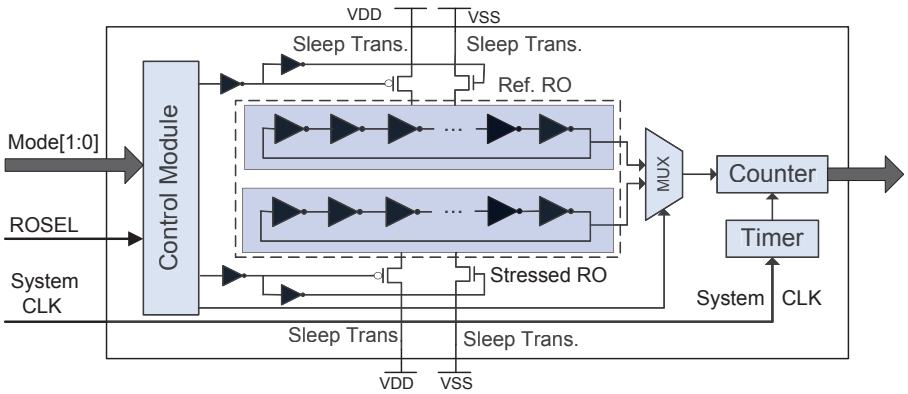


Fig. 6.6: The structure of the RO-based sensor.

signal. The Reference and Stressed ROs are identical; both are composed of HVT components. The inverters in Figure 6.6 could be replaced by any other types of gates (NAND, NOR, etc) only if they can construct a RO. It will not change the effectiveness of the RO-based sensor significantly. We use smaller-stage ROs in our RO-based sensor considering the counter's measurement speed limits given a technology. For example, in our 90nm technology, a 16-bit counter can operate under frequency of up to $1GHz$; an inverter-based RO of at least 21 stages is then required.

Sleep transistors are used to connect the ROs to the power supply in the RO-based sensor; PMOS sleep transistors control the connection between VDD and the inverters and NMOS sleep transistors control the connection between VSS and the inverters. Both the Reference RO and the Stressed RO work in three modes that are controlled by the $Mode$ signal: (i) when the IC is in manufacturing test mode, the Reference RO and Stressed RO will be disconnected from the power

supply and experience no aging. This mode only lasts a short time, depending on the test procedures of the IC. (*ii*) when the IC is in normal functional mode, the Reference RO will be disconnected from VDD and VSS but the Stressed RO will be gated on and will age. The frequency of the Stressed RO will drop while the Reference RO will not change a lot. ICs will spend most of their time in this mode. (*iii*) when the IC is in authentication mode (i.e., when an IC is taken from market and its authenticity is to be verified), both the Reference RO and Stressed RO will be gated on by connecting to the power supply. The timer and counter will be enabled to measure ROs' cycle count and $ROSEL$ signal will select which RO to measure. The rest of the functionality of the IC would be turned off by $Mode$ signals and the authentication process takes a very short period of time. The three modes of operation ensure that (*i*) the frequency difference between the Reference RO and Stressed RO will be larger over time since the Reference RO cannot be gated on alone, and (*ii*) it is extremely difficult for adversaries to force the RO-based sensor to operate in authentication mode when it is supposed to be in its normal functional mode, which would eliminate the aging difference. The only method to do that would be to modify the original RO-based sensor module, which is impossible during a simple recycling process.

As shown in Figure 6.6, the inverters of the Reference RO and the Stressed RO are placed physically next to each other, designed as a single small module. The process and environmental variations between them should be very small.

Therefore, for a new IC, the frequency difference between the Reference RO and the Stressed RO would be within a certain small range. In a recycled IC, the Stressed RO will have suffered aging from its own oscillation since the chip has been working in normal functional mode for a long time. However, the Reference RO will not have experienced as much aging since it was gated off. The frequency difference between the Reference RO and the Stressed RO will grow larger as the chip operates longer, which is demonstrated by our simulation and silicon results. If the frequency difference is outside of the new ICs' frequency difference range considering process variations, we can conclude with high confidence that the CUT was recycled from used boards. The area overhead of our RO-based sensor is negligible when compared to the millions of gates in modern ICs. Power consumption is also limited to that consumed by the Stressed RO in the RO-based sensor.

6.2.2 AF-based Sensor

In the RO-based sensor, the inverters of the Reference RO and the Stressed RO are placed physically next to each other to minimize the impact of intra-die process variations. However, it may still be difficult to completely exclude the impact of inter-die process variations on the sensor. In addition, RO-based sensor provides only an approximation of the usage time in a form of aging in the stressed RO. Therefore, the sensitivity (the minimum usage time of recycled ICs detected by

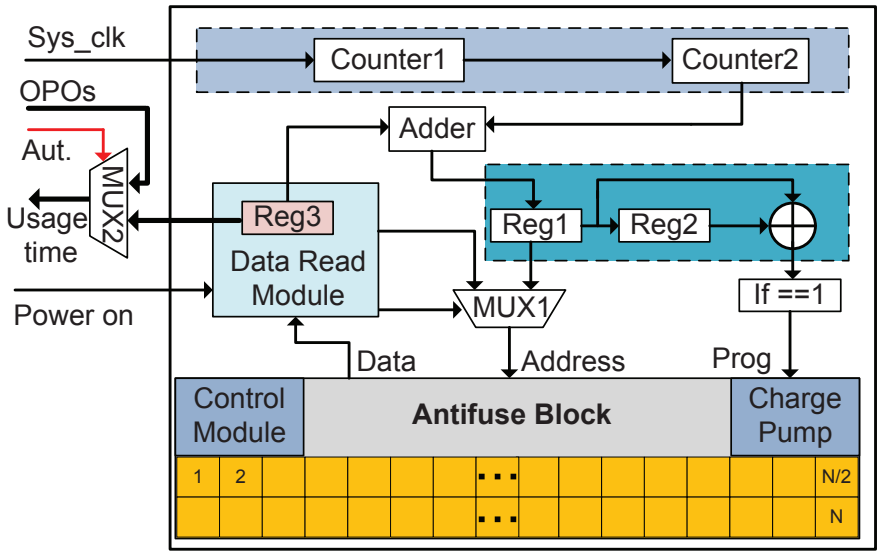


Fig. 6.7: The structure of the CAF-based sensor.

sensors) of the RO-based sensor is limited. For example, it may not identify recycled ICs used shorter than one month based on our simulation. In order to eliminate the issue of process variations, provide a more accurate usage time, and identify recycled ICs that are only used for a very short period of time (such as 1 day), we propose two AF-based sensors: CAF-based sensor and SAF-based sensor.

CAF-based sensor

Figure 6.7 shows the structure of the CAF-based sensor, which is composed of two counters, a data read module, an adder, and an antifuse OTP memory block. *counter1* is used to divide the high frequency system clock to a lower frequency signal, as shown in Figure 6.7. *counter2* is used to measure the cycle count of the lower frequency signal. The size of the two counters can be adjusted accordingly

depending on the measurement scale (T_s: defined as the time unit reported by the sensor) and the total measurement time (T_{total}). For example, if T_s is 1 hour and T_{total} is 1 year based on the specification of an IC, a 38-bit *counter1* will meet the requirement to count the usage time from 20ns (assume system clock=50 Mhz) to 1 hour and a 14-bit *counter2* will count the usage from 1 hour to 8760 hours (1 year).

Since the data stored in registers (counters) could be lost or reset when power supply is off, non-erasable memory is required in this sensor. An embedded antifuse OTP block is used instead of a field programmable read-only memory (FPROM) to store the usage time information because FPROM could be tampered or altered by attackers. In the antifuse block, *Prog* is assigned to be $1'b1$ if the value in *counter2* increases by "1". By connecting the output of *counter2* to *Address* in the antifuse block directly, the related antifuse cell will be programmed as "1". Therefore, the largest address of the cell whose content is "1" will be the usage time of CUT based on the measurement scale setup by *counter1*.

However, program and read operations share the same *Address* signals in antifuse block. Therefore, a MUX (*MUX1* in Figure 6.7), controlled by *data read module*, is used to select the address (antifuse cell) to be read or programmed. Every time power supply is on, the antifuse block will work in read mode for a short period of time. During this time, the read address generated by *data read module* will go through *MUX1* and all the antifuse cells will be traversed based on the

Algorithm for Data Read

```
initial address = (N/2);
for (i = log(N/2), i > 0, i - -) {
  if ([address] == 1)
      address = address + 1;
      if ([address] == 0)
          address = address - 1, $stop;
      else
          address = address - 1;
      address = address + 2^(i-1);
  else
      address = address - 2^(i+1);
}
```

Fig. 6.8: Algorithm for "data read" in CAF-based and SAF-based sensors.

traversing binary tree principle. Figure 6.8 shows the algorithm for data read in a N-bit antifuse block. From Figure 6.8, we can see that there are $log(N/2)$ loops in the algorithm. The address is increased or decreased by $2^{i-1}(i = 0, ...log(N/2))$ for the ith loop based on the value in the address. If the value stored in the address is "1" ($[address] == 1$) and the value stored in the next address is "0", the address will represent the usage time before power-on based on T_s. The read operation will last less than $log(N/2) + 1$ system clock cycles, depending on the value stored in the antifuse block; this time will be recorded by *counter1*, as well.

Once we get the previous usage time, it will be stored in register *Reg3* and sent to the *adder*. The reason for using an adder here is that counters start from "0" every time the power is turned on and the previous usage time must be considered when we calculate the total usage time. In addition, *Reg1* is used to sample the data in *adder*, *Reg2* delays the data in *Reg1* with one system clock, and *XOR* gates are used to compare the data in *Reg1* and *Reg2*. If they are different (denoting the usage time increased), the antifuse OTP block will work in program mode and the data in *Reg1* will go through *MUX1* to the *Address* in the antifuse block. Therefore, combined with the value in *counter2* (the usage time after power-on), the new total usage time will be stored in the antifuse OTP block by programming a new antifuse cell with a larger address. From the above discussion, we can see that the antifuse OPT block is programmed internally. By designing our sensor in this way, we can reduce the probability of altering or

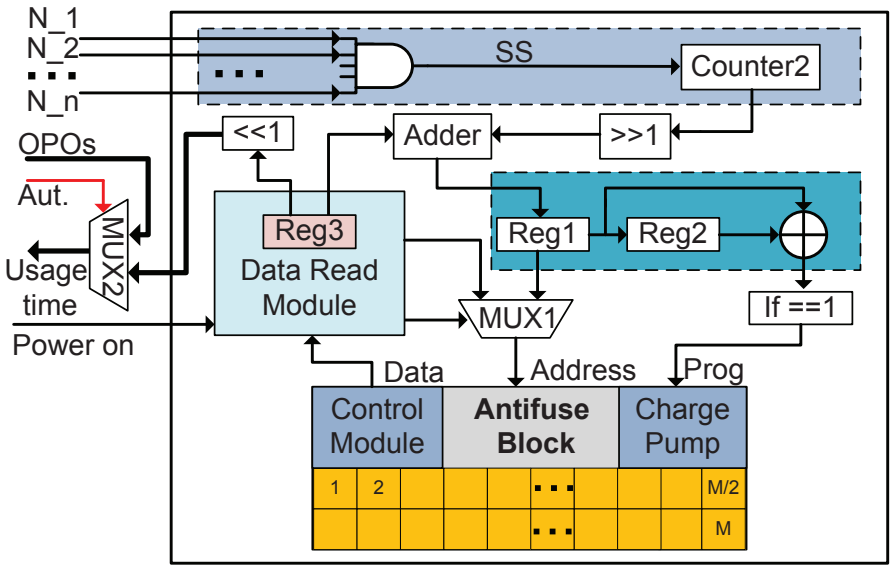


Fig. 6.9: The structure of the SAF-based sensor.

tampering attacks on the AF-based sensor.

In order to eliminate the need for additional pins for authentication purposes on the chip, our CAF-based sensor uses a MUX (*MUX2*) and an authentication (*Aut.*) pin to send the usage time to the output pins of ICs. This way, no extra output pins will be added to the original design. When the IC works in normal functional mode, original primary outputs (*OPOs*) will go through *MUX2*. If the IC is in authentication mode by enabling the authentication signal, the *data read module* will set the antifuse IP in read mode and the usage time will go through *MUX2*. In addition, when the IC works in manufacturing test mode, the functionality of our CAF-based sensor will be disabled and structural fault test patterns will be applied to the sensor.

SAF-based Sensor

With two counters, the area overhead of CAF-based sensor could still be considered large for smaller designs. In order to reduce the area overhead, we proposed SAF-based sensor based on signals' switching activity (SW) as shown in Figure 6.9. Comparing Figure 6.9 with Figure 6.7, we can see that the structure of SAF-based sensor is similar to that of CAF-based sensor. The difference is that CAF-based sensor counts the cycle of system clock to record the usage time of ICs while SAF-based sensor counts the switching activity (positive edge) of a certain number of nets in the design. With simulations, a certain number of nets are selected to be the input of an AND gate. The rule of nets selection is that the switching activity of the output of the AND gate must meet the requirement of the measurement scale. For example, if T_s is 1 hour, one of the choices could be four nets with $SW(N\_1) = 30/60mins$, $SW(N\_2) = 24/60mins$, $SW(N\_3) = 25/60mins$, and $SW(N\_4) = 24/60mins$, respectively. However, with different functional inputs, the signals' SW could be significantly different. Therefore, only the signals with consistent SW under different inputs are selected when we design a SAF-based sensor. From the analysis, we can see that the net selection could be adjusted based on different designs and measurement scales. Then the positive pulse of the output of the AND gate (SS signal in Figure 6.9) will be counted by *counter2* in the sensor.

In order to further reduce the area overhead, a 1-bit right shifter is used

to divide the value in *counter2* by 2 and then the largest address of antifuse cells with "1" will represent *[SW/2]*. A 1-bit left shifter is used to calculate the switching activity by *[SW/2] * 2*. The recorded SW will represent usage time of ICs. Therefore, the number of antifuse cells in SAF-based sensor will be reduced compared with CAF-based sensor. However, the accuracy of SAF-based sensor is lower than CAF-based sensor because (i) it is based on the switching activity of a certain number of nets in the netlist while CAF-based sensor counts the cycle count of the system clock, and (ii) the SAF-based sensor loses part of the usage time information due to the shifters.

Compared with RO-based sensor, the area overhead of the two AF-based sensors is larger because of the counters and the antifuse OTP block. However, it is still negligible when compared to the millions of gates in modern ICs. The major advantage of AF-based sensor over RO-based sensor is that the usage time stored in the AF-based sensors to identify recycled ICs will not be impacted by technologies (i.e., older technology designs do not age as much as the new ones do), packages, assemblies, or process variations. Even if the design was fabricated at different time in different foundries, the AF-based sensor could still indicate how long chip under test has been used. In addition, AF-based sensors could identify recycled ICs used for a very short period of time, such as 1 day, due to the small measurement scale.

