6-6-2016

Establishment of Trust and Integrity in Modern Supply Chain from Design to Resign

Ujjwal Guin
ujjwal.guin@gmail.com

Establishment of Trust and Integrity in Modern Supply Chain from Design to Resign

Ujjwal Guin, Ph.D.

University of Connecticut, 2016

With the advent of globalization and the resulting horizontal integration, present-day electronic component supply chain has become extremely complex and called for immediate solutions to eliminate counterfeit integrated circuits (ICs). Such counterfeit ICs have raised serious concerns regarding the safety and security of military systems, financial infrastructures, transportation, communication, household appliances, and many more applications. Various types of counterfeit ICs – recycled, remarked, overproduced, out-of-spec/defective, cloned, forged documentation, and tampered – have made the supply chain vulnerable to various attacks. However, due to the lack of efficient detection and avoidance techniques, many more instances of counterfeit ICs evade detection than those that are actually detected. Over the past few years, standards and programs have been put in place throughout the supply chain that outline testing, documenting, and reporting procedures for counterfeit IC detection. However, these test methods are mostly designed to detect only recycled and remarked ICs. Moreover, there is little uniformity in the test results among the various entities involved in the test process. Currently, there are no metrics for evaluating counterfeit detection methods. In addition, excessive cost and time to

implement these tests make the detection process even challenging. In this research, we have addressed the aforementioned issues by assessing existing test methods with newly developed test metrics, and developing different design-for-anti-counterfeit (DfAC) measures.

For the assessment of test methods, we have proposed taxonomies for (i) all different counterfeit IC types currently infiltrating the electronic component supply chain, (ii) defects present in different counterfeit ICs, and (iii) currently available test methods for detecting these ICs. Based on these taxonomies, we have introduced relevant and novel test metrics to evaluate the effectiveness of test methods. We have developed a comprehensive framework (i) for assessing a set of test methods to evaluate their effectiveness based on the newly developed metrics, (ii) selecting a set of test methods to maximize counterfeit defect coverage considering test cost and time budget, and (iii) deciding on the best set of test methods for achieving maximum counterfeit defect coverage (CDC).

Due to the sheer number of different component types (digital, analog, and mixed-signal) and sizes (large or small), it becomes extremely difficult to develop a one-size-fits-all DfAC measure to detect and prevent counterfeit ICs. Thus, we have proposed a suite of DfAC measures, which can help us to detect these counterfeit ICs without the need for conventional test methods. First, we propose a group of solutions for combating die and IC recycling (CDIR). These solutions include light-weight, on-chip structures based on ring oscillators (RO-CDIR), and semiconductor fuses (F-CDIR). Each structure meets the unique needs and limitations of different part types and sizes, providing excellent coverage for recycled ICs. Recycled digital ICs can

be effectively detected by using RO-CDIR. Any recycled ICs, specifically analog and mixed-signal ICs, can be identified by testing our F-CDIR with very low cost measurement devices, e.g., a multimeter. Second, we have proposed two improved versions of RO-CDIR as it is extremely challenging to detect a recycled IC that has been used for a very short period of time. These versions address the fact that process variations outpace the degradation caused by aging especially in lower technology nodes, making it harder to detect potential recycling by aging degradation. Simulation results demonstrate that these CDIRs can detect ICs used even for a few hours. Finally, we present FORTIS: a comprehensive solution for protecting semiconductor intellectual properties (IPs) and ICs by ensuring forward trust between all entities involved in the system-on-chip (SoC) design and fabrication process. FORTIS is designed to prevent IC overproduction; however, it can be used to prevent other counterfeit types (except recycled ones). FORTIS uses an existing logic encryption technique to obfuscate the netlist of a SoC or a third party IP and allows manufacturing tests before the activation of chips, a feature that is lacking in other competing techniques. In addition, we also propose to attach an IP digest to the IP header to prevent modification of an IP by the SoC designers. We have shown that our approach is resistant to various attacks with the cost of minimal area overhead.

Establishment of Trust and Integrity in Modern Supply Chain from Design to Resign

Ujjwal Guin

B.S., Indian Institute of Engineering and Technology, Shibpur, India, 2004

M.S., Temple University, PA, 2010

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2016

APPROVAL PAGE

Doctor of Philosophy Dissertation

Establishment of Trust and Integrity in Modern Supply Chain from Design to Resign

Presented by

Ujjwal Guin, M.S., B.E.

Major Advisor

Mark M. Tehranipoor

Associate Advisor

John Chandy

Associate Advisor

Domenic Forte

University of Connecticut

2016

Dedicated to my parents.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my adviser, Prof. Mark M. Tehranipoor for his support and guidance during my graduate study at the University of Connecticut. His support and encouragement paved the way for my successful dissertation.

I also would like to express my gratitude to Dr. Domenic Forte for his insightful guidance. My sincere thanks to Prof. John Chandy for joining my advisory committee and reading my dissertation and providing useful suggestions.

I would like to thank Daniel DiMase of Honeywell Inc. for providing valuable feedback on all taxonomies and the *Assessment Framework*.

My special thanks to all my labmates for the pleasant study and research experience. The debates and exchanges of knowledge enriched my experience.

In the end, I would like to thank parents and other family members for their love and support throughout my entire life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The ever increasing problem of counterfeiting and piracy are of great concern to government and industry because of – (i) the negative impact they can have on innovation, economic growth, and employment, (ii) the threat they pose to the welfare of consumers, (iii) the substantial resources that they channel into criminal networks, organized crime, and other groups that disrupt and corrupt society, and finally, (iv) the loss of business from the trade in counterfeits [4]. Based on a 2008 report by the International Chamber of Commerce, it was estimated that the cost of counterfeiting and piracy for G20 nations was as much as US$775 billion every year and will grow at an astonishing rate to $1.7 trillion in 2015 [5].

Counterfeit integrated circuits (ICs), which constitutes a significant part of counterfeit products, pose a significant threat to the government and industrial sectors of the economy because they undermine the security and reliability of critical systems and networks. They have a negative impact on corporate identity and reputation, and they can trigger massive revenue losses. Due to the widespread use of electronic components in our day-to-day lives - both directly and indirectly - counterfeit ICs also pose major threats to the health, safety, and security of the population at large. For example, the failure of a pacemaker due to a counterfeit component can potentially take someone's life. Similarly, the anti-lock braking system (ABS), which

is found in most cars today and is controlled by sensors and electronics, could possibly fail due to the use of a counterfeit IC. This not only causes reliability issues, it could potentially lead to life-threatening accidents. A pilot could lose control of an airplane, jeopardizing the lives of all on board. A rogue nation could even disable air defense systems with the help of counterfeit components.

In addition to the impact on public safety and security, counterfeit ICs could also cause significant damage to the economy. For example, semiconductor companies spend billions of dollars every year to develop technologies, manufacture products, and provide support for the products they create. In contrast, counterfeiters spend minimal money on developing technologies. Instead, counterfeiting practices allow private individuals to remake an existing product for their own benefit, which only hinders the research and development of new products. Also, as the counterfeiters do not take responsibility for their counterfeit components, the failure of these components damages the corporate reputation of the original component manufacturers (OCMs). In many cases, the OCM can even bear the financial responsibility and logistics of replacing the failed components.

A recent report from the Information Handling Services Inc. (Englewood, CO, USA) shows that reports of counterfeit parts have quadrupled since 2009 [6] (see Figure 1.1). This data has been compiled from two reporting entities - The Electronic Resellers Association International (ERAI) Inc. (Naples, FL, USA) and the Government-Industry Data Exchange Program, GIDEP (Corona, CA, USA). It is mentioned that the five most commonly counterfeited components (e.g., analog ICs, microprocessor ICs, memory ICs, programmable logic ICs, and transistors) represent $169 billion in potential annual risk for the global electronics supply chain

based on all reported counterfeit incidents in 2011 [3].

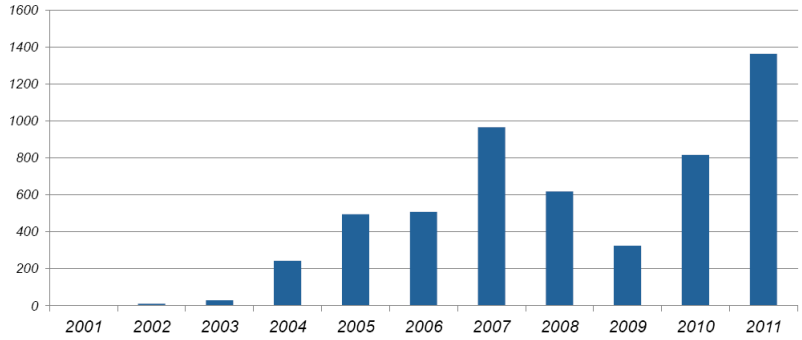


Fig. 1.1: Counterfeit incidents reported by IHS.

The rapid growth of IC counterfeiting is due to the widely available electronic waste (e-waste). In the United States, only 25% of e-waste was properly recycled in 2009 [7]. This huge resource of e-waste allows counterfeiters to pile up an extremely large supply of counterfeit components. Counterfeiters recycle electronic components from this e-waste and sell them in the open market as if they were new or even of a superior grade (for example, commercial grade components are sold as military or space grade components). In addition to that, as the complexity of electronic systems and their components have grown significantly over the past few decades, these components have been increasingly assembled (fabricated) globally to reduce production costs. For example, large foundries located in different countries can offer lower prices to the design houses. Untrusted foundries and assemblies can also be capable of selling extra components outside of the number they were contracted to manufacture. Thus, this complex supply chain leads to an illicit market willing to undercut competition with counterfeit parts.

With counterfeit incidents on the rise, it is increasingly important to understand what ICs

Table 1.1: Top-5 most counterfeited semiconductors in 2011 [3].

Rank	Commodity Type	% of Reported Incidents
#1	Analog IC	25.2%
#2	Microprocessor IC	13.4%
#3	Memory IC	13.1%
#4	Programmable Logic IC	8.3%
#5	Transistor	7.6%
#6	Others	32.4%

are most likely counterfeit and what industries are impacted the most. Table 1.1 shows the five most commonly counterfeited components that represent $169 billion in potential annual risk for the global electronics supply chain. The components are as follows: analog ICs, microprocessor ICs, memory ICs, programmable logic ICs, and transistors. Together, these five types of components make up around 68% (or, slightly more than two-thirds) of all the counterfeit incidents reported in 2011.

Table 1.2 shows the industries where these top five components are used. They include computing, consumer electronics, wireless and wired communications, automotive and industrial sectors. Automotive and industrial sectors involve critical systems and, thus, the appearance of unreliable counterfeit components in these applications is quite alarming. Untrustworthy counterfeit components are also a concern for consumer applications where we are increasing becoming more reliant on electronic devices for computing, communication, online banking, handling personal data, etc.

Table 1.2: Percentage of market revenue for most commonly counterfeited product types by application market in 2011 [3].

Part Type	Industrial	Automotive	Consumer	Wireless	Wired	Compute	Other
Analog IC	14%	17%	21%	29%	6%	14%	0%
Microprocessor IC	4%	1%	4%	2%	3%	85%	0%
Memory IC	3%	2%	13%	26%	2%	53%	1%
Prog. Logic IC	30%	3%	14%	18%	25%	11%	0%
Transistor	22%	12%	25%	8%	10%	22%	0%

1.1 Contributions

In this thesis, we have systematically addressed the problems of counterfeit ICs. First, we have analyzed the entire threat space for counterfeiting and developed taxonomies of counterfeit types, the defects present in these ICs, and test methods to detect these ICs. Second, we take a novel approach to assess the effectiveness of all the test methods for the detection of these counterfeit ICs and develop a comprehensive framework to select a minimum/best set of tests for maximizing test coverage. Finally, we propose different design-for-anti-counterfeit (DfAC) measures to detect these ICs without using conventional test methods. We propose combating die and IC recycling (CDIR) to detect recycled ICs. In addition, we present FORTIS to detect other counterfeit ICs. The main contributions of the proposed work are as follows:

1.1.1 Development of Taxonomies

We have developed a comprehensive taxonomy of counterfeit types. We have also developed a detailed taxonomy of the defects present in counterfeit ICs. To the best of our knowledge, this

is the first approach to analyzing counterfeit ICs with the defects present in them. Our counterfeit method taxonomy describes all the test methods currently available for the detection of counterfeit ICs. We have classified the counterfeit ICs into seven distinct categories: recycled, remarked, overproduced, out-of-spec/defective, cloned, forged documentation, and tampered [8] [9] [10] [11] [12]. Counterfeit defects are those anomalies and changes that are not typically found in authentic parts. Anomalies vary based on size, shape, type, number, etc., depending on the capabilities possessed by the counterfeiters. The detection of one or more anomalies may be an indication of a component being counterfeit. A taxonomy of the counterfeit defects was introduced in [13] [9] [14]. The detailed description of detects can be found in [12]. Counterfeit defects are divided into four categories: procedural, mechanical, environmental, and electrical. The test methods to detect counterfeit ICs are broadly classified into two distinct types – physical tests and electrical tests. Physical tests are performed to examine the physical and chemical/material properties of the component's package, leads and die of a component in order to detect procedural, mechanical, and environmental counterfeit defects. Electrical tests are the only way to determine the correct functionality of a component. They provide a very efficient and non-destructive way of detecting counterfeit components. Majority of the counterfeit defects can also be effectively detected by electrical tests. We will briefly describe these taxonomies in Sections 1.2, 1.4, and 1.5.

1.1.2 Development of Test Metrics

We have proposed test metrics for evaluating counterfeit detection methods. These metrics are counterfeit defect coverage (*CDC*), counterfeit type coverage (*CTC*), under-covered defects (*UCDs*) and not-covered defects (*NCDs*). We will introduce them in Chapter 2.

1.1.3 Effectiveness Assessment of Test Methods for the Detection of Counterfeit ICs

We have developed a comprehensive framework to assess the test methods [13] [14]. Different sequences of test methods have been developed by organizations for the detection of counterfeit parts. The framework evaluates the effectiveness of a sequence of test methods (test plan) used to screen for counterfeit parts. This framework works in two different modes. In the static assessment, it performs the assessment of a preexisting sequence of test methods. The output of this mode produces the test metrics (*CDC*, *CTC*, *NCD*, and *UCD*). In the dynamic assessment, the framework receives all the current available test methods as input and recommends (i) the best set of tests and (ii) an optimum set of tests that provides maximum coverage within a certain test time and cost budget. Then the assessment is done on the basis of the same test metrics. We will briefly describe the assessment process in Chapter 2.

1.1.4 Design-for-Anti-Counterfeit measures (DfAC) for the Detection and Avoidance of Counterfeit ICs

The efficiency of the physical and electrical test methods rely on finding the defects and anomalies present in the counterfeit ICs. However, these defects are not uniformly present in all the counterfeit types. In addition, these test methods are extremely expensive and slow. Moreover, counterfeiting has evolved over the last few years and the level of sophistication in the process has significantly improved. We believe that this trend will continue and that counterfeiters will continue to adopt new processes and technologies as time passes, making it more difficult to detect counterfeit ICs [12]. It may even be possible that we will not find any of today's gross defects and anomalies [12] in the counterfeit ICs of the future.

To address these shortcomings, we propose different lightweight structures for combating die and IC recycling. We call these structures as CDIRs [15] [16] [17]. These structures are of ring-oscillator (RO)-based and are similar in spirit to the CDIR proposed in [1]. However, our new structures are NBTI-aware and exploit aging much more efficiently. In addition, we present FORTIS [18], a comprehensive solution for establishing forward trust for detecting other counterfeit ICs. We will briefly describe these DfAC measures in Chapters 3, 4, and 5.

1.2 Counterfeit Types

The US Department of Commerce first proposed the definition for a counterfeit part. According to [21], a counterfeit component – (i) is an unauthorized copy; (ii) does not conform to original original component manufacturer (OCM) design, model, and/or performance standards; (iii) is not produced by the OCM or is produced by unauthorized contractors; (iv) is an off-specification, defective, or used OCM product sold as "new" or working; or (v) has incorrect or false markings and/or documentation.

The above definition does not include all possible scenarios where an entity in the component supply chain source electronic components that are authentic and certified by the OCMs. For example, one may copy the entire design of a component by reverse engineering [22] [23], manufacture them, and then sell them in the market under the OCM's identity. An untrusted foundry or assembly may source extra components without disclosing it to the OCMs [24] [25]. An adversary can insert a hardware Trojan [26] into a component to interrupt its normal operation and satisfy his/her own malevolent interests. All these scenarios impact the security and reliability of a system utilizing such components. Thus, we have expanded on the above defini-

tion of counterfeiting and developed a more comprehensive taxonomy of counterfeit types [13] [9] [11] [10] [8] [14]. Figure 1.2 shows this novel taxonomy of counterfeit types.

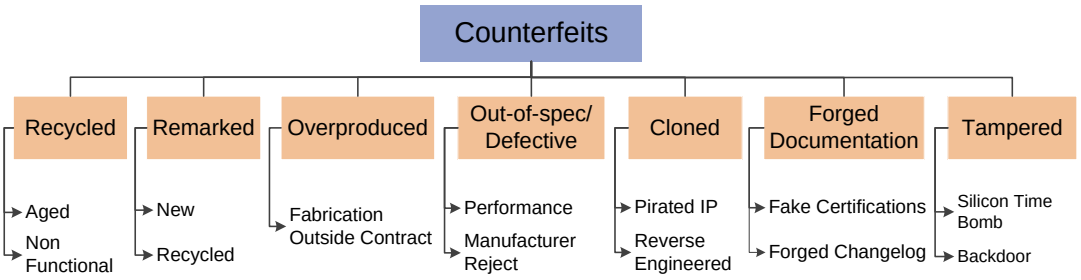


Fig. 1.2: Taxonomy of counterfeit types.

1.2.1 Recycled

The term "recycled" refers to an electronic component that is reclaimed or recovered from an used system, and is then modified to be misrepresented as a new component of an OCM. Recycled components may exhibit lower performance and have a shorter lifetime due to the aging from their prior usage. Further, the reclaiming process (removal under a very high temperature, aggressive physical removal from boards, washing, sanding, repackaging, etc.) could damage the part(s), introduce latent defects that pass initial testing but are prone to failure in later stages, or make them completely non-functional due to exposure to extreme conditions in an uncontrolled environment. Such parts will, of course, be unreliable and render the systems that unknowingly incorporate them equally unreliable.

The recycled components are discussed widely by the government, industry and test labs. The standards [27], [28], [29], and [30] recommends different test plan to detect these components. In this thesis, our aim is to highlight the most effective ways of detecting these components. In Chapter 2, we perform the assessment of existing test methods to find best set

of test methods to maximize test coverage with test time and test cost constraints. Chapters 3 and 4 exclusively describe DfAC measures to easily detect recycled components and therefore prevent them from getting into the component supply chain.

1.2.2 Remarked

The marking of an electronic component contains information such as part or identifying number (PIN), lot identification code or date code, device manufacturer's identification, country of manufacture, electrostatic discharge (ESD) sensitivity identifier, certification mark, and so forth. Clearly, a component's markings are very important as they represent component's origin and, most importantly, determine how the component should be handled and used. For example, a space grade component can withstand a wide range of temperatures, radiation levels, etc. that would cause instant failure for a commercial grade component. The component manufacturer, grade, etc. also determine how much the component is worth. The price of space and military grade components can be phenomenally higher than commercial grade components. A counterfeiter can drive up a component's price on the open market by changing its markings to that of a higher grade or better manufacturer. However, such remarked components will not be able to withstand the harsh conditions of their more durable, higher-grade counterparts. This can create substantial issues if such components end up in critical systems.

Similar to the recycled counterfeit type, remarked components are also extensively discussed by the government, industry and test labs. The standards developed thus far recommended the same test plan to detect remarked components. In this thesis, we address the detection of recycled and remarked components simultaneously.

1.2.3 Overproduced

Today's high-density ICs are mostly manufactured in state-of-the-art fabrication facilities. Building or maintaining such facilities for modern CMOS technologies is reported to cost more than several billions of dollars and this number is growing with each new technology node [31]. Given this increasing cost and the complexity of foundries and their processes, the semiconductor business has largely shifted to a contract foundry business model (horizontal business model) over the past two decades. In this model, the design houses outsource their designs for fabrication and packaging to companies all around the world, mainly to reduce manufacturing costs. Although the contracted parties may agree to only manufacture a certain number of working components, they could in fact exceed this amount. Untrusted foundries and assemblies may produce more than the number of components they are contracted to produce. In addition, they can overbuild components by hiding the actual yield (i.e. the percentage of defect-free components to the total number of components) information.