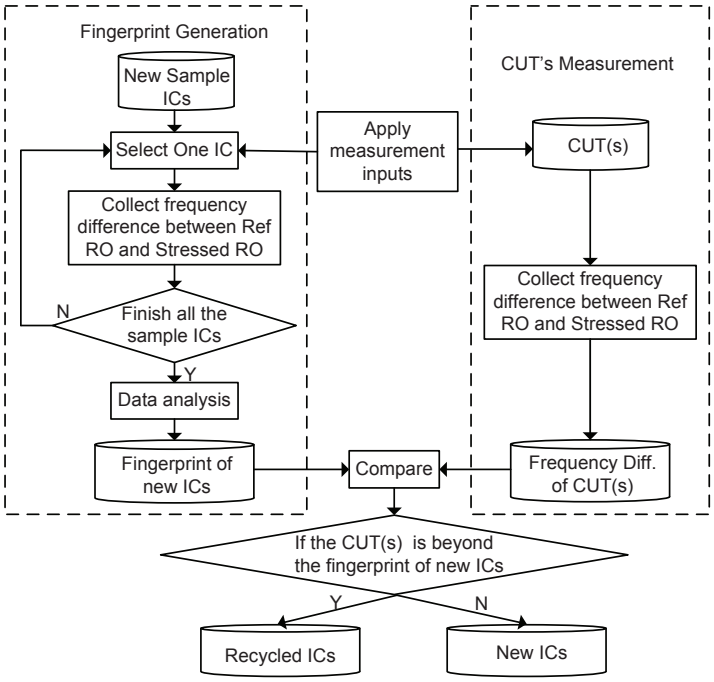


Fig. 6.10: Measurement flow using RO-based sensor for identifying recycled ICs.

6.3 Results and Analysis

In this section, we will present the experimental results of the RO-based sensor and AF-based sensor including simulation results and silicon results from test chips. Attack analysis on the two sensors will also be discussed.

6.3.1 RO-based Sensor

Figure 6.10 shows the proposed measurement flow using RO-based sensor for identifying recycled ICs in our experiments. This is done only for the purpose of validation of our proposed sensor. The way RO-based sensor is designed, it eliminates the need for a golden IC, especially when chip is used for a long period

of time in the field. First, a certain number of random, new ICs are used as sample chips to generate a fingerprint. The samples can come from the same or from different wafers and lots. The larger this sample is, the more process variation space will be covered, reducing the probability that new ICs with large process variations will be identified as recycled ICs. 1000 sample chips are tested in our simulation. In authentication mode, the Reference RO and Stressed RO's frequency is measured. We acknowledge that temperature variation should not impact the identification results significantly, since the Reference RO and Stressed RO will experience the same environmental temperature.

Once the sample chips have been measured, the frequency difference between the Reference RO and Stressed RO would be calculated, with $F_{diff} = F_{ref} - F_{str}$, where F_{ref} is frequency of the Reference RO and F_{str} is frequency of the Stressed RO. With 1000 sample chips, the range of F_{diff} will be determined using distribution analysis, creating a fingerprint for new ICs. If F_{diff} of the CUT is out of the range of the new ICs' fingerprint, there is a high probability that the CUT is a recycled IC. Otherwise, the CUT is assumed to be a new IC. The longer the CUT has been used, the more aging effects it will have experienced, making it easier to identify. The entire measurement procedure for each CUT should take only a very short amount of time (less than few seconds).

Simulation Results

In order to verify the effectiveness of the RO-based sensor, we implemented and simulated it using 90nm technology [62]. HSPICE MOSRA from Synopsys is used to simulate and measure the impact of aging on the RO-based sensor. The nominal supply voltage is 1.2V. During simulation, in the stress phase, the Reference RO was gated off and the Stressed RO was gated on, experiencing NBTI and HCI aging. The stress for the Stressed RO comes from its own oscillation. In the authentication phase, the Reference RO and Stressed RO were both gated on and measured one by one, selected by the $ROSEL$ signal. The measurement time was set up in the timer as $100\mu s$ in our simulation. Since the clock of the counter in the RO-based sensor is from the RO, the cycle count of each RO is given by the counter. The frequency of RO is equal to the cycle count divided by measurement time. The following simulation analysis is based on inverter ring oscillators.

Stage Analysis: RO-based sensors with 21-stage and 51-stage ROs were simulated at $25°C$ with 2% Tox, 5% Vth, and 5% L inter-die and 1% Tox, 5% Vth, and 5% L intra-die process variations (PV0 in Table 6.1). 1000 chips were generated using Monte Carlo simulation by HSPICE and the total aging time was set at 24 months with a one month step.

Figure 6.11(a) shows the frequency difference F_{diff} range between the 21-stage Reference RO and Stressed RO, where, in the legend, AT denotes aging time, M represents month, and Y represents years. From the figure, we can see

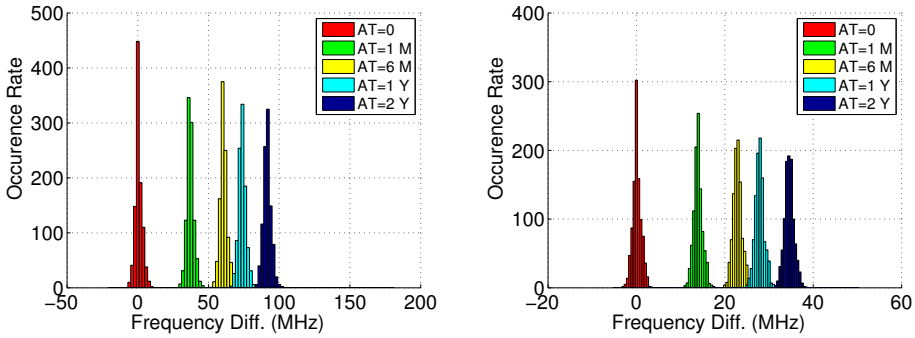


Fig. 6.11: Frequency difference distribution of RO-based sensor with PV0 using (a) 21-stage ROs, and (b) 51-stage ROs.

that the frequency difference in new ICs ($AT = 0$) could be larger or smaller than 0, which is dependent on the process variations between the two ROs. In addition, the process variations of the CUTs are different from that of the 1000 sample new ICs, but the frequency differences still follow an identical distribution. The range of frequency differences in the new sample ICs is used as the fingerprint. After being used for one month, the Stressed RO suffered from aging effects and its frequency became lower. The lowest frequency difference between the Reference RO and the Stressed RO is larger than the largest frequency difference present in the new IC set. Therefore, the recycled IC detection rate for ICs aged for one month or longer is 100%. At 6 months, 1 year, and 2 years, the frequency difference between the Reference RO and the Stressed RO becomes larger and larger. The variation of the frequency difference becomes larger as well. This is because the aging rate is different from chip to chip due to process variations;

some ICs aged faster and some others aged slower.

RO-based sensors with 51-stage ROs were also implemented using the same temperature and the same process variations. Figure 6.11(b) shows the simulation results. Comparing Figure 6.11(a) and Figure 6.11(b), we observe that the frequency difference between aged and new ICs is smaller when we use the larger-stage ROs. However, the frequency difference variation becomes smaller as well, which means that the RO-based sensor could still detect fully recycled ICs that had been used for one month with a 100% detection rate. If the RO-based sensor uses large-stage ROs, it may impact the absolute value of the frequency difference between the Reference RO and the Stressed RO, but the detection rate will not be impacted significantly. For different technologies, the stage count of the ROs could be adjusted based on the speed of the counter. In the following, we use RO-based sensors with 21-stage ROs according to the 90nm technology for further analysis.

Process Variations and Temperature Analysis: The effectiveness of the RO-based sensor is partly dependent on the variations between the Reference RO and the Stressed RO. With lower rates of variation, the RO-based sensor could identify recycled ICs that aged for a shorter period of time. However, the variations between the Reference RO and the Stressed RO are determined by intra-die process variations. The smaller the intra-die variations, the more effective the RO-based sensor will be. Table 6.1 shows the different process variation rates

Table 6.1: Process variations.

	Inter-die			Intra-die		
	Vth	L	Tox	Vth	L	Tox
PV0	5%	5%	2%	5%	5%	1%
PV1	8%	8%	3%	7%	7%	2%
PV2	20%	20%	6%	10%	10%	4%

used in our simulation to analyze their impact on detection. Moving from PV0 to PV2, inter-die and intra-die variations both become larger. RO-based sensors with 21-stage ROs were simulated at $25°C$ using these process variation rates.

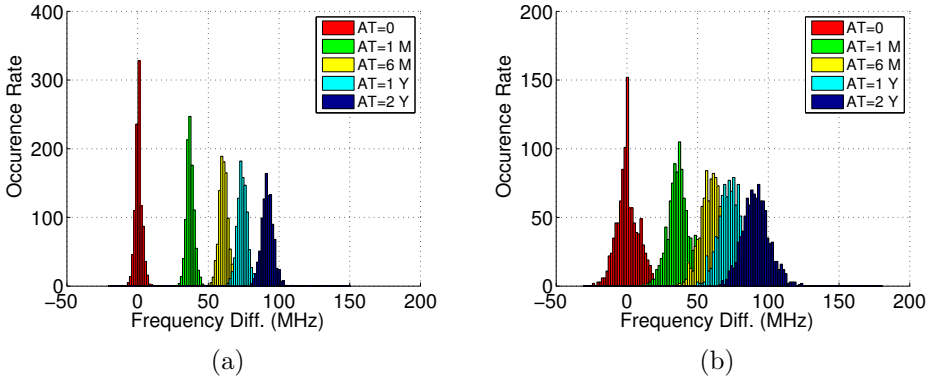


Fig. 6.12: Frequency difference distribution of RO-based sensor with 21-stage ROs with (a) PV1 and (b) PV2.

By designing the sensor as a small module (hard macro), the Reference RO and the Stressed RO were placed physically close and the variations between them were minimal. The simulation results of 1000 chips with PV1 and PV2

are shown in Figure 6.12(a) and Figure 6.12(b), respectively. Comparing Figure 6.11(a), Figure 6.12(a), and Figure 6.12(b), we can see that the variation of the frequency differences between the Reference RO and the Stressed RO in new ICs becomes larger with larger process variations. For the 1000 ICs with PV2, the detection rate of recycled ICs aged for one month is 95.2%. However, for recycled ICs that aged for six months, the detection rate is 100%. The RO-based sensor identifies shorter-aged recycled ICs with smaller intra-die process variations as in PV0, PV1, and PV2.

The 1000 circuits generated using Monte Carlo were also simulated with both process and temperature variations. Figure 6.13(a) shows the frequency difference occurrence rate between the 21-stage Reference and Stressed ROs with process variations PV1 (shown in Table 6.1) and temperature variations of $\pm 10°C$ around room temperature. Figure 6.13(b) shows the simulation results with process variations PV2 and temperature variations of $\pm 20°C$ around room temperature. The results in Figure 6.13(a) and Figure 6.12(a) are from chips with the same process variations but different temperature variations. We can see that the frequency difference variations in Figure 6.13(a) are slightly larger than those in Figure 6.12(a) due to temperature variations. The same conclusion can be made by comparing Figure 6.13(b) and Figure 6.12(b). For the 1000 chips with PV2 and $\pm 20°C$ temperature variations, the detection rate of recycled ICs aged for one months is 92.3% but it is still 100% for recycled ICs aged for six months,

Table 6.2: Structure of RO-based sensors in the test chip.

	ROs in RO-based sensors			
	Reference RO	Stressed RO	RO Structure	Vth
RO-based1	R_RO1	S_RO1	1 NAND + 200 BUFs	SVT
RO-based2	R_RO2	S_RO2	1 NAND + 200 BUFs	HVT
RO-based3	R_RO3	S_RO3	201 NANDs	HVT

demonstrating that our RO-based sensor is effective even with large process and temperature variations. Note that we do not expect such a large variation in temperature and process in practice when authenticating a CUT. The temperature difference and process variations between the two ROs in RO-based sensor will be negligible since they are placed physically near each other.

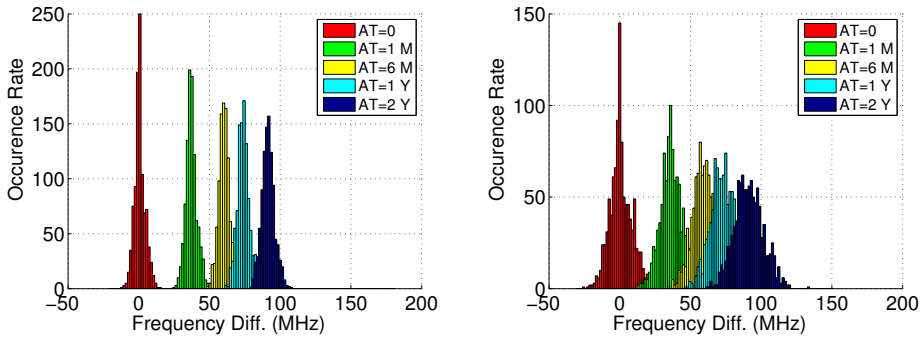


Fig. 6.13: Frequency difference distribution of RO-based sensor with (a) PV1 and $\pm 10°C$ and (b) PV2 and $\pm 20°C$.

Silicon Results

Our RO-based sensor is also verified through analysis of test chips fabricated using a 90nm technology. The test chip was originally designed to verify the effects of aging on the frequency of ROs. In this work, we use it to demonstrate the effectiveness of our RO-based sensor. In total, there are 96 delay chains in the chip which can work in ring oscillator mode by controlling different input signals [40]. Six of these ring oscillators were selected to construct three RO-based sensors as shown in Table 6.2.

- RO-based1 contains two identical ROs (R_RO1 and S_RO1) with one SVT NAND gate and 200 SVT BUFs;
- RO-based2 is composed of two identical ROs (R_RO2 and S_RO2) with one HVT NAND gate and 200 HVT BUFs
- RO-based3 includes ROs (R_RO3 and S_RO3) with 201 HVT NAND gates.

where R_RO1, R_RO2, and R_RO3 are Reference ROs while S_RO1, S_RO2, and S_RO3 are Stressed ROs, respectively.

Comparing ROs included in the test chip with those used for HSPICE simulation, there are two main differences: (1) the stage of ROs in the test chip is 201 while the stage of ROs used in Monte Carlo simulation is much smaller (e.g. 21). The much larger number of stages in test chip was used to make the measurement and observation possible with low-end oscilloscopes. (2) the gates in ROs in the

test chip are complex gates (BUFs, NANDs, etc.) while inverter-based ROs were used in simulation. That is because we aim at analyzing the impact of aging on different types of gates in the test chip. However, the number of stages and gate type of ROs do not present a significant impact on the effectiveness of the RO-based sensor.