This process of manufacturing and selling outside of the agreement with the design house (i.e., the components' intellectual property (IP) owner) is known as "overproduction". A well understood concern with overproduction is the inevitable loss in profit for the design houses. Design companies usually invest a large amount of time and effort into research and development (R&D) of their products. When an untrusted foundry or assembly overproduces and sells these components, the design house loses any possible revenue that could have been gained from those components. However, an even bigger concern with overproduced components is that of reliability. Overproduced components may simply end up in the market with minimal or no testing for reliability and functionality. These components may find their way back into the

supply chain for many critical applications such as military equipment and consumer products, which raises concern for safety and reliability. Further, since these components bear the same name of the design houses, failure of these components would then tarnish the reputation of the original component manufacturer. In Chapter 5, we will briefly discuss overproduction and will present FORTIS to prevent IC overproduction.

1.2.4 Out-of-Spec/Defective

A part is considered defective if it produces an incorrect response in post-manufacturing tests. During the manufacturing process, the first test performed is the wafer test to inspect which ICs, fabricated on the wafer, are defective. If there are too many defective ICs on the wafer, the foundry sometimes rejects the whole wafer. A wafer generally contains hundreds of dies depending on the size and type of ICs and may worth hundreds of dollars. An untrusted entity may source these defective wafers to an assembly and produce defective or out-of-specification ICs. After wafer tests, the defect free dies are sent to assembly for packaging. The healthy chips are then sorted out by using package tests and the chips that have been damaged during the packaging process are discarded. An untrusted entity again can supply these chips into the supply chain. The final test is performed as a part of quality assurance of the final packaged chips before sending them to the market. Burn-in, using accelerated temperature and voltage, is often performed to test latent defects in order to avoid the failures in the early operational stages of chips.

All the rejected chips from various test process should be destroyed (if they are non-functional), downgraded (does not satisfy the specification), or otherwise be properly disposed of. However, if they are sold on the open market instead, either knowingly by an untrusted

entity or by a third party who has stolen them, there will be an inevitable increase in their risk of failure. The detection of these defective/out-of-spec components is not an easy task. It may be easy to detect a defective chip that has been rejected in the early test process by using simple parametric tests. However, it will be extremely difficult if the chips are rejected in the later phase of the test process. Rather than depending on the detection of these ICs, we can add DfAC measures such that they cannot enter into the supply chain for the first place. In Chapter 5, we will present FORTIS to prevent getting these ICs into the supply chain.

1.2.5 Cloned

Cloning is widely used by a range of adversaries/counterfeiters (from small entities to large organizations) to copy a design in order to eliminate the large research and development (R&D) costs of a part. Cloning is a major concern for semiconductor intellectual properties (IPs), such as layouts, netlists and HDL design blocks as well as fabricated integrated circuits. Cloning can be done by reverse engineering or by illegally obtaining semiconductor intellectual property (IP) such as layouts, netlists etc. (also called IP theft). Cloning can also occur by unauthorized knowledge transfer from a person with access to the part's design. Such cloned components violate intellectual property rights of the rightful IP owners and could cause them significant losses in revenue. In Chapter 5, we will briefly discuss IP piracy and provide solutions to prevent cloning.

1.2.6 Forged Documentation

The documentation shipped with any component contains information regarding its specifications, testing, certificates of conformance (CoC) and statement of work (SoW). By modifying

or forging these documents, a component can be misrepresented and sold even if it is nonconforming or defective. It is often difficult to verify the authenticity of such documents because the archived information for older designs and older parts may not be available at the OCM. Legitimate documentation can also be copied and associated with parts from a lot that do not correspond with the legitimate documentation. The incentive for counterfeiters and risks associated with parts with forged documentation are similar to those discussed above for remarking.

1.2.7 Tampered

The vulnerabilities of ICs to malicious alteration has become predominant due to the globalization of the semiconductor supply chain. ICs that have been tampered with can have dangerous consequences to military infrastructures, aerospace systems, medical, financial, and transportation infrastructures, and commercial infrastructures. An adversary can insert a hardware Trojan [26] in a design to interrupt its normal operation and/or disable it in the future, effectively making it a "silicon time bomb". A hardware Trojan may also create a backdoor that gives access to critical system functionality or leaks secret information to an adversary. Hardware Trojans can be implemented by modifying (i) the hardware in application-specific integrated circuits (ASICs), digital signal processors (DSPs), microprocessors, microcontrollers, or (ii) the bitstream for field programmable gate arrays (FPGAs). A hardware Trojan can modify the functionality of a design in a variety of ways. For example, a hardware Trojan can disable the crypto module on a design and leak unencrypted plain text which can easily be intercepted, or disable the system clock of a module for a small duration to launch a sabotage. A detailed taxonomy for hardware Trojan can be found in [32].

Since the detection and avoidance of tampering is a large problem unto itself, we shall

consider it beyond the scope of this thesis. Interested readers are suggested to read *Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection* [26] for further information on hardware Trojan insertion, detection, and prevention.

1.3 Supply Chain Vulnerability

Typically an electronic component will go through a process as shown in Figure 1.3. This process includes design, fabrication, assembly, distribution, usage in the system, and finally end of life. The vulnerabilities associated with each step are discussed in more detail below.

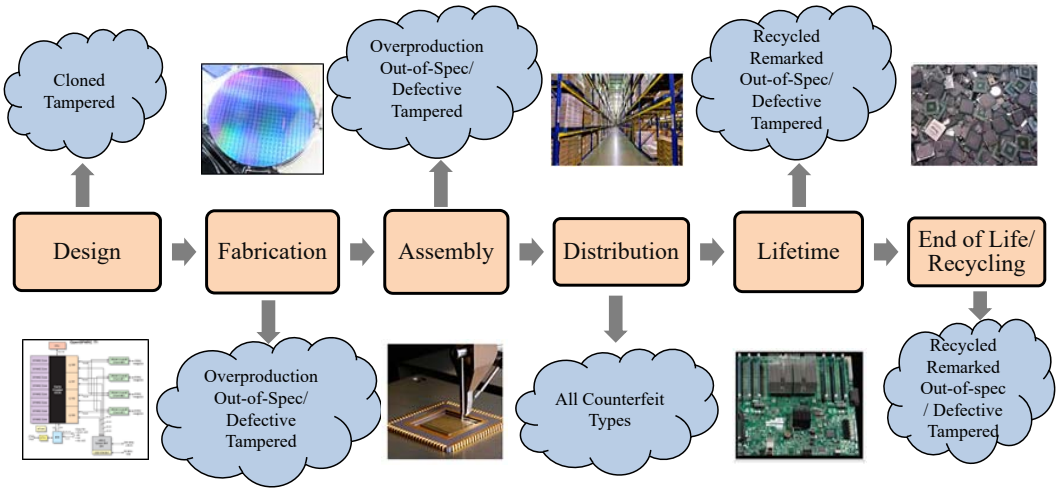


Fig. 1.3: Electronic components supply chain vulnerabilities.

1.3.1 Design

The design implementation of large complex integrated circuits has evolved to a stage where it is extremely challenging to complete the entire design in-house. The flow from RTL to GDSII is performed in many different places (even in different countries) mainly to reduce the devel-

opment cost and design-to-market time. Design reuse has also become an integral part of SoC design. Hard IPs (layout level designs), firm IPs (designers can optimize codes with parameterized constraints), and soft IPs (synthesizable register-transfer level (RTL) designs) can be used for this purpose. Attacks on the design stage can be performed in the two following ways: (i) the counterfeiter can steal these IPs to create cloned components, (ii) the counterfeiter can tamper with codes to modify the functionality, create backdoors, etc.

1.3.2 Fabrication

Today's integrated circuits are manufactured in fabrication facilities (fabs) located all around the world primarily to reduce the manufacturing cost. The design house contracts a foundry to fabricate their designs, discloses the details of their IPs, and also pays for mask-building costs based on their designs. The contract agreement between the foundry and design house is protected by IP rights [33]. However, this contract foundry business model (namely horizontal business model) creates a trust issue between the design house and foundry. The design house must trust foundry not to overproduce ICs or pirate IPs. An untrusted foundry can potentially (i) make extra/overproduced ICs, by hiding their yield, and selling those extra ICs in the open market, (ii) tamper the design, and (iii) source defective and out-of-specification wafers to packaging companies to make finished parts.

1.3.3 Assembly

After fabrication, the foundry send tested wafers to assembly to cut the wafers into dies, package the dies, and perform final tests before being shipped to the market. An untrusted assembly can (i) build overproduced ICs by hiding the yield information, (ii) sell the defective/out-of-

specification ICs, and (*iii*) remark, forge, or upgrade a component's marking.

1.3.4 Distribution

The tested ICs are sent either to the distributors or system integrators. The distributors sell those ICs in the market. There are two types of distributors – authorized and unauthorized – existing in the supply chain. The threat lies mostly from unauthorized distributors, where the identity of the distributors are obscure. There are several reports pointing to phony distributors potentially sourcing all seven types of counterfeit components in the supply chain.

1.3.5 System Integration/Lifetime

System integration is the process of assembling together all the components and subsystems into one complete system. An untrusted system integrator can potentially use all types of counterfeit components in their system. They can maximize the profit by using the cheap or tampered counterfeit components.

1.3.6 End of life/Resign

When electronics age or become outdated, they are typically retired/resigned and subsequently replaced. Proper disposal techniques are highly advised to extract precious metals and to prevent hazardous materials (lead, chromium, mercury, etc.) from harming the environment [34]. Yet, these techniques are largely ignored, resulting in a large amount of electronic waste or e-waste. For instance, in the United States, only 25% of electronic waste was properly recycled in 2009 [7]. That percentage might be lower for many other countries. A profitable business has grown out of reclaiming used components from this e-waste, remarking them, and then,

re-inserting them into the supply chain as new components. According to current reports, these recycled and remarked components account for over 80% of the reported counterfeit parts in the supply chain [35] and represent a growing threat [36]. Also, in this stage the counterfeiter can potentially tamper used components for sabotage or malfunction.

1.4 Counterfeit Defects

Counterfeit defects are those anomalies and changes that are not typically found in authentic parts. A counterfeit part may often contains one or more different anomalies, deviation from normal and usual form and/or functionality of a genuine component. These anomalies may be on the leads/package, degradation in its performance, or a change in its specifications. Since we assume that the assemblies comprehensively test their components, we should not expect any defects in genuine parts. Any anomalies or defective behaviors in a part must, therefore, be attributed to its being counterfeit.

To determine the effectiveness of a test plan, which consists of a sequence of tests (see Section 1.5), it is necessary to determine how many different defects are detected by it. Not same defects are simultaneously visible in all counterfeit components. Different defects are present in this wide variety of counterfeit components. It would be better to detect more defects to achieve higher confidence of detecting a component as counterfeit. Thus there is an urgent need to find out all possible defects present in different counterfeit components.

Figure 1.4 shows a detailed taxonomy of the counterfeit defects, which was introduced in [12] [13] [9] [14]. In this section, we present a comprehensive taxonomy of defects, which is divided into four categories: procedural, mechanical, environmental, and electrical. The

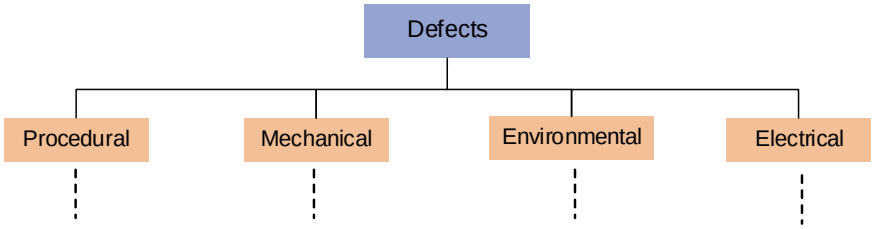


Fig. 1.4: A taxonomy of the defects and anomalies present in counterfeit components.

detailed descriptions of these defects can be found in our book *Counterfeit Integrated Circuits: Detection and Avoidance [12].*

1.4.1 Procedural Defects

The components are traveled in the supply chain with proper protection against shipping, handling, and environments. Any damage or without that protection may cause the components fail during operation, and they must not be accepted as reliable components. Again, the customers should receive documents verifying the authenticity of the components they purchased based on purchasing requirements. If there is a mismatch exist between the documents received compared to the original, then this would be flagged for further testing. These procedural defects are related to the packaging and shipping of components and the markings of the component itself. Figure 1.5 shows a taxonomy of procedural defects.

1.4.2 Mechanical Defects

Figure 1.6 shows a detailed taxonomy of mechanical defects. These defects are directly related to components' physical properties. For example, leads on an IC can show how the part has been handled if it was previously used. Physically, leads should adhere to datasheet specifica-

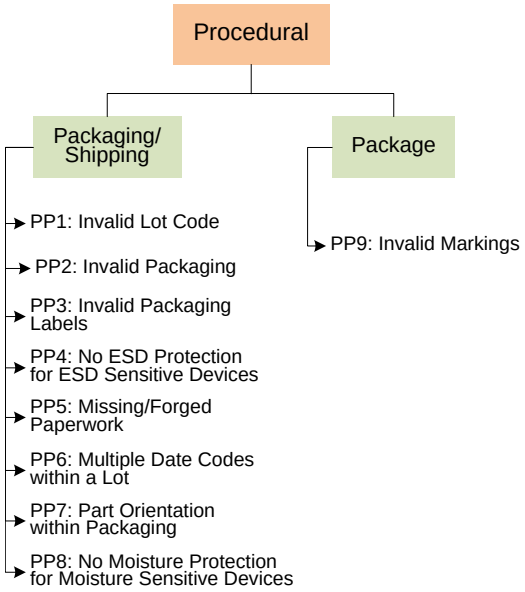


Fig. 1.5: Procedural defects.

tions, including straightness, pitch, separation, etc. The leads' final coating should be consistent throughout the entire lot, as well. Leads should also have a consistent elemental construction. The package of an IC provide significant information. For example, this is the location where all model numbers, country of origin, date codes, and other information are etched. If the package exhibits any sanding or grinding marks externally, it has likely been remarked. Further inspection for the blacktop coating should be done to determine whether this is the case. Ghost markings, color variations, improper textures, and extraneous markings on the package clearly indicate that a part has been reused.

1.4.3 Environmental Defects

Environmental defects are caused when the environmental parameters interact with the outer structure of a component. Oxidation and corrosion on leads are caused when a part is kept a

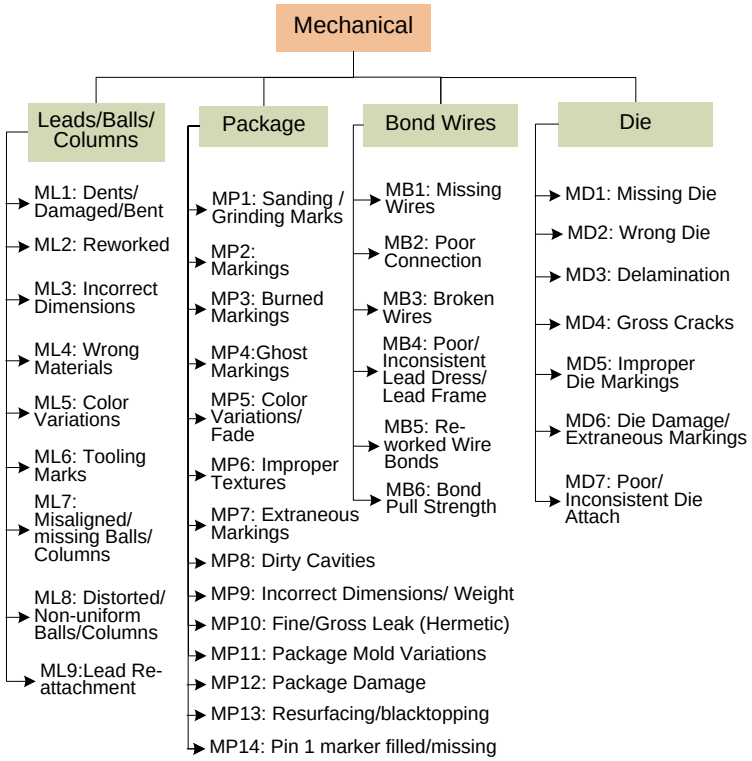


Fig. 1.6: Mechanical defects.

long time without proper protection. Again, during the recycling process, the leads can easily get oxidized at higher temperatures and contaminated by other materials. Figure 1.7 shows a taxonomy of environmental defects.

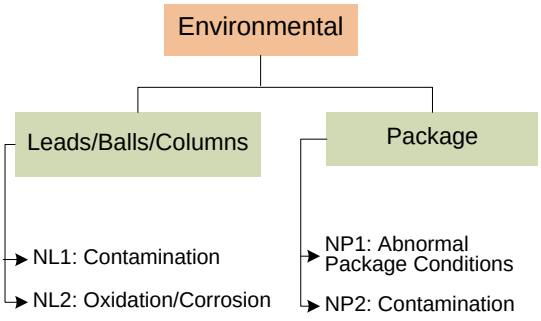


Fig. 1.7: Environmental defects.

1.4.4 Electrical Defects

Typical electrical defects can be classified into two distinct categories: These are parametric defects and manufacturing defects, both of which are shown in Figure 1.4. Parametric defects are the manifestation of the shift of component parameters due to prior usage or temperature . A shift in circuit parameters due to aging will occur when a chip is used in the field for some time. Aging of a chip used in the field can be attributed to four distinct phenomena, which are becoming more prevalent as feature size shrinks. The most dominant phenomena are negative bias temperature instability (NBTI) [37] [38] [39] [40] [41] and hot carrier injection (HCI) [42] [43] [44] [41] which is prominent in PMOS and NMOS devices, respectively. NBTI occurs in p-channel MOS devices stressed with negative gate voltages and elevated temperature due to the generation of interface traps at the Si/SiO_2 interface. Removal of the stress can anneal some of the interface traps, but not completely. As a result, it manifests as the increase of threshold voltage (V_{th}) and absolute off current (I_{off}) and the decrease of absolute drain current (I_{DSat}) and transconductance (g_m). HCI occurs in NMOS devices caused by the trapped interface charge at Si/SiO_2 surface near the drain end during switching. It results in non recoverable V_{th} degradation. These effects also lead to out-of-spec leakage current and out-of-spec transient current. Delay defects are also the direct effect of all the parametric variations mentioned above. Figure 1.8 shows a detailed taxonomy of electrical defects.

1.5 Test Methods

As the incidents of counterfeit components in the electronic component supply chain are on the rise, it has become necessary for manufacturers, distributors, and users of electronic com-

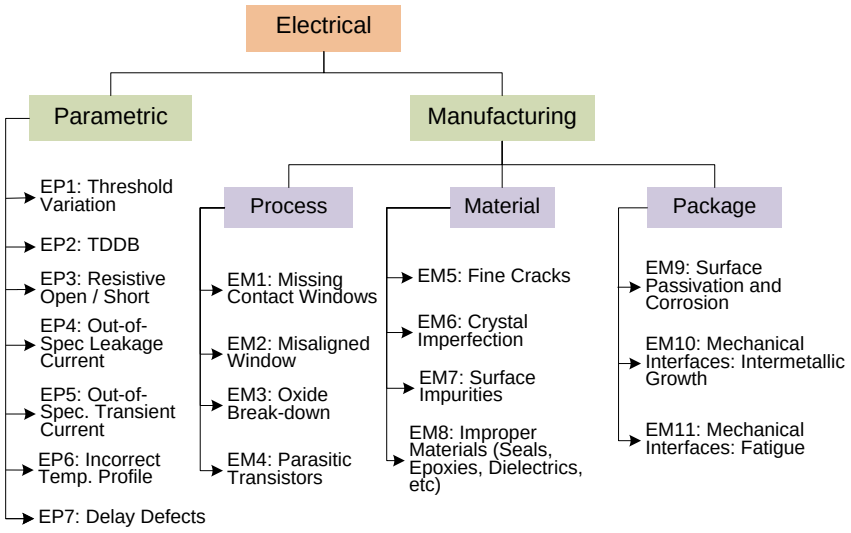


Fig. 1.8: Electrical defects.

ponents to inspect all incoming electronic components for authenticity. All components must go through a detailed acceptance test before being used in the system to ensure that they meet the quality and reliability requirements and that they are authentic, especially when they are used in critical infrastructures such as aerospace, medical, transportation, etc. We introduce a detailed taxonomy of counterfeit detection methods in [9] [11] [8] [13] [14]. Figure 1.9 shows a taxonomy of counterfeit detection methods. They are broadly classified into two distinct types – physical tests and electrical tests. The detailed descriptions of all these physical and electrical tests can be found in [12].