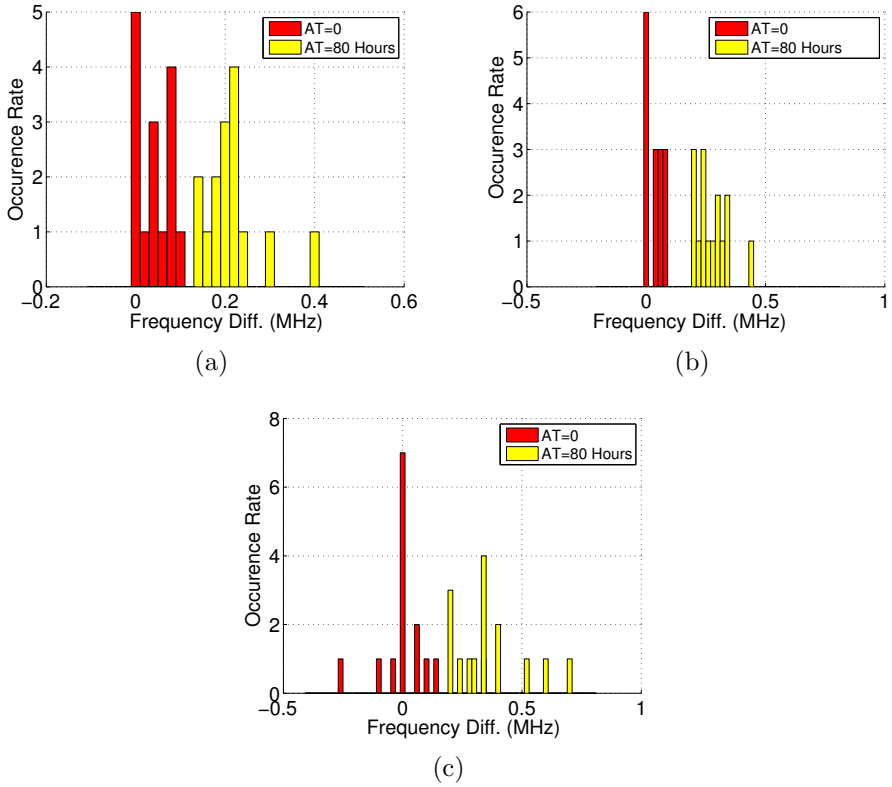


Fig. 6.14: Frequency difference distribution in (a) RO-based1, (b) RO-based2, and (c) RO-based3.

Currently, we only have 15 test chips in our lab and all of them are used in this experiment to present the impact of process variations and aging. To replicate the RO-based sensor's stressed mode, S_RO1, S_RO2, and S_RO3 were enabled

and experienced accelerated aging for 80 hours at $135°C$ with an elevated supply voltage (1.8V instead of 1.2V). The reason we used accelerated aging is that it takes a long time (usually weeks/months) to observe aging effects under normal conditions. The remaining three ROs were gated off and experienced no aging. In authentication mode, all of the ROs were enabled and the temperature was brought back to room temperature (around $25°C$). With the 15 new test chips, the average frequency of ROs is about $7.5Mhz$. Figure 6.14 shows the experimental results of the three RO-based sensors over the test chips. The red bars in the figure show the frequency difference between Reference RO and Stressed RO in each RO-based sensor at time zero (new/unused ICs). Similarly, the yellow bars are the frequency difference between the two ROs after 80 hours of aging.

Since a much larger number of stages are used in these sensors compared to those used in our simulations, the mean frequency of the ROs in the test chip and the frequency difference values are quite different from that in simulations. However, even with 201 gates in these ROs, the detection rates of recycled ICs that aged 80 hours using RO-based1, RO-based2, and RO-based3 are all still 100%, which demonstrates that the RO stage count in RO-based sensor does not have a significant impact on the sensor's effectiveness in detecting recycled ICs. According to our detailed results, the average frequency degradation of the stressed ROs in RO-based1, RO-based2 and RO-based3 (shown in Figure 6.14) is 3.2%, 4.0%, and 3.8%, respectively, Comparing Figure 6.14(a) and Figure 6.14(b),

we can see that the frequency difference gap between new chips and aged chips in RO-based2 is larger than that in RO-based1. This is due to the fact that RO-based sensors with HVT gates (RO-based2) will be more effective than those with SVT gates (RO-based1), which is also demonstrated in Figure 6.1(c) through simulation results. Comparing detection rates in Figure 6.14(b) using RO-based2 (composed of HVT buffers) and Figure 6.14(c) using RO-based3 (composed of HVT NAND gates), we can see that the gates used in the RO can slightly change the effectiveness of RO-based sensor but not significantly.

Note that the ROs in the RO-based sensors in the test chip were not placed as close as they were supposed to. For instance, the results at time zero show that for RO-based1 and RO-based2, the R_ROs are faster than S_ROs in most cases while this is not the case for RO-based3. This could be because of the spatial variations that exist between the ROs not placed near each other, which made some ROs faster than others. *For a RO-based sensor to be the most effective, it is recommended to place both ROs in a single localized module to reduce the variation between them.* Limited by the amount and structure of the test chips, we cannot perform the same analysis with silicon data as we did with the Monte Carlo simulations, however, the silicon results from these test chips demonstrate the effectiveness of the RO-based sensor.

6.3.2 AF-based Sensors

From the above analysis, we can see that detection of a recycled chip depends on the amount of degradation caused by aging, workload, process and environmental variations. However, if the chip is used for a very short period of time or if the chip is designed and fabricated using an older technology node, it will not experience much degradation, thus negatively impacting the effectiveness of detection. For AF-based sensor, since the usage time of the ICs is calculated by counters and stored in the antifuse block, process and temperature variations cannot impact the data in antifuse cells. Therefore, the only step required to know how long the IC has been used is to read the antifuse block by enabling authentication signal. Note that a non-zero usage time from an AF-based sensor in a CUT does not suggest that it is a recycled IC due to the burn-in process. The CUT can be identified as a recycled one only if the usage time is longer than the time for burn-in process. Therefore, recycled ICs used for a very short period of time can still be detected by the AF-based sensors.

Area Overhead Analysis: In order to verify the effectiveness of AF-based sensors, we analyzed the area overhead on the implementation of a design (named as CSAFTEST) with about 500K gates and 12KB in-system programmable memory. Table 6.3 shows the area overhead caused by RO-based, CAF-based, and SAF-based sensors with different measurement scales and total measurement time. From the table, we can see that the area overhead caused by AF-based sen-

Table 6.3: Area overhead caused by RO-based, CAF-based, and SAF-based sensors on CSAFTEST.

Measurement		Area Overhead			
Scale (T_s)	Total Time (T_{total})	RO-based	CAF-based	SAF-based	Reduction
1 minute	1 month	-	7.37%	3.72%	49.5 %
1 hour	1 year	-	1.57%	0.82%	47.8%
1 day	1 year	-	0.18%	0.12%	33.3%
1 day	4 years	-	0.37%	0.21%	43.2%
-	-	0.025%	-	-	-

sors change with T_s and T_{total} since the structure of AF-based sensors change with measurement resolutions. For CAF-based sensor, the size of *counter1* depends on T_s while the size of *counter2* and the size of the antifuse memory block both depend on T_{total}/T_s. For SAF-based sensor, the area overhead is much smaller than that of CAF-based sensor due to the shifters. The reduction, calculated by {*Overhead(CAF-based)-Overhead(SAF-based)*}/*Overhead(CAF-based)*, is shown in the sixth column in Table 6.3. For example, with T_s=1 hour and T_{total}=1 year (8760 hours), CAF-based sensor was designed with 20-bit *counter1*, 14-bit *counter2*, and 8760-bit antifuse memory block. The area overhead of this CAF-based sensor is 1.57% while the area overhead caused by SAF-based sensor is 0.82% and the reduction is 47.8%. However, if T_s=1 minute & T_{total}=1 month and T_s=1 day & T_{total}=1 year, the area overhead of CAF-based sensor are 7.37%

and 0.18%, respectively.

From the above analysis, we can see that the area overhead caused by AF-based sensors depends on the application and specification of ICs. For example, if an IC is used in a system that requires a small T_s and a large T_{total}, the area overhead would be large. Otherwise, the overhead would be small (less than 1%). On the other hand, the time recorded by our AF-based sensors is power-on time and the intervals between power-on are not calculated. Therefore the usage time stored in the sensor (T_{total}) is usually shorter than the time with power-off intervals. With a smaller T_{total}, the size of the antifuse memory block in our AF-based sensors will be smaller and accordingly the area overhead will be smaller.

Furthermore, comparing RO-based sensor with AF-based sensors, we can see that (i) the area of RO-based sensor is much smaller than that caused by AF-based sensors and also stays constant because the number of gates used in RO-based sensor does not vary with designs. Here, the RO-based sensor was about 0.025% area overhead, which is negligible. (ii) the accuracy of RO-based sensor is lower than that of AF-based sensors since it only provides an approximation of the usage time in a form of aging in the stressed RO.

Usage Time Analysis: Since the AF-based sensor only records usage time larger than T_s, if the power-on time of an IC is smaller than T_s, part of the usage time will be lost during the measurement. In order to verify the usage time, CAF-based and SAF-based sensors are analyzed with different T_s. Take the worst case for

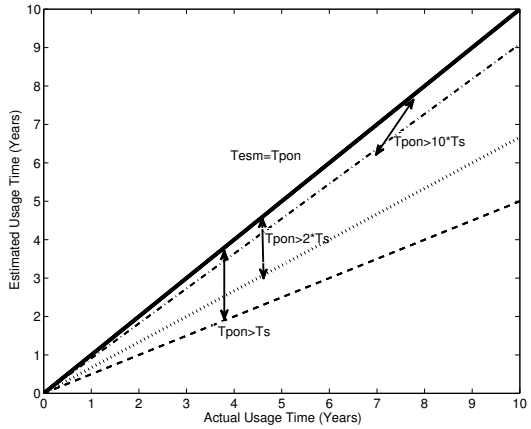
example, if every time the IC is turned on, the power-on time (T_{pon}) is shorter than T_s, then the AF-based sensors will not record any usage time. The value stored in the antifuse memory will always be equal to the time for burn-in process. Our AF-based sensors will be ineffective in this case, which should be avoided when we design an AF-based sensor.

With appropriate T_s, $N = [T_{pon}/T_s]$ will be recorded in *counter2* every time power is on and combined with previous usage time to be stored in the antifuse memory block in CAF-based sensor. Figure 6.15(a) shows the estimated usage time under different usage situations using CAF-based sensor. The X axis represents the worst case when $T_{pon} < T_s$. In this case, the estimated usage time recorded by the sensor is always zero. The solid line represents the ideal case when the estimated usage time (T_{esm}) is equal to the actual usage time. The range between the dashed line and solid line represents the estimated usage time when $T_{pon} > T_s$. Similarly, the range between the dash-dot line and solid line represents the estimated time when $T_{pon} > 10 * T_s$. From the figure, we can see that the longer the chip is used on each power-on, the more accurate estimated usage time will be recorded by CAF-based sensor.

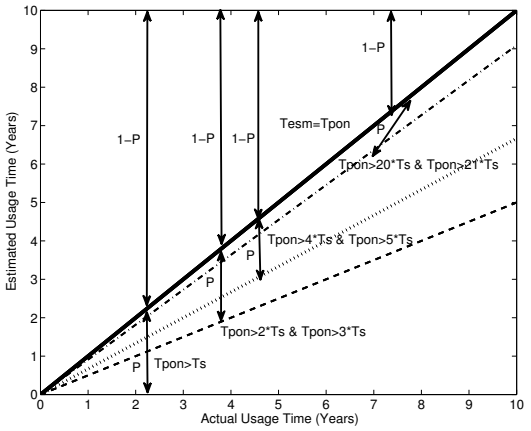
For SAF-based sensor, the estimated usage time under different usage situations is shown in Figure 6.15(b). Comparing Figure 6.15(b) with Figure 6.15(a), we can see that the accuracy of SAF-based sensor is slightly lower than that of CAF-based sensor. For example, when $T_{pon} > T_s$, the usage time recorded by

CAF-based sensor would be $T_{est} = [T_{pon}/T_s] * T_s$ while the usage time recorded by SAF-based sensor would be $T_{est} = [T_{pon}/2T_s] * 2T_s$. In addition, since SAF-based sensor is based on the switching probability of several nets in the netlist, the estimated usage time shown in Figure 6.15(b) is based on a probability. Assuming that the output of the AND gate in SAF-based sensor (SS signal in Figure 6.9) switches once during T_s with probability p, then SS will switch more than once with probability $1 - p$. Note that the case that SS does not switch during T_s will not be considered since this situation should be avoided when we design a SAF-based sensor. With this assumption, when $T_{pon} > 2 * T_s$, the estimated usage time will be in the range between the dashed line and solid line with probability p, shown in Figure 6.15(b).

Note that even with time lost during the measurement by using AF-based sensors, we can still identify a recycled IC since the usage time recorded by the antifuse memory block in used ICs will be longer than the time for burn-in process. After the burn-in process and before being sent to market, the AF-based sensor in all CUTs report almost identical usage time. However, when ICs are used in the field, the usage times recorded by the sensor in CUTs would be larger and different from each other based on the usage conditions before recycling.



(a)



(b)

Fig. 6.15: Usage time analysis using (a) CAF-based sensor and (b) SAF-based sensor.

6.3.3 Attack Analysis

Considering the capability of professional recyclers, we will discuss about a couple of attacks circumventing RO-based and AF-based sensors. The first attack to RO-based sensor could be removal and tampering attacks. However, it is inherently difficult for the recycler to remove the sensor, due to the expected measurement results from the two ROs. The second attack could be that the recycler tries to intentionally age the Reference RO to mask the difference between the ROs in the RO-based sensor. Similarly, it is impossible to do that since Reference RO cannot be gated on alone. However, one can argue that attackers with unlimited resources may be able to remove the chip package, modify the original design, and tamper with the RO-based sensor. For such ICs where additional security is required, alterations could be made to the RO-based sensor to prevent these kinds of attacks. The RO-based sensor could be obfuscated inside the IC by multiplexing functional gates. This modification would make it more difficult for an attacker to analyze the IC, and make it more difficult to tamper with the sensor or modify it in any way.

For AF-based sensors, attackers would try to mask the usage time of ICs by disabling the sensor. However, the AF-based sensor will automatically run whenever power is on and the usage time will be stored in the antifuse memory directly. Therefore, it is impossible for attackers to disable the sensor without removing the package and breaking the chip. The second attack could be erasing

and alteration of antifuse cells; this is not possible because the memory used in our sensors is an antifuse OTP block. The most important advantage of antifuse OTP technique is its ability to resist all existing reverse engineering methods because the oxide breakdown in antifuse cells occurs in a random location within a bounded enclosure and is extremely small [63]. Therefore, the state of a bit cell stays well hidden in the silicon atoms, which makes it extremely difficult for attackers to tamper with the memory. The third attack could be modification of counters or signals connection in the sensor. However, with limited resources and without access to the original design, attackers cannot modify the nets connection. Decapping, professional cleaning, and remarking would not help attackers either.

6.4 Conclusions

In this chapter, we proposed two techniques using light-weight on-chip sensors to detect recycled ICs. The frequency difference between the Reference RO and the Stressed RO in the RO-based sensor makes identification of recycled ICs easily possible. The usage time stored in the antifuse memory using AF-based sensors could indicate how long an IC has been used and then identify a recycled IC. Experimental results and analysis demonstrated the effectiveness of these sensors.

Chapter 7

Path-Delay Fingerprinting for Identification of Recycled ICs

In Chapter 6, we proposed several light-weight on-chip sensors to detect recycled ICs based on the degradation of ring oscillators and usage time reported by counters. There are very effective for recycled ICs detection. However, they only work best for designs already with those sensors but cannot address detection of existing and legacy ICs that have no such sensors embedded in them. In order to address this issue, we proposed a new technique based on path-delay fingerprinting.

For new ICs, the delay distribution of paths will be within a certain range. The fingerprint of the new ICs can be generated during manufacturing test of these ICs and stored in a secure memory for future use when identifying recycled ICs. Due to aging effects, such as NBTI and HCI, the path delays in recycled ICs will be larger than those in new ICs. For a chip under authentication (CUA), the larger the path delays are, the higher the probability there is that the CUA has been used and is a recycled IC. In this chapter, we propose a fingerprinting

and authentication flow for accurately identifying recycled ICs. Statistical data analysis is used to distinguish the path delay changes caused by process and temperature variations from those caused by aging. Since the path delay information is measured during the manufacturing test process, no extra hardware circuitry is required for this technique. In addition, there is no change required in current industrial design and test flows. Finally, this technique presents no area overhead, no power consumption, and is resilient to attacks.

7.1 Path-Delay Degradation Analysis

When a chip is used in the field, aging effects could cause some of its parameters to shift over time. NBTI increases the absolute value of the PMOS threshold voltage and results in decreasing transistor current and increasing gate delay [57] [59] [60]. HCI creates traps at the silicon substrate/gate dielectric interface, as well as dielectric bulk traps, and therefore degrades device characteristics including voltage threshold [57] [59] [60]. Since recycled ICs have been impacted by all of these aging effects, the path delay of recycled ICs will be different from those of new ICs.

In order to demonstrate the impact of aging on path delay in ICs, in a simple manner, different gate chains were simulated using a 45nm technology [66] as shown in Figure 7.1(a). The simulation was conducted by HSPICE MOSRA [67] with the built-in aging model [67] and combined NBTI and HCI aging effects

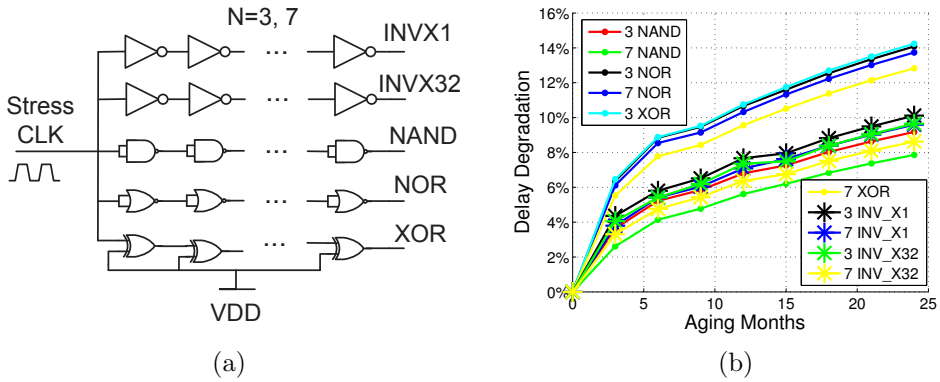
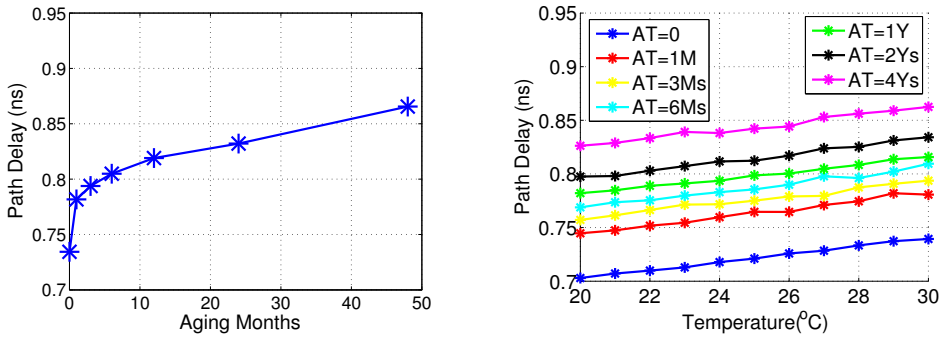


Fig. 7.1: (a) An illustrative circuit with NAND, NOR, XOR, and INV chains and (b) Delay degradation of the chains.

at a temperature of 25°C. Standard threshold voltage (SVT) INVX1, INVX32, NAND, NOR, and XOR gate chains of different lengths were simulated for up to 2 years of usage. Figure 7.1(a) shows that all chains are experiencing stress from a 500MHz clock. Any other stress (e.g., DC stress which is a constant "0" or "1", or AC stress with different duty ratios) and usage time could be used in this simulation. Figure 7.1(b) presents the delay degradation caused by 2 years (24 months) of aging. From the figure, we can see that different gate chains age at slightly different rates, which depends on the structure of the gates. The XOR gate chain has the fastest aging rate amongst these chains. Comparing the delay degradation rates of the INVX1 and INVX32 chains, we can conclude that larger gates will age at a lower rate than smaller gates with the same stress. In addition, the workload (input value and the switching frequency of each gate) also has a significant impact on the aging rate. ICs may be recovered from different used



(a) (b)

Fig. 7.2: (a) Delay degradation of path P_i and (b) P_i delay increases with increased temperature.

boards from different users who may have applied different workloads to the IC at different times. It is practically impossible to know the exact input vectors applied by the user. We will discuss this and the impact workload has on a chip's path delay degradation in detail later.

Figure 7.2(a) shows the delay of a randomly selected critical path P_i (this path includes 22 gates) from the ISCAS'89 benchmark s38417 with stress from a random workload (functional patterns) applied to the primary inputs. The path was aged for 4 years with NBTI and HCI effects at room temperature 25°C. From the figure, we can see that the degradation of path P_i used for 1 year is around 10% while if the circuit is used for 4 years, the degradation is about 17%, indicating that most aging occurred at the early usage phase of the design. Therefore, if there are no environmental or process variations, such degradation should provide great opportunities to identify recycled ICs by measuring one path delay from the

circuit. However, these variations have a significant impact on the path delay. On the other hand, different paths age at different rates as demonstrated earlier in this section. Figure 7.2(b) shows the delay of path P_i under different temperatures at different aging times. In the figure, AT denotes aging time, M represent months, and Y denotes years. From Figure 7.2(b), we can see that the delay of path P_i increases as we increase the temperature and paths age at different speed under different temperature.

To analyze variations' impact on P_i's delay, we perform Monte Carlo simulation using HSPICE on s38417. 300 Monte Carlo simulation results of P_i at 25°C are shown in Figure 7.3(a), with 3-sigma 2% T_{ox}, 5% V_{th}, and 5% L inter-die and 1% T_{ox}, 5% V_{th}, and 5% L intra-die process variations. We can see that P_i's delay varies around 12% due to process variations. In addition, process variations also have a significant impact on the aging rate of path delay, as shown in Figure 7.3(b). P_i's delay degradation in the 300 ICs varied around 8% (4% $\sim$ 12%) for one year of aging. *These variations evidently make the detection difficult, thus, the path delay shifts caused by aging effects in recycled ICs must be separated from those caused by process variations in new ICs if we are to use path-delay fingerprints to identify recycled ICs.*

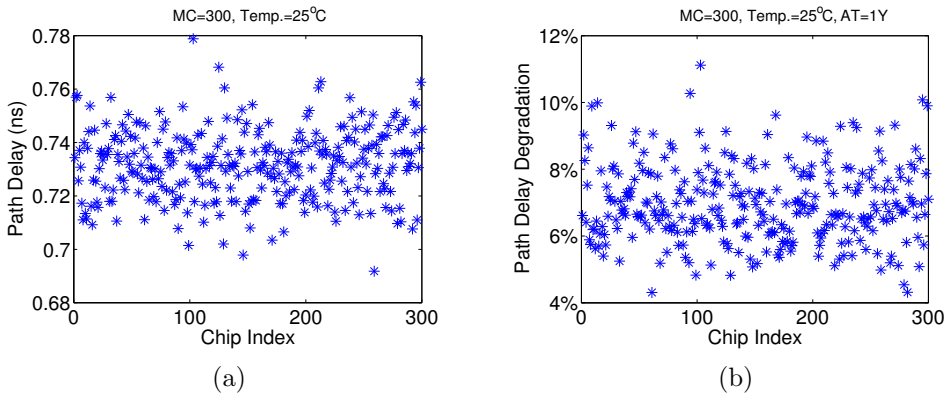


Fig. 7.3: (a) Delay of path P_i with process variations and (b) Delay degradation of path P_i changing with process variations.

7.2 Path-Delay Fingerprinting Considering Aging

Figure 7.4 shows our flow for identifying recycled ICs using path-delay fingerprints and statistical analysis. The proposed flow is divided into three major steps. First, paths are simulated and selected according to their aging rate. Next, the delay information of these paths are measured by a clock sweeping technique in new ICs (either during manufacturing test on all ICs or during authentication on a sample of new ICs) and in any available CUAs. Finally, statistical analysis is used to decide whether the CUAs are recycled ICs or not.

• **Step 1. Path Selection:** Due to the large number of critical and long paths in a circuit, in this step, we select paths which age at faster rates by analyzing the gate types in different paths and simulating the circuit with different workloads. Paths with higher rates of aging are preferred for fingerprint genera-

tion, since the differences in the delay of those paths between recycled ICs and new ICs will be much larger than the differences in paths which degrade slower. Fingerprints generated by fast-aging paths could help identify recycled ICs used for a shorter time. However, there are several parameters impacting the aging rate of a path, including the type of gates composing the path and the workload. Based on these parameters, and the observations made from simulation shown in Figure 7.1, we propose the following rules to select fast-aging paths: (i) paths with more fast-aging gates, such as NOR or XOR gates, will be selected, and (ii) paths that experience more zeros and more switching activity will be selected. More zeros in the path will increase the effect of NBTI on the PMOS transistors, and a high switching frequency will increase the HCI effects on gates, increasing the path delay degradation more significantly.

Paths with more fast-aging gates would be identified by analyzing the type of gates composing the paths. However, it is very difficult to identify paths that experience more zeros and more switching activity without knowing the specific workload. Therefore, in this work, different workloads (input combinations) are applied to ICs' primary inputs during logic simulation. For each gate on a critical path, the average switching activity and the zeros it has experienced are calculated. Paths with more switching activity and zeros are then selected using our flow. These paths, along with those composed of the more fast-aging gates, are used to generate fingerprints to identify recycled ICs. The number of selected

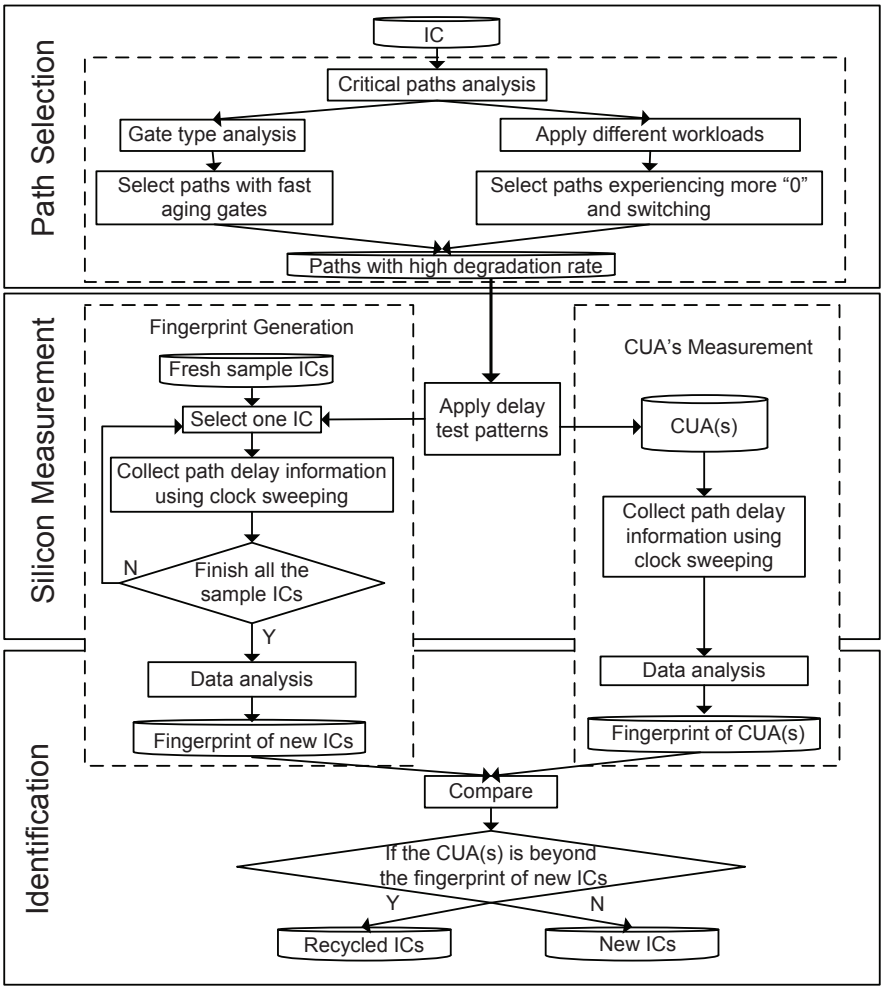


Fig. 7.4: Recycled IC identification flow.

paths could be adjusted according to the design and its testing procedure. In our simulation, we select the top 50 paths with fast-aging gates and the top 50 paths experiencing more switching activity and zeros in the benchmark circuit.

• **Step 2. Silicon Measurement:** The second step in Figure 7.4 is to collect the selected paths' delay from the ICs. Note that the fingerprint generation can be done during manufacturing test of a large sample of ICs before shipping

them to the market or on a number of new ICs from each production kept by the design house for the purpose of authentication or recycled ICs identification. The larger the size of sample is, the wider of a range of process variations will be included in the fingerprint, reducing the probability that we identify new ICs with large process variations as recycled ICs. Path delay information from the new ICs is measured by performing test procedures on the ICs. Traditionally, test patterns are generated by ATPG before fabrication to detect path and transition delay faults. These patterns will be applied to all new ICs using clock sweeping techniques [68] to measure the path delay of the targeted paths. Note that using clock sweeping is a common practice in industry for speed binning of ICs [68].