1.5.1 Physical Tests

Physical tests are performed to examine the physical and chemical/material properties of the component's package, leads and die of a component in order to detect procedural, mechanical, and environmental counterfeit defects (Section 1.4). When an order is received, it first goes

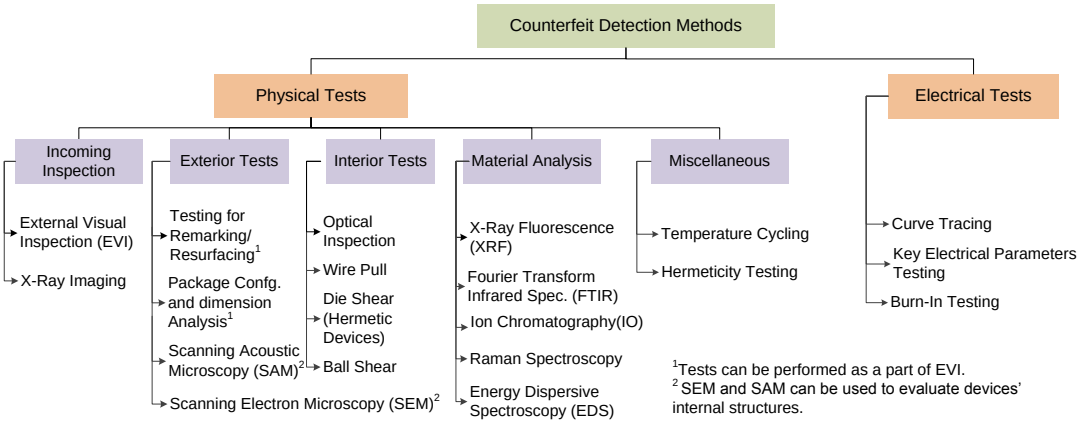


Fig. 1.9: A taxonomy of test methods for the detection of counterfeit components.

through an incoming inspection. All the components under test are inspected thoroughly. The exterior part of the package and leads of the component are analyzed by using exterior tests. The physical dimensions of the components are measured either by hand-held or automated test equipment. Any abnormal deviation of measurement from the specification sheet indicates that it may be a counterfeit component.

The internal structures, die and bond wires of the components are inspected by delid/ decapsulation. There are three mainstream methods commercially available for decapsulation. These are chemical, mechanical or laser-based solutions. Chemical decapsulation involves etching away the package with an acid solution. Newer laser-based techniques can remove an area of the package. Mechanical decapsulation involves grinding the part until the die is exposed. Once the part has been decapsulated and the required structures exposed, the interior tests need to be performed. The chemical composition of the component is verified using material analysis. This is the only category of tests that can detect defects and anomalies related to materials. Defects such as wrong materials, contamination, oxidation of leads and packages, etc., can be

detected. There are several tests that can perform material analysis including X-Ray fluorescence, energy dispersive spectroscopy, etc.

1.5.2 Electrical Tests

The correct functionality of a component is efficiently verified by electrical tests. The non-destructive nature of these tests, they can be very efficient of detecting counterfeit components. Electrical tests are the only way to determine the correct functionality of a component. These tests are a very efficient and non-destructive way of detecting counterfeit components. Majority of the counterfeit defects (see Section 1.4) can effectively be detected by electrical tests. Currently, there are curve tracing, key electrical parameters testing, and burn-in tests are widely used for counterfeit IC detection.

Curve tracing is gaining popularity in the detection of counterfeit components as it tests ICs non-destructively and without requiring extensive details of the ICs under test. It is not necessary to require a golden IC during authentication. In a typical curve tracer, standard voltage or current curves can be generated for any combination of pins of the ICs. These traces are formed by sweeping voltage V over a specified range and plotting the current I. The traces follow the Ohm's Law of $V = I * Z$, where Z is the impedance between the pins of an IC.

Testing of the key parameters, along with functional testing for evaluating the parameters, is the most effective way of verifying the functionality of a component. These tests, which are usually conducted at room temperature ($25^{\circ}C$) or even higher temperatures, are generally used to test components on the manufacturing floor of assemblies in order to increase confidence that the packaged ICs are free from defects and anomalies. These tests can be useful in detecting counterfeit components, especially those re-marked to a higher grade part. A counterfeit com-

ponent may fail under these tests if any defects and anomalies are present within it. By checking the correct functionality of a component, a glut of gross defects related to leads/balls/columns, bond wires, and die related defects can easily be detected. However, these tests are perhaps the most expensive test methods available for the detection of counterfeit components when performed on complex devices. For the functional tests, a series of algorithms that exercise and test specific elements of the design are needed which requires an expensive test setup and the development of complex test programs.

The reliability of a device is mainly ensured by burn-in tests [45]. In burn-in tests, the device is operated at stressed conditions to accentuate infant mortality and other unexpected failures. Such failures are often due to the latent defects, which do not necessarily expose themselves and may be skipped during manufacturing tests. Due to the electrical and thermal stresses during the usage in the field, these defects eventually expose themselves and, consequently, the devices shall fail to produce the correct functionality. During burn-in tests, the devices are operated at elevated levels of electrical (higher supply voltage) and thermal (higher temperature) stresses which accelerates the device's degradation. As a result, months to years of life time of the device are consumed in hours, allowing one to detect the presence of latent defects. Thus, by performing such tests, one can assure the reliability of a device over time as well as harsh conditions.

1.5.3 Challenges and Limitations of the Test Methods

The counterfeiting of ICs is a multifaceted and evolving problem. Counterfeiters are enriching their knowledge and technology as they are getting mature in this illegal business. The test methods that are capable of detecting counterfeit ICs today, may not be efficient in the near

future. Thus, it is important to analyze the limitations and challenges to implement these tests. The physical and electrical test methods described above provide some unique challenges for the detection of counterfeit ICs.

Physical Tests

Physical Tests suffers from several limitations and implementation challenges. In the following, we will describe them in details.

- *Counterfeit Types*: The physical inspections are primarily designed to detect recycled and remarked ICs. These tests are not applicable for authenticating any other counterfeit (overproduced, cloned, and out-of-spec/defective) types.

- *Dynamic Nature*: The dynamic nature of counterfeiting makes the detection even more challenging. Counterfeiters are evolving and adapting to new ways of making more deceptive counterfeit product. Currently, detection is mostly based on inspecting the physical appearances of devices. It is hardly a matter of time before these test methods will be ineffective in the near future.

- *Sampling*: Most of the physical tests are destructive. Sample preparation is extremely important as it directly relates to test confidence. If a few counterfeit components are mixed with a large batch, the probability of selecting the counterfeit ones for test is extremely small.

- *Test Time and Cost*: The test time and cost are major limiting factors in the use of physical tests for counterfeit detection. The equipment used for physical inspections (e.g., SEM

and SAM) are not custom-designed to detect counterfeit parts. It takes several hours to test a single component in detail.

- *Automation*: These tests are done in an ad-hoc fashion with no metrics for quantifying against a set of counterfeit types, anomalies, and defects. Most of the tests are carried out without automation.

- *Metrics*: Currently, there are no metrics to evaluate the effectiveness of physical inspections. The test results mostly depend on the subject matter experts (SMEs). The decision-making process is entirely dependent on the operator (or SMEs) – this is indeed error prone. A chip could be considered counterfeit in one lab while it could be marked as authentic in another lab. This was proven by a test run by Honeywell, where some labs reported a chip as counterfeit and others labeled it authentic [46].

Electrical Tests

Electrical tests have the potential to be an efficient means of counterfeit detection, as they do not have the same limitations of physical inspections. However, there are major challenges that are unique to electrical tests and they are as follows:

- *Process Variation:* Due to increased process variations and environmental variations (temperature, noise, aging, etc.), the electrical parameters of a component vary significantly. It will be very difficult to conclude whether the variations in the parameters of a component are due to the aging (for recycled and remarked components) or to the process variations in the circuit. One can perform a statistical analysis based on the data observed from the parametric tests to determine the confidence level that a part is counterfeit with

or without a golden IC. The efficiency of such analysis must be proven on a large number of golden and counterfeit parts.

- *Test Time and Cost:* Burn-in tests are useful in detecting infant mortality failures of components [12]. However, because of excessive test time (tens of hours) and cost, these tests are only attractive and useful only for critical and high-risk applications.

- *Function Verification:* Test program generation for obsolete and active parts with limited knowledge of the part will be extremely difficult, if not impossible. The requirement of having a high-speed tester in order to apply functional test patterns to chips make it extremely expensive. It is nearly impossible to get the complete set of test vectors for an obsolete part from the OCM. In some cases, the OCM may no longer exist or the information required may no longer be available in archived records at the OCM.

- *Counterfeit Types:* These tests not designed for the detection of overproduced ICs. The cloned ICs cannot be detected by these tests if the device operates within the specification.

1.6 Design-for-Anti-Counterfeit Measures

The detection of counterfeit ICs poses a significant challenge to securing global electronic component supply chain due to the lack of efficient, robust, and low-cost detection and avoidance technologies. While there are electrical and physical tests described above (Section 1.5) to identify counterfeit ICs, these approaches are usually suffered by several challenges and limitations (Section 1.5.3). In this section, we discuss alternative approaches that can be integrated into new components to detect and prevent different counterfeit ICs. These approaches are a part of

the design methodology to detect counterfeit components, which we term as *Design-for-Anti-Counterfeit (DfAC)* measures.

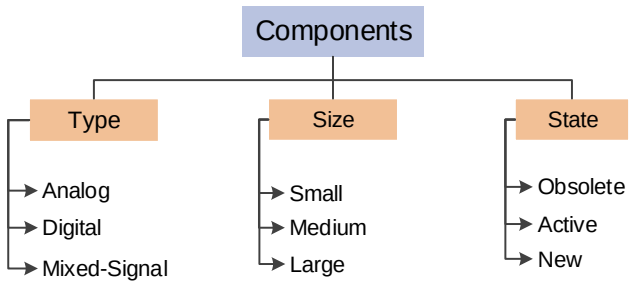


Fig. 1.10: Taxonomy of component types.