Figure 7.5 shows the flow of the clock sweeping technique. The delay test patterns are applied to ICs at different clock frequencies (f_1, f_2, ... f_n). Under different frequencies, the paths could pass or fail. If the time period t_i of the frequency f_i ($t_i = \frac{1}{f_i}$) is larger than the path delay, the path will pass. Otherwise, the path will fail. When a path fails, the largest passing frequency will determine the path delay. The frequency step size ($\Delta f = f_i - f_{i-1}$), which depends on the tester, will determine the accuracy of path delay measurement results of silicon chips. For example, with the Ocelot ZFP tester [69], the main frequency is 400MHz and the frequency step size is 1MHz. In our simulation, a 5MHz step size around 1.0GHz circuit frequency is used for the clock sweeping technique. The measurement environment should keep the temperature as stable as possible,

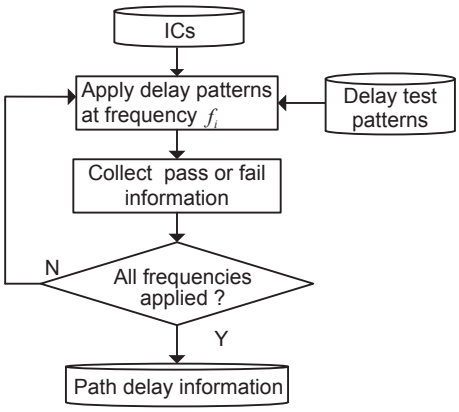


Fig. 7.5: Clock sweeping flow.

which can be controlled by the manufacturing test environment.

• **Step 3. Identification:** Once the path delay in all sample chips are measured, statistical data analysis will be used to generate a fingerprint for new ICs. For a circuit under authentication (CUA) taken from the market, the same test patterns will be applied in a near-identical environment. The path delay information of the CUA will be processed by the same statistical data analysis methods. In a simple analysis, if the fingerprint of the CUA is outside of the range of the new ICs' fingerprint, there is a high probability that the CUA is a recycled IC. Otherwise, the CUA is likely a new IC. The longer the CUA has been used, the more aging effects it will have experienced, making it easier to identify.

Without extra hardware circuitry embedded into the ICs, our recycled IC identification technique imposes no area or power overhead. It provides a negligible test time overhead during manufacturing test on a sample of ICs, since only a few patterns must be applied several times at different frequencies. Also, there

is no change in the current IC design and test flow since there is no additional circuitry in the IC used for detection. In addition, this method is resilient to tampering attacks. It is inherently difficult for recyclers to mask the impact of aging on the recycled ICs' path-delay fingerprints during the recycling process.

7.3 Statistical Data Analysis

Two statistical data analysis methods are used in this chapter: simple outlier analysis (SOA), and principal component analysis (PCA). When performing SOA, we randomly select a single path from the selected path set, and use its delay range in new ICs to generate a fingerprint. The process variations of the CUA may or may not be the same as those within the sample ICs. The selected path delay of the CUA and sample ICs will follow the same distribution, which makes SOA effective in certain conditions. However, a single-path based analysis will not be very effective, due to the limited aging information collected. In general, this method is expected to be effective in distinguishing recycled ICs used for a long time from new ICs with small process variations.

In order to improve the effectiveness of our technique, we also use PCA to separate the aging effects on path delay from process variations. The path delay information of all selected paths, which have been measured by clock sweeping, will be processed by PCA. In our simulations, the top 100 paths with faster aging rates were selected to generate fingerprints. The delay of each path is one of the

variables for PCA to use. Therefore, with N ICs, the dimension of the data set for PCA to generate fingerprint is N*100. The first three components of PCA in all new ICs were plotted, and a convex hull was created as the fingerprint for new chips. The path delay information of the CUA was also analyzed by the same process and plotted in the same figure. If the CUA is outside of the convex created by the new ICs, there is a high probability that the CUA is a recycled IC.

7.4 Results and Analysis

In order to verify the effectiveness of our recycled IC identification flow and data analysis methods, we implemented it using 45nm technology on a few benchmarks. HSPICE MOSRA [67] is used to simulate the effects of aging on the path delay of different benchmarks. The supply voltage of the 45nm technology is 1.1V. Random workloads (random functional input patterns) were applied to several ISCAS'89 benchmarks. Path delay information was collected using clock sweeping at different aging times. Different process and temperature variations were also simulated to analyze their impact on the effectiveness of our recycled IC identification method.

7.4.1 Process and Temperature Variations Analysis

Table 7.1 shows the three process variations rates we used in our simulations. Moving from PV0 to PV2, inter-die and intra-die variations both become larger.

Table 7.1: Process variation rates.

	Inter-die (3σ)			Intra-die (3σ)		
	V_{th}	L	T_{ox}	V_{th}	L	T_{ox}
PV0	3%	3%	2%	2%	2%	1%
PV1	5%	5%	2%	5%	5%	1%
PV2	8%	8%	2%	7%	7%	2%

PV1 represents a realistic rate of process variations that a foundry might have. Four sets of Monte Carlo simulation (MCS) were run using different levels of variations, as shown in Table 7.2. For each set of MCS, 300 Monte Carlo simulations were run to generate 300 chips. During the simulations, the aging effects of NBTI and HCI were simulated with random stress for the benchmark s38417. From the top 500 paths, the paths P_1, P_2,..., P_{50} with fast-aging gates and the paths $P_{51}, P_{52}, ..., P_{100}$ with more zeros and higher switching activities were selected to generate fingerprints.

Analysis using SOA: First, 300 Monte Carlo simulations were run in MCS1. The maximum aging time is 2 years. Here, SOA was used to process the path delay information. 3 paths (P_1, P_2, and P_{51}) were selected to show the results of SOA. Figures 7.6(a), 7.6(b), and 7.6(c) show the path delay distribution of the 3 paths from 300 ICs used for different aging times. Similar results were obtained for the other 97 paths as well. For each path, the range of the path delay at AT='0' is the fingerprint of the new ICs. If the path delay of the CUA is

Table 7.2: Simulation setup.

Experiments	Process Variations	Temperature
MCS1	PV0	25°C
MCS2	PV1	25°C
MCS3	PV2	25°C
MCS4	PV1	25°C ±10°C

out of that range, there is a high probability that IC is a recycled one. Note the 300 different Monte Carlo simulations are used for recycled ICs from those used as sample new ICs. From these figures, we can see that the delay distribution of each path in recycled ICs shifts to the right, relative to the distribution of delays in new ICs. This is because path delay in recycled ICs increases due to aging. The longer the ICs have been used, the more path delay degradation they will have experienced. In addition, we see that the path delay variation increases as the aging time increases. The reason for this is that ICs with different process variations age at different speeds, and the path delay variations become larger as we increase the aging time.

Figure 7.6(a) shows the distribution of path P_1's delay, and we can see that the smallest delay of P_1 in recycled ICs used for 1 month is smaller than the largest delay in new ICs. Therefore, the detection rate of recycled ICs used for 1 month is less than 100% (98.3%) when we use the fingerprint generated by SOA from path P_1. However, the detection rate of recycled ICs used for 3 months or longer is

100%, which demonstrates that it is easier to detect recycled ICs that have been used for longer amounts of time. If we choose path P_2 to detect recycled ICs, the detection rate of ICs used for 1 month (95.7%) is slightly less than when using path P_1. However, if path P_{51} is used, which has the fastest aging rate among the 100 paths, the detection rate is 100% even if the ICs are only used for one month. P_{51} is the most effective path for identifying recycled ICs in this benchmark. From the above analysis, we can see that different paths generate different fingerprints due to their different aging speeds, which makes SOA slightly less effective.

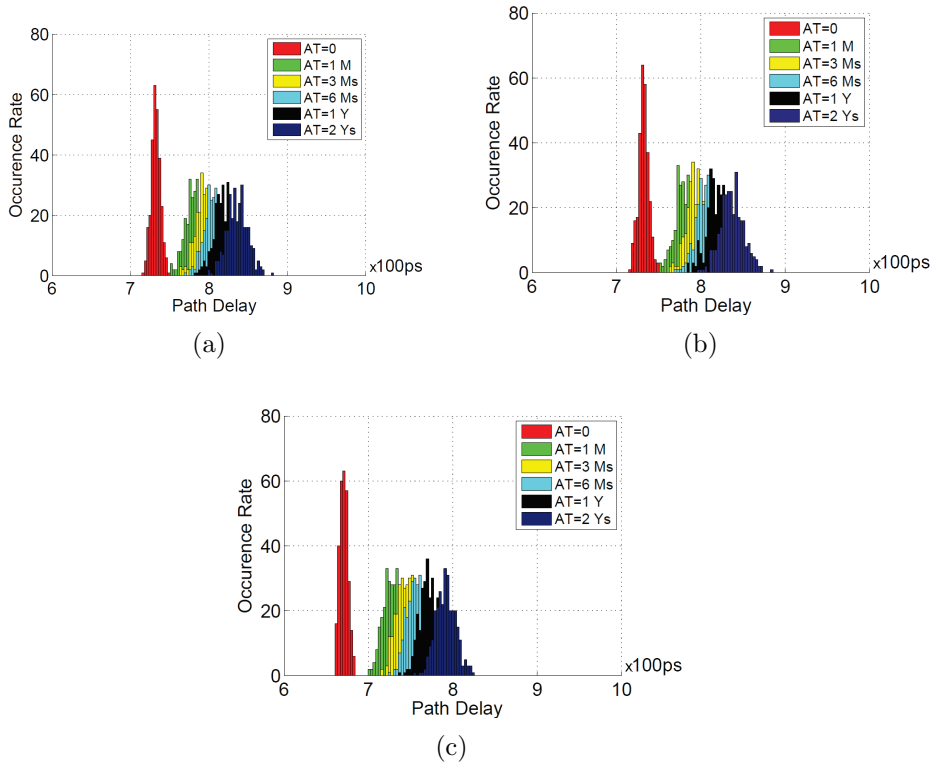


Fig. 7.6: Path delay distribution in ICs with PV0 in MCS1 at different aging times (a) Path P_1, (b) Path P_2, and (c) Path P_{51}.

Figures 7.7(a) and 7.7(b) show the delay distribution of path P_{51} across 300 Monte Carlo simulations in MCS2 and MCS3. Overall, Figures 7.6(c), 7.7(a), and 7.7(b) present the delay distribution of the same path (P_{51}) in ICs with different process variations. By comparing these figures, we can see that the larger the process variations are, the larger the path delay variations in new ICs will be, which makes it more difficult to detect recycled ICs. Even when using the most effective path P_{51}, the detection rates of ICs used for 1 month with PV1 and PV2 drop from 100% with PV0 to 78.0% and 50.7%, respectively. A 100% detection rate could be achieved if the ICs were used for 1 year or longer with PV1, or longer than 2 years with PV2.

300 Monte Carlo simulations were also run with ±10°C temperature variation during the aging process in MCS4 as shown in Figure 7.7(c). The measurement temperature is $25°C$. It shows the delay distribution of path P_{51} and the detection rate of ICs used for 1 month using it is 67.7%. Comparing Figure 7.7(c) and Figure 7.7(a), we can see that the larger the temperature variation is, the larger the path delay variation is, which makes it more difficult to detect recycled ICs.

Analysis using PCA: A similar analysis is done using PCA for different MCSs. Figure 7.8(a) shows the PCA results of the 100 paths in s38417 with 300 chips in MCS1. FC denotes the first component from PCA, SC represents the second component, TC is the third component, and DR denotes the detection

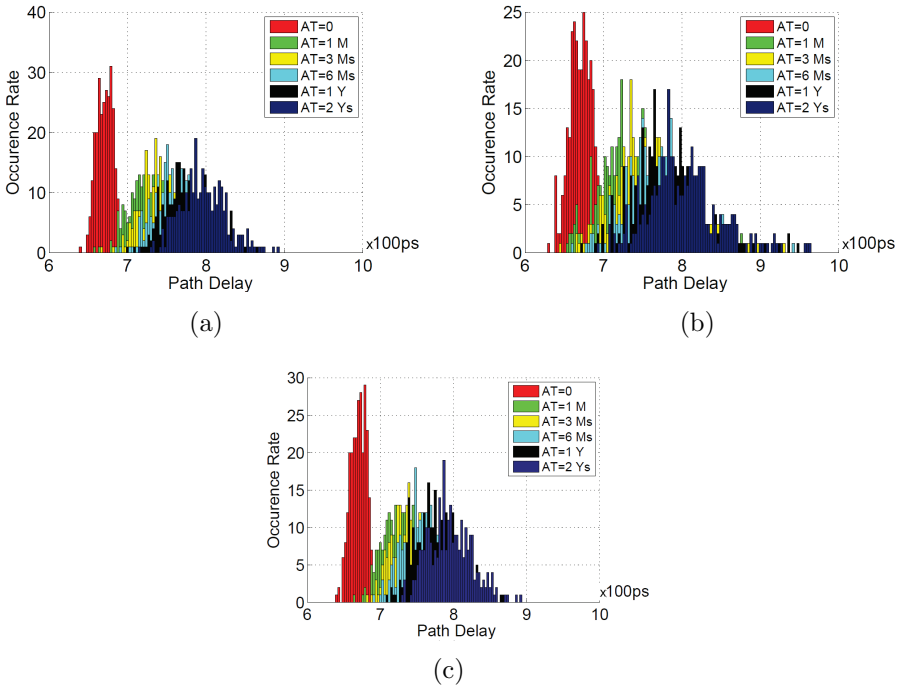


Fig. 7.7: Path P_{51} delay distribution in ICs at different aging times (a) in MCS2, (b) in MCS3, and (c) in MCS4.

rate. The convex is built up from new IC data, and represents the fingerprint for new ICs. The red asterisks represent chips used for 1 month. From the figure, we can see that the 300 used ICs were completely separated from the signature of the new ICs. Thus, the detection rate using path delay fingerprints generated by PCA is 100% for recycled ICs used for 1 month. For recycled ICs used for a longer time, the detection rate will obviously be 100% as well.

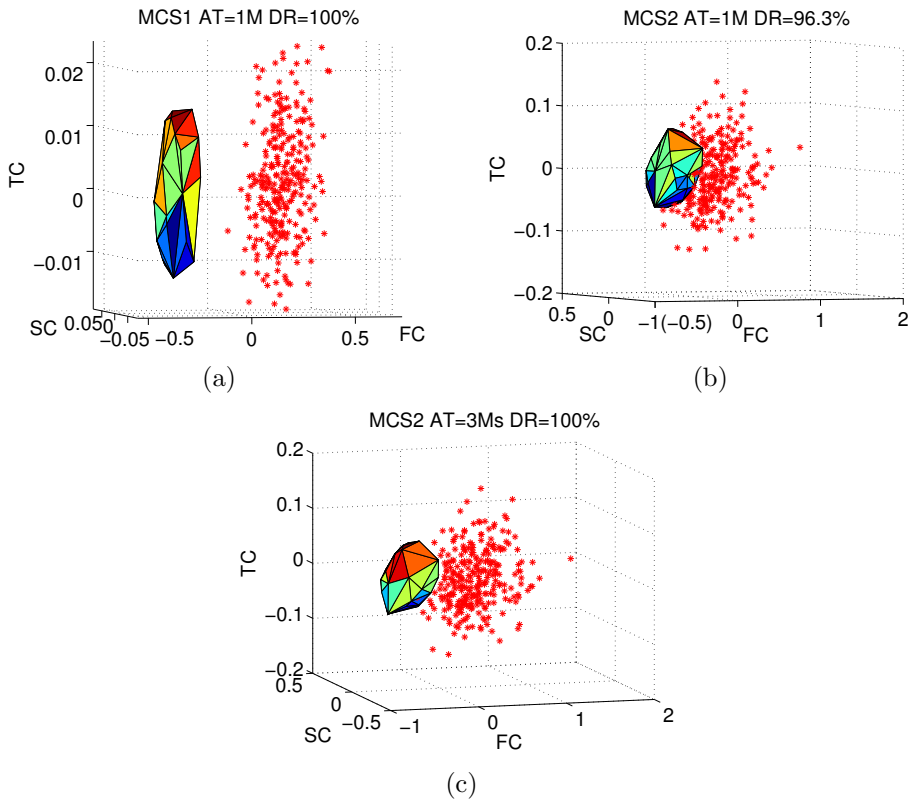
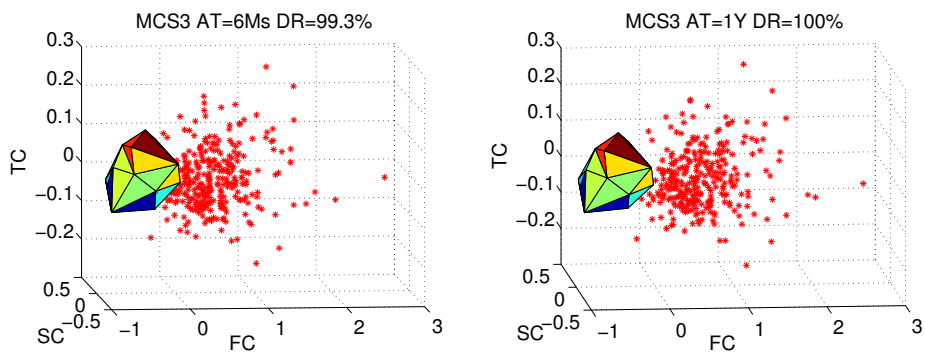


Fig. 7.8: PCA results of ICs under 25°C (a) used for 1 month with PV0 in MCS1, (b) used for 1 month with PV1 in MCS2, and (c) used for 3 months with PV1 in MCS2.

The path delay information from the remaining three sets of MCSs were also analyzed by PCA. Figure 7.8(b) shows the analysis results of new chips and recycled ICs used for 1 month in MCS2. From the 3-dimensional figure, we can see that some of the recycled ICs are close to the new ICs' fingerprint. The detection rate is 96.3%, which is much higher than using SOA. Comparing Figure 7.8(b) and Figure 7.8(a), we can see that (i) the convex hull built up from new ICs in MCS2 is much larger than that in MCS1 (note that the convex hull in MCS1 looks larger than MCS2 due to its small scale of axes), and (ii) the recycled ICs in MCS2 are closer to new ICs than those in MCS1, which makes the detection rate in MCS2 less than that in MCS1. The path delay information of 300 ICs used for 3 months in MCS2 were also processed, and the results are shown in Figure 7.8(c). Comparing Figures 7.8(b) and 7.8(c), we can see that the longer the chips have been used, the farther they will be from the new ICs' fingerprint. The detection rate of recycled ICs used for 3 months or longer with PV1 at 25°C is 100%.

Figure 7.9 shows the PCA results of ICs in MCS3. The detection rate of recycled ICs used for 1 month, 3 months, 6 months, and 1 year are 72.7%, 89.3%, 99.3%, and 100%, respectively. The figures of PCA results of recycled ICs used for 1 month and 3 months are not shown here since the detection rates are so far from 100%. Figures 7.9(a) and 7.9(b) show the new ICs' fingerprint and the recycled ICs used for 6 months and 1 year, respectively. The recycled ICs used for longer times are easier to detect, as seen by comparing Figures 7.9(a) and 7.9(b).

Comparing the detection rates in these simulations, we can see that it is more difficult to detect recycled ICs which have higher levels of process variations. The 99.3% detection rate of ICs used for 6 months and the 100% detection rate of ICs used for 1 year in MCS3 shows the effectiveness of our technique. We acknowledge that PV2 is an extremely high variation compared to what is expected in practice (e.g., PV1).



(a) (b)

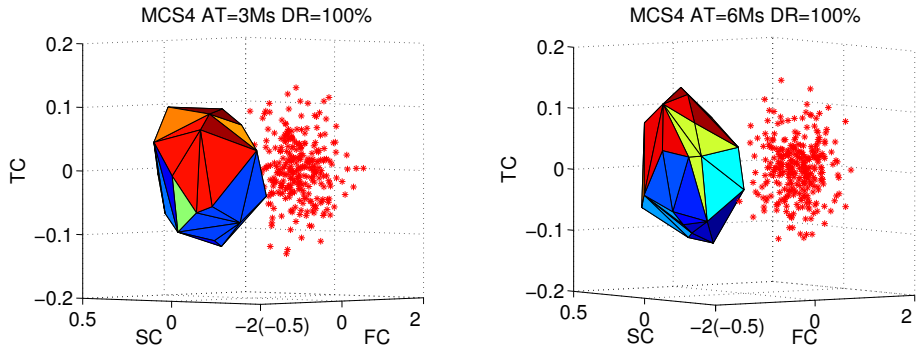
Fig. 7.9: PCA results of ICs with PV2 under 25°C in MCS3 used for (a) 6 months and (b) 1 year.

With the same measurement temperature 25°C, ±10°C temperature variation is used in MCS4 during the aging process. The detection rate of ICs used for 1 month, 3 months, and 6 months in MCS4 are 90.6%, 100%, and 100%, respectively. The new ICs' fingerprint and the detected recycled ICs used for 3 months and 6 months are shown in Figure 7.10. Comparing Figure 7.10(a) with Figure 7.8(c), we can see that the recycled ICs used for 3 months in MCS4 are closer to the fingerprint than recycled ICs used for 3 months in MCS2. This phenomenon

Table 7.3: Recycled IC detection rates for s38417.

	SOA				PCA			
	1M	3M	6M	1Y	1M	3M	6M	1Y
MCS1	100%	100%	100%	100%	100%	100%	100%	100%
MCS2	78%	96.7%	99.7%	100%	96.3%	100%	100%	100%
MCS3	50.7%	76.3%	85.3%	95.6%	72.7%	89.3%	99.3%	100%
MCS4	67.7%	93.3%	98%	100%	90.6%	100%	100%	100%

demonstrates that temperature variations could increase the path delay variations in new ICs and make it more difficult to detect recycled ICs. However, the 100% detection rates of ICs used for 6 months in MCS4 demonstrates the effectiveness of our method with process and temperature variations.



(a) (b)

Fig. 7.10: PCA results of ICs with PV1 and ±10°C temperature variations in MCS4 used (a) 3 months, and (b) 6 months.

Figures 7.7 through 7.10 presented some detailed results relating to using this technique on s38417 with SOA and PCA. Table 7.3, however, tabulates these

Table 7.4: Recycled IC detection rates - benchmark comparison under MCS4 using PCA.

Benchmark	1M	3M	6M	1Y
s9234	88%	100%	100%	100%
s13207	89.6%	100%	100%	100%
s38417	90.6%	100%	100%	100%

results in addition to some other results obtained using both statistical analysis approaches. These results clearly demonstrate that PCA is more effective than SOA when it comes to identifying ICs used for shorter periods of time.

7.4.2 Benchmark Analysis

In addition to s38417, the ISCAS'89 benchmarks s9234 and s13027 were also simulated to demonstrate the efficiency of this technique on different designs. The process variation and temperature variation rates used in MCS4 were applied to these two benchmarks. The aging stress causing NBTI and HCI degradation in these benchmarks comes from random workloads. 300 MCS were run for each benchmark for a maximum 2 years of aging. The path selection method was also applied to these benchmarks, and 100 paths from each benchmark were used to run statistical data analysis using PCA.

Table 7.4 shows the recycled IC detection rate for all three benchmarks under MCS4 for up to a year of aging. The detection rate for ICs used for 3

months in the benchmarks s9234 and s13207 is 100%, which matches the results obtained from s38417.

The results shown in this section clearly demonstrate that our recycled IC detection method using a path delay fingerprint generated by PCA is very effective, even in designs with large process and temperature variations.

7.5 Conclusions

We presented a recycled IC identification method using path-delay fingerprinting in this chapter. Paths with fast aging speed were selected to generate a fingerprint for the chip under authentication. The path delay signature from a recycled IC is beyond those from new ICs due to aging. With no additional hardware circuitry required, this method provides no overhead on area and power consumption. The simulation results of different benchmarks with different process and temperature variations demonstrated the effectiveness of our method.

Chapter 8

High Performance True Random Number Generator

True random number generator is another important security module integrated in most ICs for secure data communication and storage. TRNG is frequently used in the generation of (*i*) public/private keypairs for cryptographic protocols, such as RSA, DSA, and Diffie-Hellman; (*ii*) initialization vectors or seeds for randomness requirement structures; (*iii*) private keys for digital signature algorithms; (*iv*) challenges to be used in entity authentication mechanisms; (*v*) values to be used in key establishment protocols; and (*vi*) passwords, padding bytes, blinding value, cookies and nonces. In order to improve the security of ICs, we propose novel TRNGs to generate random numbers with better randomness for different applications in this chapter.

8.1 Basic structure of TRNG

Generally, true random number generators produce randomness by using a non-deterministic source, such as resistance noise, atmospheric noise, or nuclear decay. They usually follow a generic structure, composed of noise source, digitizer, post-

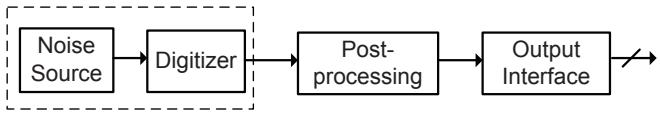


Fig. 8.1: The generic structure of TRNG.

processing module, and output interface (shown in Figure 8.1) [32]. Normally, noise source uses some non-deterministic physical phenomenon to generate analog signal, which is digitized by a digitizer. The purpose of the post-processing module is to make sure that the probability distribution of the internal random sequence is close to uniform distribution. The post-processing module is also used to increase the randomness of the generated sequence by applying a compression function on its input. One of the typical post-processor is XOR corrector, which will be used in our proposed TRNG. The output interface slows down the bitstream and sends out the random sequence.

The randomness of the data generated by TRNG is traditionally evaluated by statistical tests provided by National Institute of Standards and Technology (NIST) [34] or other test suites. If statistical tests from NIST are used, a P-value ($0 =< P - value =< 1$) will be generated to indicate randomness of sequence under test: (i) if P-value=1, the sequence appears to be perfectly random; (ii) if P-value=0, the sequence appears to be completely non-random; and (iii) if a P-value>=0.01, the sequence is considered to be random [34]. Therefore, when we design a TRNG, we would like increase the P-value of random sequence generated by the proposed TRNG.

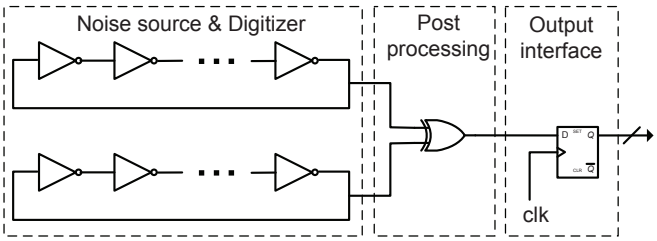


Fig. 8.2: B-TRNG.

A TRNG structure (B-TRNG) sampling the phase jitter of digital ring oscillators was proposed to generate true random numbers in [33] (shown in Figure 8.2). Due to the unstable balance and small stimulus such as process and environmental noise, the ring oscillator will enter metastability state and start to oscillate. The metastability will also create uncertainty of the ring oscillator, resulting in phase jitter. In order to verify that, we run simulations using HSPICE with 90nm technology. The simulated circuit is composed of two 13-stages ring oscillators, an XOR gate, and one flop-flop (FF). Power supply noise shown in Figure 8.3 is used in the simulation, changing from 1.2v to 0.9v within 10ns. The waveform of different signals in B-TRNG are shown in Figure 8.4. From the figure, we can see that the phase of ring oscillators is approximately random due to the randomness of the noise. XOR gate and the FF are used to capture the randomness and generate a random sequence.

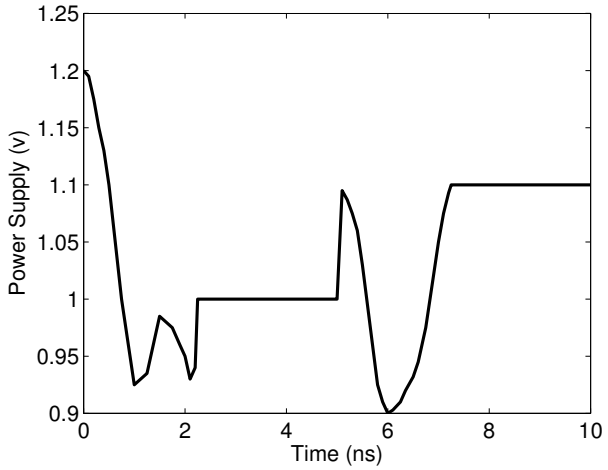


Fig. 8.3: Power supply noise for B-TRNG

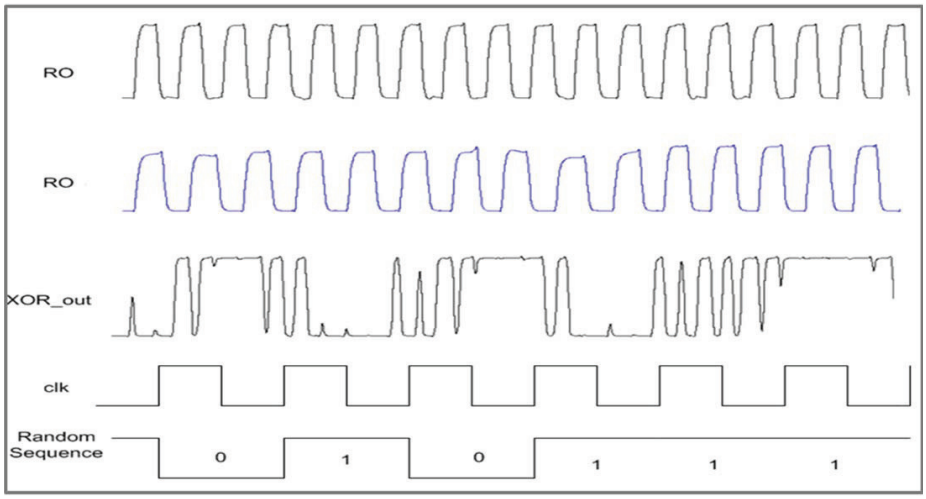


Fig. 8.4: The waveform of signals in B-TRNG

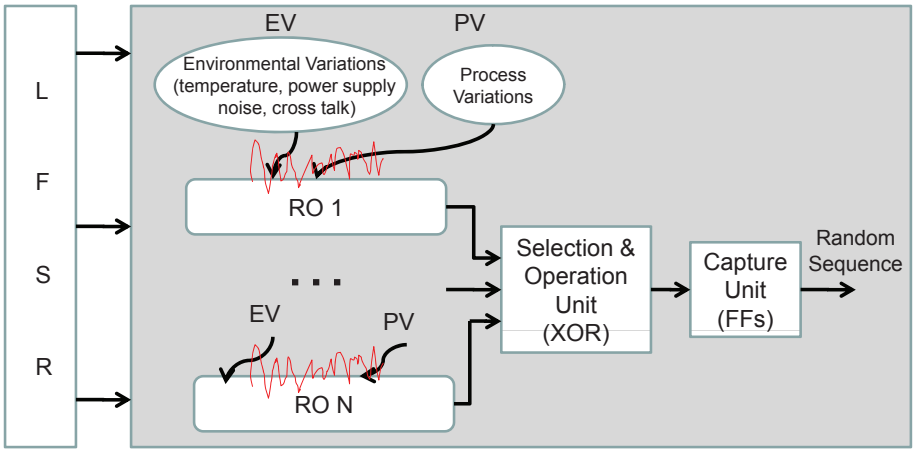


Fig. 8.5: The structure of BN-TRNG

8.2 Proposed TRNG

In order to increase the randomness of the sequence, we propose two new TRNG structures to generate random numbers based on increased environmental variations: (*i*) benchmark noise TRNG (BN-TRNG) increases the randomness of the generated sequence by capturing random noise created by the benchmark and (*ii*) ring oscillator noise TRNG (RN-TRNG) increases the randomness of the generated sequence by capturing random noise created by surrounding ring oscillators. The generic structure of the proposed BN-TRNG is shown in Figure 8.5. The BN-TRNG is composed of an LFSR, a set of ring oscillators, XOR gates, and FFs. The LFSR is used to generate random patterns to invoke different functions of the circuit. This causes an increase in temperature variations, power supply noise, and cross talk, which will in turn increase the randomness of the sequence.