Different types of components (shown in Figure 1.10) can significantly impact the implementation of DfAC measures. Components can be classified by their type, size and state. The descriptions for the type and size are self-explanatory. We categorize state into three distinct types – obsolete, active, and new. Obsolete refers to components which are no longer manufactured by original component manufacturers (OCM) as they may switch to newer designs to improve performance, reliability, and/or manufacturing cost. These components are only be available through OCM authorized or independent distributors of electronic components. Active components are still being manufactured by OCMs, but their designs cannot be changed because of – (i) the extra cost of developing new masks and (ii) performance and reliability concerns. New components are very flexible in implementing avoidance measures as they are still in the design phase where the OCM can – (i) validate the performance and reliability parameters and (ii) modify masks.

Figure 1.11 shows the taxonomy of currently available DfAC measures. It can be broadly

classified into two categories – die level and package level.

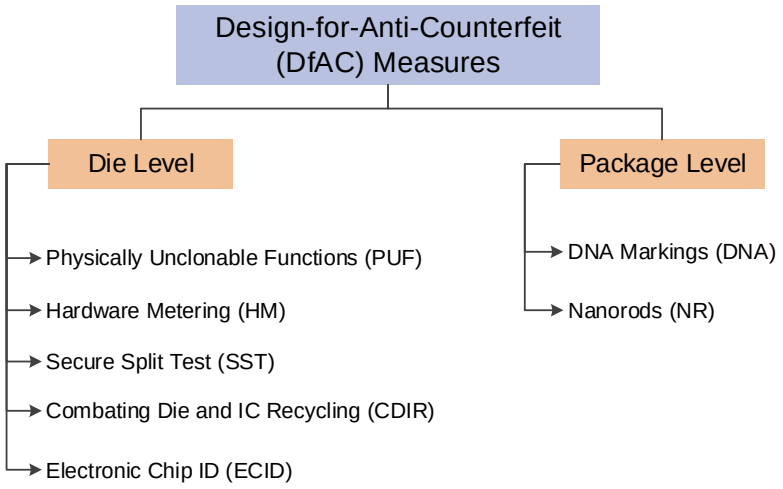


Fig. 1.11: A taxonomy of DfAC measures.

1.6.1 Die Level

Techniques to design DfAC measures are based on extracting unique features and parameters from a circuit to help uniquely identify each chip. These solutions needs to be embedded into the die during the manufacturing of ICs. These die level technologies are described below:

Physically Unclonable Functions

PUFs have received much attention from the hardware security and cryptography communities as a new approach for IC identification, authentication, and on-chip key generation [47] [48] [49] [50] [51]. Silicon PUFs exploit inherent physical variations (process variations) that exist in modern integrated circuits. These variations are uncontrollable and unpredictable, making PUFs suitable for IC identification and authentication [52] [53]. These variations can help generate

a unique signature for each IC in a challenge-response form, which allows later identification of genuine ICs. In recent years, various PUF architectures have been proposed. They are the arbiter PUF [54] [55], the ring oscillator PUF [48] [54], the SRAM PUF [56] [57], memristor PUF [58]. etc.

PUFs can be used to detect cloned ICs as they generate unique IDs resulting from randomness in the IC manufacturing process that cannot be controlled or cloned. These unique IDs of the genuine ICs can be stored in a secured database for future comparison. Overproduced ICs can also be detected, by searching the chip IDs under authentication in these secured databases. If no match is found, there is a high probability that the IC is not registered and is a member of an overproduced type.

Hardware Metering

Hardware metering is a set of security protocols that enables the design house to achieve post-fabrication control of the produced ICs. The design house can distinguish different ICs produced with the same masks, as hardware metering provides a unique way to tag each chip and/or its functionality [59] [24]. Hardware metering approaches can be either passive or active. Passive approaches uniquely identify each IC and register the IC using challenge-response pairs. Later, suspect ICs taken from the market are checked for proper registration [48] [50] [59] [60] [61] [62]. Active metering approaches, however, lock each IC until it is unlocked by the IP holder [53] [63] [64] [65] [66] [67]. This locking is done in a variety of ways, including: *(i)* initializing ICs to a locked state on power-up [53], *(ii)* combinational locking by, for instance, scattering XOR gates randomly throughout the design [65–67], and *(iii)* adding a finite-state machine (FSM) which is initially locked and can be unlocked only with the correct sequence of primary

inputs [64] [68].

Secure Split-Test

Due to the globalization of the semiconductor industry and the prohibitively high cost of creating foundries and assembly companies for packaging, test, and burn-in processes, foundries now often fabricate the wafers/dies, test them, and ship them to the assembly. The assembly then packages the dies, tests them, and ships the ICs to the market. The foundry/assembly, however, can ship defective, out-of-spec, or even overproduced chips to the black market, as described in Section 1.2. Secure Split-Test (SST) secures the manufacturing test process to prevent counterfeits, allowing intellectual property (IP) owners to protect and meter their IPs [25] [69] [70]. SST introduces hardware components for cryptography and to block the correct functionality of an IC until it is activated by the IP owner. In SST, IP owner/trusted party decides whether a chip is functionally correct or not. Besides, SST is designed to be resilient against different types of attacks to prevent the IC from being activated without IP owner's key. SST introduces the IP owner back into the manufacturing test process. SST is designed to prevent different types of counterfeited ICs such as cloned, overproduced, defective/out-of-spec ICs.

Combating Die and IC Recycling (CDIR)

The first CDIR to prevent parts from being recycled has been presented in [1] [71]. The technique in [1] inserts a light-weight sensor in the chip to capture the usage of the chip in the field and provides an easy detection capability. This type of sensor relies on the aging effects of MOSFETs to change a ring oscillator frequency in comparison with the golden one embedded in the chip. As a part used in the field ages because of the wearout mechanisms such as NBTI

and HCI, the shift in the frequency of this sensor indicates the level of aging and provides a simple readout of the value.

The antifuse-based CDIR sensor first proposed for recycled IC detection appeared in [72]. It is composed of counters and an embedded antifuse memory block. The counters are used to record the usage time of ICs while its value is continuously stored in an antifuse memory block. Since the antifuse memory block is one time programmable, counterfeiters can not erase the context during the recycling process. Two different structures of the AF-based sensor have been proposed to measure the usage time of ICs. *CAF-based sensor* records the cycle count of the system clock during 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. On the other hand, *SAF-based sensor* uses circuit activity as trigger (clock) to the counter. A number of signals with low switching probability is selected to calculate the usage time. It generally requires less area overhead than the CAF-based sensor.

Electronic Chip ID (ECID)

To track ICs throughout the supply chain, each IC needs to be tagged with a unique ID. This electronic chip ID (ECID) can be easily read during the chip's lifetime. The conventional approach for writing the unique ID into a non-programmable memory (such as One-Time-Programmable [OTP], ROM, etc.) requires post-fabrication external programming, such as laser fuses [73] or electrical fuses (eFuses) [74]. The eFuse is gaining popularity over the laser fuse because of its small area and scalability [74]. ECID can only be useful to detect remarked ICs only and can be subjected to cloning.

1.6.2 Package Level

The DfAC measures discussed so far only target new ICs. However, a large portion of the supply chain is populated by active and obsolete components (see Figure 1.10). There is no opportunity for adding any extra hardware to create DfAC measures in those designs. For tagging such active and obsolete components, we need to create DfAC measures on the package level such that one does not require the access to designs. No package modifications should be allowed during the implementation of such measures. DNA markings, and Nanorods are three viable options for creating such measures.

DNA Markings

Plant DNA is scrambled to create new and unique genetic sequences, and these new sequences integrated with inks. These inks are then applied on the packages of the ICs at the end of the packaging process. Authentication includes first checking whether the ink fluoresces under specific light, and second sending a sample of the ink to a lab to verify that the DNA is in the database of valid sequences [75]. Recently, DoD mandated [76] the DNA marking be placed on the components in order to track them throughout the supply chain. DNA markings have several limitations that introduce some serious concerns of their applicability in counterfeit avoidance. The fast authentication achieved by observing the fluorescence of the marking under specific light can be imitated by counterfeiters, either by invalid DNA or by other materials. But detailed DNA validation is extremely time-consuming and costly [77].

Nanorods

IBM researchers introduced gold nanorods on a surface using a simple printing process [78]. In this technique, a microscopic pattern is created by growing an array of nanospheres into nanorods that are less than 100nm long. Each time the process is repeated, the same pattern is created, but the exact angle and length of each individual nanorod varies, so that each set of nanorods is distinct. After the array of nanorods is grown, it is applied to a chip using a specialized printer. A chip with gold nanrods on its surface can be authenticated by comparing the overall pattern and visual properties of each nanorod to a database. Along with nanorods, IBM researchers also created different patterns using red, green and blue fluorescent spheres [79]. A fluorescence microscope image (channel overlay) is formed, which consists of 1-m diameter fluorescent polystyrene spheres assembled in a corner array. Here, the color of the sphere is not predictable even though the position of single particles is known. It is impossible to reproduce the same colored arrays as the number of possible color combinations is considerably large.

1.6.3 Challenges and Limitations

We believe that research in the DfAC measures to prevent counterfeit ICs is still in its infancy. There are major challenges that must be overcome in the development of effective test methods. In this section, we will discuss the challenges, which urgently need to be resolved in the near future. Table 1.3 presents a comparative study of all the different counterfeit DfAC technologies. We have assigned each technology a score of high, medium, or low, depending on effectiveness [9] [10].

Table 1.3: Implementation challenges of different DfAC measures.

Avoidance Techniques	**Reliability**	**Uniqueness**	**Tamper Resistance**	**Area overhead**	**Target Counterfeit Types**	**Target Component**	**Implementation Cost**
Physically Unclonable Functions (PUF)	Medium	High	High	Low	Remarked, Overproduced, Cloned	Digital ICs	Medium
Hardware metering (HM)	Medium	High	Medium	Low/ Medium	Overproduced, Cloned	Digital ICs	High
Secure Split Test (SST)	NA	NA	Medium	Medium	Overproduced, Cloned, Out-of-spec/ defective	Digital ICs	High
Combating Die and IC Recycling (CDIR)	Medium	NA	High	Low	Recycled, Remarked	Digital ICs	Low
Electronic Chip ID (ECID)	High	High	Low	Low	Remarked	Digital ICs	Low
DNA Markings (DNA)	Medium	Medium	Medium	NA	Recycled, Remarked	All (Digital/Analog/RF/etc.)	High
Nanorods (NR)	Not Verified	Medium	Not Verified	NA	Recycled, Remarked	All (Digital/Analog/RF/etc.)	Not Verified

Reliability

A major issue that must be overcome for many of these techniques is reliability. For example, the response of a PUF must be constant for a given challenge over a wide range of environmental variations, ambient noise, and aging effects. Hardware metering approaches may also have similar reliability problem as they use PUFs. Reliability and robustness are the major challenges to incorporate the PUF in real applications [80] [81] [82]. Several designs have been proposed to make the PUF robust, unique, and reliable [81] [82] [80]. We still believe that much work needs to be done in the reliability domain. There is a serious reliability concern regarding DNA marking, as environmental conditions such as high temperatures can potentially damage the DNA and either make the sequence unreadable or change the sequence. The reliability of nanorods are not yet been verified.

Uniqueness

It is a measure of randomness between two IDs produced from DfAC measures. Ideally, two IDs should differ with a probability of 0.5 under the same test conditions. Better uniqueness makes it difficult for counterfeiters to guess new IDs after obtaining a set of IDs. PUFs and magnetic PUFs produce responses nearly equal to the ideal case [83]. In DNA, due to the very large number of base pairs, there are enough sequences to support billions of unique markings. However, in the fast-authentication mode of DNA testing, the observation of a specific "light" can be easily imitated by an adversary. For nanorods, the uniqueness of the marking is based on the number of nanorods in the pattern and the sensitivity of the measuring device to color and intensity of light. Since the exact angle of each individual nanorod is random, it is very

unlikely that the same process will produce the same result, and manually cloning the marking at nanoscale is not practical.

Tamper resistance

The difficulty faced by the attacker/counterfeiter when attempting to disable the counterfeit avoidance system is referred to as tamper resistance. It is extremely difficult to physically clone the IDs generated by PUFs. The CDIR sensors also provide high tamper resistance because they employ unavoidable aging. It is easy to clone the ECID as it is static and readable. It is easy for counterfeiters to imitate the color generated by DNA markings during fast-authentication mode. The tamper resistance of nanorods has not yet been verified.

Area overhead

It provides the area required on the die to implement a counterfeit avoidance measure. PUFs, CDIR sensors and ECID require low area overhead whereas hardware metering, and SST offer medium area overhead. DNA markings, nanorods, and magnetic PUFs do not require any area overhead on the die.

Target counterfeit types

Different available technologies target different counterfeit types. PUFs can detect remarked, overproduced, and cloned counterfeit types. SST can likely detect overproduced, out-of-spec/defective, and cloned component types. CDIR sensors are designed to target recycled and remarked types. ECID can potentially detect remarked type. DNA markings, and nanorods can detect recycled and remarked counterfeit types.

Target components

Another challenge to consider is what type of components should be targeted for implementing DfAC measures. DNA markings, and nanorods may be implemented in both analog and digital components whereas the other DfAC measures can only target the digital components. From Figure 1.12, it is clear that we have only DNA and NR to address the avoidance of small component types (small digital, entire analog and mixed signal components). However, as we described earlier that the authentication and reliability issues with DNA and NR, these entire spectrum of components need much more attention to the research community. Again, there are no technologies available to us to address the authentication of these components to prevent overproduced and out-of-spec/defective types getting into the supply chain.

Implementation cost

The cost for implementing a PUF would entail storing and maintaining the challenge-response pairs in a secure database, along with its area overhead. For hardware metering and SST, back-and-forth communication between the design house and foundry make it expensive to implement. For CDIRs, the cost comes from the area overhead. To authenticate the ICs, low-cost equipment is required. We need only a secure database to store the ECID. Thus, the cost from area overhead is negligible. The detailed authentication for identifying the plant DNA applied to the IC is expensive.

1.6.4 Summary of DfAC Measures

The prior work for detecting counterfeit ICs can be categorized in different groups discussed above. As all the previously discussed DfAC measures have limitations, we propose new DfAC measures to detect the complete spectrum of counterfeit ICs. Figure 1.12 illustrates the DfAC technologies we have identified to detect various counterfeit ICs. The x-axis and y-axis represent the counterfeit types and component types respectively. The component types on the y-axis are arranged top to bottom from lowest to highest frequency of counterfeit incidents in the supply chain [3].

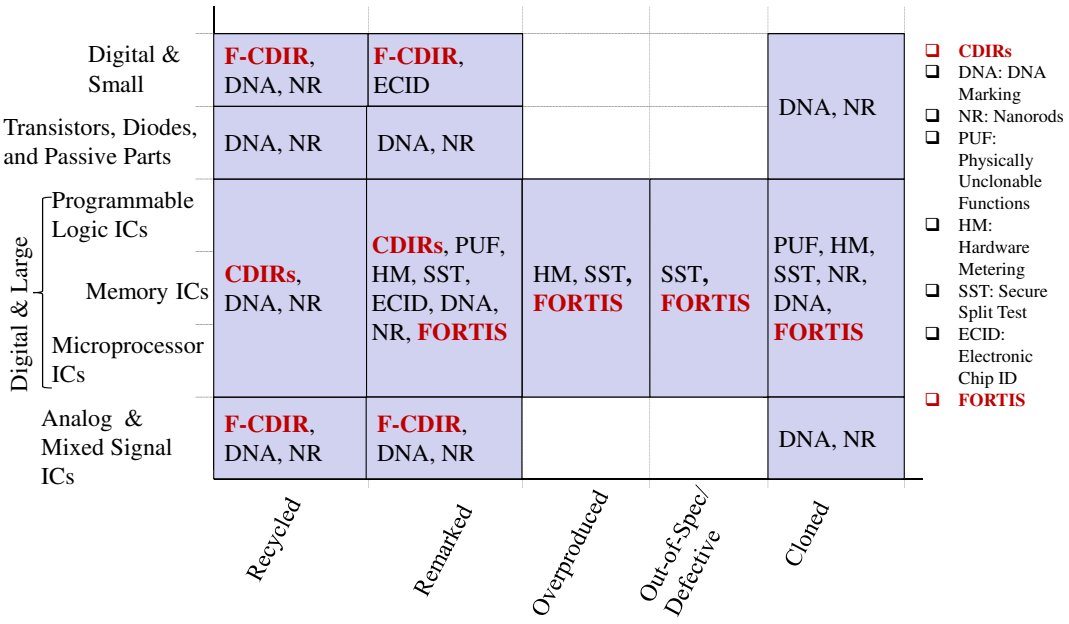


Fig. 1.12: Counterfeit avoidance technologies.

We develop the technologies which are represented in red and will be presented in Chapters 3, 4, and 5 in detail.

1.7 Organization

This thesis is divided into 6 chapters. The motivation, background, and contributions are provided in Chapter 1. This chapter provides a detailed introduction to counterfeit ICs, and all the information necessary to know about counterfeit ICs. A comprehensive taxonomy of counterfeit ICS, the vulnerabilities present in the different stages of the electronic component supply chain, all counterfeit defects, and an overview of the current state-of-the-art test methods for counterfeit IC detection. The challenges and limitations for existing tests and test procedures are also discussed in this chapter. We also describe orthogonal approaches for addressing counterfeit detection and avoidance. Rather than relying on expensive test equipment and setups, these approaches integrate new test structures and primitives into the die and/or package (i.e., DfAC measures) to actively target different counterfeit types with much greater ease.

Chapter 2 introduces novel test metrics to evaluate the effectiveness of currently available physical and electrical tests. We have developed a web-based tool called *Assessment Framework* for this purpose. The framework provides two options - (i) *static assessment* where an user can evaluate a preexisting test plan based on our newly developed test metrics; and (ii) *dynamic assessment* where the user finds an optimum set of test methods to maximize *CDC* under test time and cost constraints.

Chapter 3 presents several low-cost solutions for combating die and IC recycling (CDIR) and to detect recycling in wide range of electronic component types (from large digital ICs to small analog and discrete components). These solutions include light-weight, on-chip structures based on ring oscillators (RO-CDIR), and fuses (F-CDIR). Each structure meets the unique needs and limitations of different part types and sizes providing excellent coverage of recy-

cled parts. We present the effectiveness of our proposed negative-bias temperature instability (NBTI)-aware RO-CDIR for detecting ICs. Small analog and digital recycled components can be identified by testing our F-CDIR with very low cost measurement devices, e.g., a multimeter.

Chapter 4 introduces two improved CDIRs, which can efficiently detect recycled ICs with little misprediction when the chips are aged for a very short period of time. We present a new N-CDIR with multiple reference and stressed RO-pairs. We introduce an averaging approach to reduce the impact of process variations during the estimation of a threshold which will be used in the authentication process. We call this design multiple-pair NBTI-aware RO CDIR with averaging (AN-CDIR). This design provides a much better detection for ICs used for only few hours in the field with the cost of small misprediction. In addition, we propose another modified design of N-CDIR by adding multiple NBTI-aware reference and stressed RO-pairs like AN-CDIR. However, in this case we propose a selection algorithm to find the best reference and stressed RO-pair. We call this design multiple-pair NBTI-aware RO CDIR with selection (SN-CDIR). This CDIR with the best selected RO-pair, provides even better detection (no misprediction) of recycled ICs, even if they have been used only for a few hours, unlike the AN-CDIR. These CDIRs (AN-CDIR and SN-CDIR) are based on multiple NBTI-aware reference-stressed RO-pairs.

Chapter 5 presents a comprehensive solution for preventing IP piracy and IC overproduction by assuring forward trust between all entities involved in the system-on-chip (SoC) design and fabrication process. We propose a novel design flow to prevent IC overproduction and IP overuse. We use an existing logic encryption technique to obfuscate the netlist of an SoC or a 3PIP and propose a modification to enable manufacturing tests before the activation of chips

which is absolutely necessary to prevent overproduction. We have used asymmetric and symmetric key encryption, in a fashion similar to Pretty Good Privacy (PGP), to transfer keys from the SoC designer or 3PIP owners to the chips. In addition, we also propose to attach an IP digest (a cryptographic hash of the entire IP) to the header of an IP to prevent modification of the IP by the SoC designers. We have shown that our approach is resistant to various attacks with the cost of minimal area overhead.

We conclude this thesis in Chapter 6 with suggestions for future work.

Chapter 2

Assessment of Counterfeit Detection Methods

Because of the deficiencies in today's testing mechanisms, the detection of counterfeit integrated circuits has become a major challenge [8] [9] [10]. The detection of such components is still in its infancy, and there are major challenges that must be overcome in order for effective counterfeit detection methods to be deployed. Because counterfeiting is an evolving problem with counterfeiters acquiring increasing amounts of experience with each passing day, we must make every effort to stay ahead of them so that we can prevent the widespread infiltration of counterfeit parts into our critical infrastructures. By detecting counterfeit parts efficiently, we can also enhance the public's confidence in the security of systems that surround them. In order to achieve this goal, we must be able to continuously monitor counterfeiting activity and assess counterfeit detection methods in order to evaluate their effectiveness in detecting counterfeit components. We also need to develop a common platform to evaluate the efficacy of a set of test methods.