The structure of the proposed RN-TRNG is shown in Figure 8.6. It is com-

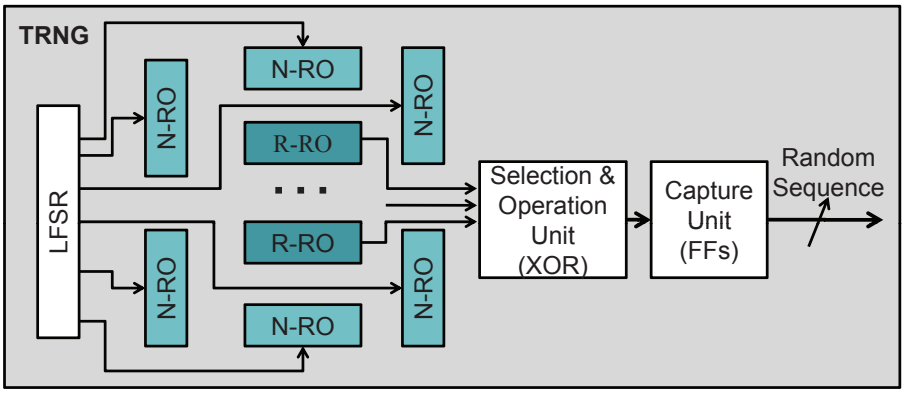


Fig. 8.6: The structure of RN-TRNG

posed of an LFSR, noise ring oscillators (NROs), random ring oscillators (RROs), an XOR gate, and a FF. One of the inverters in these ring oscillators is replaced with a NAND gate so that they can be enabled or disabled separately to reduce power consumption. When RN-TRNG works in the random number generation mode, NROs will be enabled or disabled randomly by the LFSR. Therefore, more random power supply noise and temperature variations will be introduced to the circuit since these NROs are oscillating at a very high frequency. Then RROs will capture these random noise and the randomness of their phase jitter will be increased. In addition, NROs are placed around RROs to increase the impact of NROs on RROs. Consequently, compared to BN-TRNG, the sequence generated by RN-TRNG have higher randomness due to noise introduced by NROs. Furthermore, the area overhead caused by RN-TRNG is negligible in modern designs with millions of gates. Power consumption of RN-TRNG is also limited since all ring oscillators are disabled when RN-TRNG is disabled.

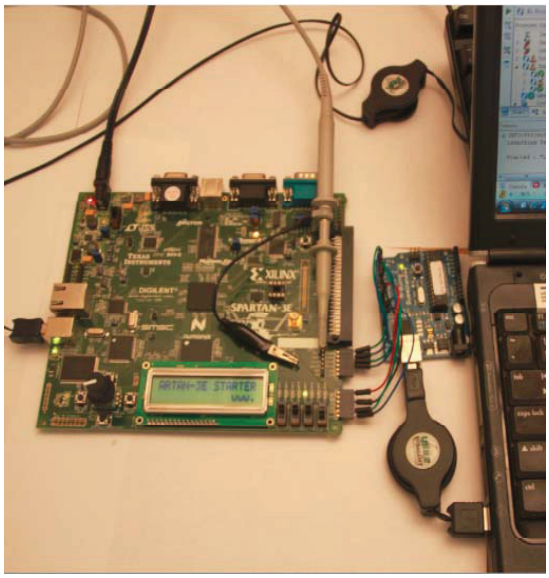


Fig. 8.7: Experimental Setup

8.3 Experimental Results and Analysis

In order to verify the effectiveness of our proposed structures, we implement our BN-TRNG and RN-TRNG on Xilinx Spartan-3E FPGA boards. The experimental setup is shown in Figure 8.7. An Atmel Atmega328P microcontroller is connected to the FPGA to facilitate in the collection of generated random sequence. In our experiments, the sequence collected from the FPGA boards will be evaluated by test suite sts-2.1.1 provided by NIST [34]. Based on the specification, the suite should be invoked with a stream length of 1,000,000 bits [34]. Therefore, 1,000,000 bits will be collected from the FPGA boards from each measurement.

With the test suit sts-2.1.1, different algorithms could be used to evaluate the randomness of generated random sequence. Figure 8.8 shows part of the report

generated by sts-2.1.1. From the figure, we can see that different algorithms (statistical tests) reports different P-values for a given sequence. In our experiments, we will report the P-value generated by serial statistical test due to its popularity. If the P-value of the 1,000,000 bit stream is close to "1", it demonstrates that our proposed TRNGs can generate random numbers with high randomness.

Four test cases with different configuration of BN-TRNG and RN-TRNG are implemented in our experiments to evaluate and compare the randomness of generated sequence: (*i*) Case 1: BN-TRNG is implemented on benchmark AES with LFSR disabled. (*ii*) Case 2: BN-TRNG is implemented on benchmark AES with LFSR enabled. (*iii*) Case 3: RN-TRNG is implemented without benchmark with LFSR disabled. (*iv*) Case 4: RN-TRNG is implemented without benchmark with LFSR enabled.

From the above description, we can see that (*i*) the same BN-TRNG structure is implemented in Case 1 and Case 2, and (*ii*) the same RN-TRNG structure is implemented in Case 3 and Case 4. The difference is related to LFSR. In Case 1 and Case 3, LFSR is disabled while in Case 2 and Case 4, the LFSR is enabled. Since LFSR will create more random noise in the circuit, we expect the cases with LFSR enabled to give higher P-values. The feedback polynomial of the LFSR used in our experiments is

$$X^7 + X^3 + 1 \tag{8.1}$$

The 8-bit LFSR supplies random patterns for benchmark AES in Case 1 and Case

01:	**P-value**		**Statistical Test**
02:	0.000000		Frequency
03:	0.008879		LongestRun
04:	0.122325		Rank
05:	0.066882		FFT
06:	0.739918		Serial
07:	0.213309		LinearComplexity
...	...	...	...

Fig. 8.8: Part of P-value report generated by test suite sts-2.1.1

2. It is also used to enable or disable the noise generation ring oscillators in Case 3 and Case 4. The seeds for the LFSR are stored in RAM in the FPGA board.

For each case, 1 million bits of data are collected and analyzed by test suite sts-2.1.1. The evaluation results are shown in Table 8.1. From the table, we can see that the sequence generated by RN-TRNG with LFSR disabled has the lowest randomness due to limited random noise introduced by the original environment. When LFSR is disabled (noise ring oscillators are quite), the RN-TRNG is reduced to a B-TRNG. Therefore, we can conclude that B-TRNG generates sequence with the lowest randomness. By comparing the P-value in Case 1 and Case 2, we can see that random patterns generated by LFSR can introduce more random noise to the circuit, thereby increasing the randomness of generated sequence. The table also shows that the P-value of the sequence generated in Case 4 is 0.9114, which is very close to 1. That means that with LFSR enabled, RN-TRNG could

Table 8.1: Evaluation results of different TRNGs by using sts-2.1.1.

Case #	TRNG	LFSR	Benchmark	Noise	Data Size	P-value
Case 1	BN-TRNG	Disable	AES	Original environment	1 million bits	0.3505
Case 2	BN-TRNG	Enable	AES	Original environment +Benchmark Noise	1 million bits	0.7399
Case 3	RN-TRNG	Disable	-	Original environment	1 million bits	0.1223
Case 4	RN-TRNG	Enable	-	Original environment + Noise generated by NROs	1 million bits	0.9114

generate sequence with almost perfect randomness. The reason for that is the noise generation ring oscillators in RN-TRNG controlled by the LFSR introduce lots of random noise into ICs.

8.4 Conclusions

In this chapter, we presented two TRNGs that can generate sequence with high randomness We used noise generation ring oscillators and benchmarks to introduce more random noise into the TRNG. The improved TRNG has very small area and power overhead compared to the basic TRNG. Using test suite from NIST to evaluate the randomness of the generated sequence, we found that our proposed TRNG structure can generate sequences with much higher randomness than the basic TRNG.

Chapter 9

Conclusions and Future Research

ICs are becoming increasingly vulnerable to malicious activities and alterations due to the globalization of the semiconductor design and fabrication process. To address this issue, we have presented several techniques for hardware Trojan detection, recycled ICs detection, and true random number generation to improve the trustworthiness, security, and reliability of ICs in this thesis. The major contributions of the thesis will be presented in this chapter. Future research for hardware Trojan detection, recycled IC detection, and true random number generator will also be discussed.

9.1 Summary of Contributions

9.1.1 Hardware Trojan Detection

Hardware Trojan Detection in 3PIPs: due to the complexity of IP trust problem , there is no silver bullet available. In this thesis, we conducted the first case study for hardware Trojan detection in 3PIPs based on formal verification

and code coverage analysis. For 3PIPs, the only trust source is the specification from IP buyers. Each item in the specification is translated into a property in the test bench. Functional coverage reports whether the assertion, translated from the property, is successful or not. Code coverage reports which lines and statments in the design are executed during verification. If the code coverage is 100% and all the assertions in the test bench are successful, we can conclude that the 3PIP is Trojan-free. Otherwise, the uncovered parts are suspicious. Then redundant circuit removal, sequential ATPG, and equivalence theorems are used to reduce the number of suspicious parts in the design.

This work is the first time that formal verification and code coverage analysis are used to verify the trustworthiness of 3PIPs. The case study and proposed flow not only focus on the RTL-level verification, but also on the gate-level analysis and ATPG, thereby extending the research area for hardware Trojan detection in 3PIPs. More solutions could be developed by following our case study and proposed flow.

Hardware Trojan Detection in ICs: we proposed an effective structure to detect hardware Trojans inserted into ICs. The RON architecture generates a power fingerprint, used to identify malicious alterations. We also proposed a framework combining the improved RON with off-chip current measurements. In the improved RON, the n-stage ROs were placed with one component located in each of the n rows of the standard cell design. All the rows in the design are

covered by ROs to ensure complete coverage of the power distribution network. We showed that our technique has the capability of detecting very small Trojans with very little contribution to circuit transient current. Advanced outlier analysis algorithm was developed to distinguish the effects of hardware Trojans from process variations.

In addition, the proposed RON structure was analyzed using 38 ICs containing the ISCAS s9234 benchmark circuit fabricated using the IBM 90nm process. We showed that ring oscillator frequencies increase with increasing Trojan partial activity and that ring oscillators which share power lines with nearby Trojans will be most impacted. The silicon results also demonstrated that even in the presence of obfuscating process variations, measurement noise, and environment variation ICs may still be effectively classified using a PCA-based classification technique.

9.1.2 Recycled ICs Detection

In this thesis, we defined the recycled ICs problem and proposed three light-weight on-chip sensors and a path-delay fingerprinting flow to detect recycled ICs. The frequency difference between the Reference RO and the Stressed RO in the RO-based sensor makes recycled ICs identification possible. We showed that the RO-based sensor is very effective to detect recycled ICs by placing the two ROs next to each other. In addition, the usage time stored in AF-based sensors could indicate how long an IC has been used and then identify the recycled IC. The

effectiveness of AF-based sensors will not be impacted by technologies, packages, assemblies, or process variations. Even if the design was fabricated at different time in different foundries, AF-based sensors can still indicate how long the chip under test has been used.

We also presented a recycled ICs identification method using path-delay fingerprinting. With no additional hardware circuitry required, this method provides no overhead on area and power consumption. The simulation results of different benchmarks with different process and temperature variations demonstrated the effectiveness of our methods.

9.1.3 True Random Number Generator

Based on the generic structure of TRNG, we proposed two TRNGs that can generate sequence with high randomness. The noise generation ring oscillators and benchmarks, both of which are randomly controlled by LFSR, introduce more random noise to the circuit. We demonstrated that the random noise can increase the randomness of generated sequences.

9.2 Future Research

The security and reliability of ICs will be of utmost importance in the future due to the fast IC development. More issues might appear and cause damages to ICs and their applications. However, currently most research still focuses on hardware

Trojan detection, IC authentication, and recycled ICs detection. In this section, we will share our ideas about future research directions of these topics.

9.2.1 Hardware Trojan Detection

Hardware Trojan Detection in 3PIPs: for hardware Trojan detection in 3PIPs, there are several reseach directions:

- Improving test benches to achieve 100% code coverage. With 100% code coverage and assertion analysis, suspicious parts could be easily identified.
- Extending the research area. In this thesis, the trustworthiness of 3PIPs was verified not only by RTL-level verfication but also by gate-level analysis. More solutions could be developed by following this direction. On the other hand, software Trojan detection methods could be used to detect hardware Trojans in 3PIPs since software Trojans and hardware Trojans in 3PIPs are both inserted into codes.

Hardware Trojan Detection in ICs: future research about hardware Trojan detection in ICs includes:

- Developing methods to detect hardware Trojans without golden ICs. In this thesis, we assume that a signature for Trojan-free ICs could be generated from trusted ICs. However, hardware Trojans could be inserted into all the fabricated ICs, which makes the generation of the Trojan-free signature is

impossible. Without Trojan-free ICs (golden ICs), new techniques need to be developed. Inserting reference circuits into ICs could be one solution.

- Proposing new techniques to prevent hardware Trojan insertion. Right now, most research focuses on hardware Trojan detection. However, it would be better if we can prevent hardware Trojan insertion during IC fabrication process.

9.2.2 Recycled ICs Detection:

Since recycled ICs is a very new problem, there are several research directions related to this topic.

- Classifing recycled ICs into different categories. Since different recyclers could recycle ICs in different ways, recycled ICs could be classified into simple cleaned recycled ICs, professional remarked recycled ICs, and altered recycled ICs. With such classification, the future work for recycled ICs would be easier.

- Developing recycled ICs detection using other side-channel information, such as leakage current and transient current. In this thesis, we focus on the path-delay fingerpriting flow to identify recycled ICs. We believe that our flow could also be effective by using leakage current and transient current.

- Proposed techniques to detected recycled analog and memory devices. Since

these recycled devices have been used, techniques based on performance degradation must be effective to detect them.

9.2.3 True Random Number Generator

In the future, TRNGs with P-value "1" will need to be developed by increasing random noise. Each component in the generic structure could be improved to increase the randomness of the generated sequence, such as digitizer, post-processing module, and output interface.

9.3 Conclusions

This thesis is devoted to developing on-chip structures and techniques to improve the security, trustworthiness, and reliability of ICs. Hardware Trojan insertion and recycled ICs counterfieting are the two major malicious activities that endanger ICs used in critical applications. Since the presence of hardware Trojans can have an impact on certain parameters of their neighboring cells and the entire circuit, it is possible to detect them based on the impact. However, the impact of hardware Trojans with a very small number of gates could be too small to be measured by using off-chip equipments. A verification based flow and on-chip structures were developed to address this issue. Trojan-inserted 3PIPs and ICs can be identified by using our techniques. The area overhead and test overhead of our methods are negligible compared to modern designs with millions of gates. Moreover, recycled

IC problem is defined in this thesis. Performance degradation in recycled ICs provides an opportunity to identify them. Our proposed on-chip sensors and path-delay fingerprinting flow can easily identify whether the chip under authentication is used or not. In conclusion, we presented different techniques and strategies to solve hardware Trojan and recycled IC problems in this thesis. However, there still can be future research directions to improve the security, trustworthiness, and reliability of ICs. Researchers can bring IC security to the next level by developing new solutions.

Bibliography

[1] "Report of the Defense Science Board Task Force on High Performance Microchip Supply," Defense Science Board, US DoD, http://www.acq.osd.mil/dsb/reports/2005-02-HPMSi_Report_Final.pdf, Feb, 2005.

[2] "Defense Industrial Base Assessment: Counterfeit Electronics," Bureau of Industry and Security, U.S. Department of Commence, *http://www.bis.doc.gov/defenseindustrialbaseprograms/osies/defmarketresearchrpts/final_counterfeit_electronics_report.pdf*

[3] Business Week, "Dangerous Fakes," *http://www.businessweek.com/magazine/content/08_41/b4103034193886.htm,* 2008.

[4] L. W. Kessler and T. Sharpe, "Faked Parts Detection," *http://www.circuitsassembly.com/cms/component/content/article/159/9937-smt,* 2010.

[5] J. Stradley and D. Karraker, "The Electronic Part Supply Chain and Risks of Counterfeit Parts in Defense Applications," *IEEE Transactions on Components and Packaging Technologies*, pp.703-705, Sept. 2006.

[6] Military Times, "Officials: Fake Electronics Ticking Time Bombs," *http://www.militarytimes.com/news/2011/11/ap-fake-electronics-ticking-time-bomb-110811/*, 2011.

[7] Tezzaron Semiconductor, "3D-ICs and Integrated Circuit Security," *http://www.tezzaron.com/about/papers/3D-ICs_and_Integrated_Circuit_Security.pdf*, 2008.

[8] *http://www.combatcounterfeits.com/gallery.htm.*

[9] "Counterfeit Electronic Parts; Avoidance, Detection, Mitigation, and Disposition", *http://standards.sae.org/as5553/.*

[10] M. Tehranipoor and C, Wang "Introduction to Hardware Security and Trust," Springer, New York, USA, 2011.

[11] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in Proc. *IEEE Symposium on Security and Privacy (SP)*, pp. 296-310, 2007.

[12] R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals," in Proc. *IEEE Int. Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 3-7, June, 2008.

[13] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans," in Proc. *IEEE Int. Conference on Computer-Aided Design*, pp. 10-13, Nov. 2008.

[14] M. Potkonjak et al., "Hardware Trojan Horse Detection Using Gate-Level Characterization," in Proc. *Design Automation Conf. (DAC 09)*, ACM Press, pp. 688-693, 2009.

[15] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware Trojan Detection and Isolation using Current Integration and Localized Current Analysis," in Proc. *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFTVS08)*, pp. 87-95, 2008.

[16] Y. Jin and Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint," in Proc. *IEEE HOST*, pp. 51-57, 2008.

[17] J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection," in Proc. *IEEE HOST*, pp.8-14, 2008.

[18] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," in Proc. *IEEE Int. Hardware-Oriented Security and Trust (HOST)*, pp. 15-19 2008.

[19] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design and Test of Computers*, pp. 10-25, 2010.

[20] S. Narasimhan, D. Dongdong, R. Chakraborty, S. S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia, "Multiple-parameter Side-channel Analysis: A non-invasive Hardware Trojan Detection Approach," in Proc. *IEEE HOST*, pp. 13-18, 2010.

[21] L. Leinweber, C. A. Papachristou, and S. Bhunia, "Towards Trojan-free Trusted ICs:Problem Analysis and Detection Scheme," in Proc. *Design, Automation and Test in Europe(DATE)*, pp. 1362-1365, 2008.

[22] S. Jha and S. K. Jha, "Randomization Based Probabilistic Approach to Detect Trojan Circuits," in Proc. *IEEE High Assurance System Engineering Symposium*, pp. 117-124, 2008.

[23] M. Banga and M. Hsiao, "A Region based Approach for the Identification of Hardware Trojans," in Proc. *IEEE HOST*, pp. 40-47, 2008.

[24] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards Trojan-free Trusted ICs: Problem Analysis and Detection Scheme" in Proc. *Design, Automation and Test in Europe (DATE)*, pp. 1362-1365, 2008.

[25] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," *IEEE Transactions on VLSI*, 2011.

[26] L. Kim, J. Villasenor, and C. K. Koc, "A Trojan-resistant system-on-chip Bus Architecture," *Proceedings of IEEE Military Communication (MILCOM)*, Boston, Oct. 2009.

[27] M. Abramovici and P. Bradley, "Integrated Circuit Security: new Threats and Solutions," in *5th Annual Workshop on Cyber Security and information intelligence Research : Cyber Security and information intelligence Challenges and Strategies*, pp. 13 - 15, April. 2009.

[28] A. J. Hu, "Formal Hardware Verification with BDDs: An Introduction," *IEEE*, 1997.

[29] Synopsys, "The Synopsys Verification Avenue Technical Bulletin", Vo1.4,issue 4, December 2004.

[30] I. Ugarte and P. Sanchez, "Formal Meaning of Coverage Metrics in Simulation-Based Hardware Design Verification," IEEE, 2005.

[31] *ttp://trust-ub.org/resources/benchmarks.*

[32] D. Schellekens, B. Preneel, and I. Verbauwhede, "FPGA Vendor Agnostic True Random Number Generator," *Int. Conference on Field Programmable Logic and Applications*, pp.1-6, 2006.

[33] B. Sunar, W. J. Martin, and D. Stinson, "A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks," *http://www.cacr. math.uwaterloo.ca/dstinson/papers/rng-IEEE.pdf*, 2005.

[34] *ttp://csrc.nist.gov/groups/ST/toolkit/rng/index.tml.*

[35] F. Koushanfar "Hardware Metering: A Survey," *http://aceslab.org/sites/default/files/05-fk-metering.pdf.*

[36] Y. Wang, S. Cotofana, and L. Fang "A unified Aging Model of NBTI and HCI Degradation towards Lifetime Reliability Management for Nanoscale MOSFET Circuits," *IEEE Int. Symposium on Nanoscale Architecture,* 2011.

[37] *http://www.eetimes.com/design/memory-design/4376742/Anti-fuse-memory-provides-robust--secure-NVM-option.*

[38] *http://www.sidense.com/technology.html.*

[39] *http://www.kilopass.com/products/otp-memory-ip/xpm-otp-nvm/.*

[40] N. Reddy, S. Wang, L. Winemberg, and M. Tehranipoor, " Experimental Analysis for Aging in Integrated Circuits," *IEEE North Atlantic Test Workshop (NATW)*, 2011.

[41] M. Bushnell and A. Vishwani, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits," 2000.

[42] S. Zhao, K. Roy, and C. Koh, "Frequency Domain Analysis of Switching Noise on Power Supply Network," *Tecnical Reports*, 2000.

[43] I. T. Jolliffe, "Principal Component Analysis (2ed Edition)," Springer, pp. 150-165, 2002.

[44] F. P. Preparata and S. J. Hong, "Convex Hulls of Finite Sets of Points in Two and Three Dimensions," *Commun. ACM*, vol. 20, no. 2, pp. 8793, 1977.

[45] S.H.K. Embabi. "Digital BiCMOS Integrated Circuit Design." Kluwer, 1993.

[46] T. Sakurai and R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584-594, Apr. 1990.

[47] *http://digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2.*

[48] K. Lofstrom, W. R. Daasch, and D. Taylor, "IC Identification Circuit Using Device Mismatch," in *Proc. ISSCC*, pp. 370-371, 2000.

[49] R. Pappu, "Physical One-way Functions," *Phd thesis, MIT*, 2001.

[50] G. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Proc. DAC*, pp. 9-14, 2007.

[51] E. Ozturk, G. Hammouri, and B. Sunar, "Physical Unclonable Function with Tristate Buffers," in *Proc. ISCAS*, pp. 3194-3197, 2008.

[52] A. Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-Friendly Secure Primitive," *IACR Journal of Cryptology, special issue on Secure Hardware*, 2011.

[53] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," in proc. *DATE08*, pp. 1069-1074, 2008.

[54] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design & Test of Computers*, 2010.

[55] T. Kim, R. Persaud, and C. H. Kim, "Silicon Odometer: An On-Chip Reliability Monitor for Measuring Frequency Degradation of Digital Circuits," *IEEE Journal of Solid-State Circuits*, pp. 974-880, 2008.

[56] J. Keane, X. Wang, D. Persaud, and C.H. Kim, "An All-In-One Silicon Odometer for Separately Monitoring HCI, BTI, and TDDB," *IEEE Journal of Solid-State Circuits*, pp. 817-829, 2010.

[57] S. Mahapatra, D. Saha, D. Varghese, and P. B. Kumar, "On the Generation and Recovery of Interface Traps in MOSFETs Subjected to NBTI, FN, and HCI Stress," *IEEE Trans. on Electron Devices,* vol. 53, no. 7, pp. 1583-1592, 2006.

[58] *http://www.synopsys.com/Community/UniversityProgram/Pages/Library.aspx.*

[59] K. Uwasawa, T. Yamamoto, and T. Mogami, "A New Degradation Mode of Scaled P+ Polysilicon Gate P-MOSFETs Induced by Bias Temperature Instability," in *Proc. Int. Electron Devices Meeting*, pp. 871-874, 1995.

[60] P. Heremans, R. Bellens, G. Groeseneken, and H. E. Maes, "Consistent Model for the Hot Carrier Degradation in N-Channel and P-Channel MOSFETs," *IEEE Trans. Electron Devices*, vol. 35, no. 12, pp. 2194-2209, 1988.

[61] Y. Wang, S. Cotofana, and L. Fang "A unified Aging Model of NBTI and HCI Degradation towards Lifetime Reliability Management for Nanoscale MOSFET Circuits," *IEEE Int. Symposium on Nanoscale Architecture,* 2011.

[62] *http://www.synopsys.com/Community/UniversityProgram/Pages/Library.aspx.*

[63] *http://www.eetimes.com/design/memory-design/4376742/Anti-fuse-memory-provides-robust--secure-NVM-option*.

[64] *http://www.sidense.com/technology.html*.

[65] *http://www.kilopass.com/products/otp-memory-ip/xpm-otp-nvm/*.

[66] *http://www.nangate.com/?page_id=22*.

[67] *Synopsys, HSPICE user guide, 2010*.

[68] J. Lee, I. Park, and J. McCluskey "Error Sequency Analysis," *Proc. VLSI Test Symp.*, 2008

[69] INOVYS, *http://www.etesters.com/listing/40e8f648-a2d6-23b8-949b-4b3c005c86fb/Ocelot_ZFP_-_Test_System_for_Complex_SOCs*.

[70] http://www.poly.edu/csaw2011/csaw-embedded

Appendix

Publications Related to this Thesis

[1] **Xuehui Zhang**, Andrew Ferraiuolo, and Mohammad Tehranipoor, "Detection of Trojans using a Combined Ring Oscillator Network and Off-chip Transient-Power Analysis, ACM Journal on Emerging Technologies in Computing Systems (JETC), 2012

[2] **Xuehui Zhang** and Mohammad Tehranipoor, "Path-delay Fingerprinting for Identification of Recovered ICs, IEEE Int. Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012

[3] **Xuehui Zhang**, Kan Xiao, and Mohammad Tehranipoor, "Identification of Recovered ICs using Fingerprints from a Light-Weight on-chip Sensor, Design Automation Conference (DAC), 2012

[4] Andrew Ferraiuolo, **Xuehui Zhang**, and Mohammad Tehranipoor, "Experimental Analysis of a Ring Oscillator Network for Hardware Trojan Detection in a 90nm ASIC, IEEE/ACM Int. Conference on Computer-aided Design (ICCAD), 2012

[5] **Xuehui Zhang**, Nichlas Tuzzio, and Mohammad Tehranipoor, "Red Team: Design of Intelligent Trojans with Known Defense Schemes, International Conference on Computer Design (ICCD), 2011

[6] **Xuehui Zhang** and M. Tehranipoor, "RON: An On-chip Ring Oscillator Network for Hardware Trojan Detection," Design, Automation, and Test in Europe (DATE), 2011

[7] **Xuehui Zhang** and Mohammad TehranipoorM. Tehranipoor, "Case Study: Detecting Hardware Trojans in Third-Party Digital IP Cores, in Int. IEEE Hardware-Oriented Security and Trust (HOST), 2011

[8] M. Tehranipoor, H. Salmani, Xuehui Zhang, X. Wang, R. Karri, J. Rajendran, and K. Rosenfeld, "Hardware Trojan Detection: Solutions and Design Challenges, IEEE Computer Magazine, Dec. 2010

[9] **Patent**: Mohammad Tehranipoor, Xiaoxiao Wang, and **Xuehui Zhang**, Embedded Ring Oscillator Network for Integrated Circuit Security and Threat Detection, Pending