In this chapter, we will first develop metrics for evaluating test methods. These test metrics are (i) counterfeit defect coverage (CDC) to represent the confidence of detecting defects by a set of test methods, (ii) counterfeit type coverage (CTC) to represent the confidence of detecting specific counterfeit types by the same set of test methods, (iii) under-covered defects

(*UCDs*) and not covered defects (*NCDs*) to identify the defects those are partially detected and missed for a given set of tests. We will then present *Assessment Framework* for (i) assessing a set of test methods to evaluate their effectiveness based on these newly developed metrics. We call this as static assessment. (ii) selecting a set of test methods to maximize the test coverage considering test cost and time budget and (iii) deciding on the best set of test methods for achieving maximum test coverage. We call the combination of (ii) and (iii) as dynamic assessment. The assessment of test methods was initially introduced in [14] [13].

2.1 Test Lab Comparison Analysis

Assessing the capabilities of different test labs is now an urgent requirement. Honeywell performed round robin testing in 2012 and 2013 to certify different test laboratories (labs) and evaluate their capabilities [46] [84]. In 2012, they gave five samples of one counterfeit part (National Semiconductor DAC1230LCJ) and one authentic part (Tundra CA91L860B-50CE) to twelve test labs. The labs were encouraged to process the parts as per their standard flow with no special processing. It is of great surprise that some test labs failed to detect the counterfeit components and few others could not identify the authentic ones. In 2013, Honeywell performed this assessment again with six test labs providing five samples of two counterfeit parts (Intel TB28F400B5T80, and TDK C5750Y5V1H226Z). All the test labs correctly identified these counterfeit parts. However, they missed several defects (see Section 1.4) present in these counterfeit components.

The following were the conclusions drawn from the above test lab comparison [46]: (i) the test labs showed improved performance in identifying counterfeit parts as they gained

greater experience and exposure to different counterfeit parts, and (ii) these labs accurately detect some easy-to-detect counterfeit defects related to blacktopping, dimension and color variations, and solder issues, but they had more difficulty with hard-to-detect defects related to lead finish, dimple depth, improper materials, and electrical parameters. For some labs, defect identification was as low as 32%. Thus, there is an urgent need to assess test labs' capability in terms of quantitative measures, which will finally lead to the development of test metrics.

2.2 Terminologies used for Assessment Framework

The purpose of assessing test methods is to establish the effectiveness of the testing currently being performed to detect counterfeit components. To make it easy to understand the assessment process, we will first describe several key elements that will be used as inputs to our proposed assessment framework (see Section 2.4).

2.2.1 Tier Level

Tier level (TL) was introduced in AS6171 [28] as a means of assessing the risk associated with the use of a part while also determining the recommended level of testing that should be performed. While assessing risk, three main factors was taken into consideration: (i) the final product in which a part is used, (ii) the functionality of a part within a product, and (iii) quality attributes associated with the supplier that sells or distributes parts to various entities in the electronics supply chain. One can find a detailed description of TL in [28]. It is extremely important for user/requester to know the tier level they belong, before implementing a test plan for the screening of counterfeit parts.

Table 2.1: Target confidences for different Tier Levels.

Tier Level	**Risk Category**	**Target Confidence (TC)**
4	Critical	0.90
3	High	0.8
2	Moderate	0.65
1	Low	0.5
0	Very Low	0.35

2.2.2 Target Confidence

The target confidence (TC) for each defect is the level of confidence achieved after performing a set of tests. The value of TC for each tier level is shown in Column 3 of Table 2.1. The value of TC increases from very low to critical tier applications. We need to have higher levels of test confidence for each defect in order for higher tier levels to increase the overall level of test confidence. Based on this confidence, we will develop under-covered defects (Section 2.3.4) and guide dynamic test assessment (Section 2.4.2).

2.2.3 Test Methods

The test methods (M) are outlined in Section 1.5. One can find a detailed description in [12] [28]. All test methods are associated with their corresponding cost (C) and time (T). The test cost and time are defined as the cost and time involved in testing one batch of components.

2.2.4 Counterfeit defects

Counterfeit defects (D) are defined as the defects and anomalies seen in electronic components. These defects are presented in Section 1.4. One can find a detailed description of these defects

in [12]. Test methods are assessed based on their ability to detect one or more defects, and test confidence increases as the number of detected defects increases.

2.2.5 Confidence Level Matrix

The confidence level matrix (*CL*) represents the capability of test methods to detect counterfeit defects. When a test is performed, it detects some of the counterfeit defects. However, it does not necessarily follow that the same test will detect the same defects in different counterfeit components. Generally a confidence is involved in this detection process. In this *CL* matrix, each entry represents a certain level of confidence for detecting a defect by a given test method.

$$CL = [x_{ij}]_{m,n}$$

where, x_{ij} is the probability of detecting a defect j by a method i. Here, the rows and columns of *CL* are denoted as the methods and the defects, respectively.

If two or more methods detect the same defect, then the resultant confidence level will be increased and is given by the following equation,

$$x_{Rj} = 1 - \prod_{i=1}^{m_s} \left(1 - x_{ij}\right) \quad \text{for defect } j \tag{2.1}$$

where m_s represents the number of tests in the recommended test set. The vector for resultant confidence becomes,

$$x_R = [x_{R1}\ x_{R2}\ \ldots\ x_{Rn}] \tag{2.2}$$

2.2.6 Defect Frequency

Defect frequency (DF) is defined as how frequently the defect is visible in a counterfeit component. This is one of the key parameters for evaluating test coverage, as the detection of high frequency defects has more of an impact than the detection of low frequency defects.

2.2.7 Decision Index

The decision index (DI) is defined as the probability that a counterfeit type contains one or more known counterfeit defects. It can also be interpreted as the probability of identifying a component belonging to a counterfeit type after targeting all defects. It is not necessarily true that every occurrence of a counterfeit type will contain a one or more defects. For example, DI may approach zero for certain counterfeit types - such as overproduced and cloned counterfeit types - due to the rare occurrence of defects. Table 2.2 shows the DI values for different counterfeit types.

Table 2.2: Decision index for each counterfeit type.

Counterfeit Type	**Decision Index**
Recycled	0.98
Remarked	0.90
Overproduced	0.03
Out-of-Spec/Defective	0.50
Cloned	0.10
Forged Documentation	0.70

Table 2.3: Terminologies used in our proposed method selection algorithm

Terminology	**Notation**
Test Methods	$M = [M_1\ M_2\ \ldots\ M_m]$, where m is the number of test methods.
Test Cost	$C = [C_1\ C_2\ \ldots\ C_m]$
Test Time	$T = [T_1\ T_2\ \ldots\ T_m]$
Counterfeit Defects	$D = [D_1\ D_2\ \ldots\ D_n]$, where n is the number of defects.
Tier Level	$TL = [L_1\ L_2\ \ldots\ L_5]$, L_1: Critical, L_2: High, L_3: Moderate, L_4: Low, L_5: Very Low
Target Confidence	$TC = [TC_1\ TC_2\ \ldots\ TC_5]$, TC_1: Critical, TC_2: High, TC_3: Moderate, TC_4: Low, TC_5: Very Low
Confidence Level	$CL = [x_{ij}] = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix}$ where, $x_{ij} = Pr$ (Detecting a defect j by a method i). Here, the rows and columns of CL are denoted as the methods and the defects.
Defect Frequency	$DF = [DF_1\ DF_2\ \ldots\ DF_n]$
Defect Mapping	$DM = [w_{ij}] = \begin{bmatrix} w_{11} & w_{12} & \ldots & w_{17} \\ w_{21} & w_{22} & \ldots & w_{27} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{n7} \end{bmatrix}$ where, $w_{ij} \in \{0,1\}$ ={Not Present, Present}, and rows and columns represent defects and counterfeit types respectively.

2.2.8 Defect Mapping Matrix

The defect mapping (DM) matrix represents the presence of a defect in a counterfeit type. It is not necessary for a defect to be visible in all the counterfeit types simultaneously. For example,

the defect *Invalid Lot/Date/Country Code* may not be present in overproduced, or cloned types. Here, each entry of *DM* equals 1 if the defect may be present for a counterfeit type and 0 if the defect is never present.

In summary, Table 2.3 summarizes all the key elements that will be used in the assessment framework.

2.3 Test Metrics

To evaluate the effectiveness of these test methods, it is of the utmost importance to develop test metrics. These metrics are described below:

2.3.1 Counterfeit Defect Coverage

Counterfeit defect coverage (CDC) is defined as the ratio of all probable detectable defects by a set of (single) test methods (method) to the total number of known counterfeit defects. It provides a cumulative confidence of identifying a component as counterfeit by a sequence of test methods. CDC can be expressed as:

$$CDC = \frac{\text{Probable Detectable Defects}}{\text{Total Defects}} \times 100\% \tag{2.3}$$

A level of confidence is involved when a test method detects a counterfeit defect, and this is captured in the *CL* matrix. When a defect is detected by multiple test methods, the confidence of identifying it increases. The maximum value of this confidence is bounded by "1", which signifies this particular defect will surely be detected by these test methods. Now the total confidence of finding a part as counterfeit, *CDC*, becomes the ratio of the cumulative sum of the resultant confidence of all the defects to the total number of defects. Mathematically

it can be expressed as:

$$CDC = \frac{\sum_{j=1}^{n}(x_{Rj})}{n} \times 100\% \tag{2.4}$$

where x_{Rj} denotes the resultant confidence for defect j and n denotes total number of defects.

Equation 2.4 (shown above) treats all the defects as equally likely in the component supply chain. However, some defects are more frequent than others, and we need to incorporate defect frequency in the calculation of *CDC*. Therefore, the modified equation of *CDC* becomes,

$$CDC = \frac{\sum_{j=1}^{n}(x_{Rj} \times DF_j)}{\sum_{j=1}^{n} DF_j} \times 100\% \tag{2.5}$$

where DF_j represents the defect frequency for defect j.

2.3.2 Counterfeit Type Coverage

Defects are not equally visible in all counterfeit types. Some defects may be present in some particular counterfeit types, but, not in other types, which is captured in defects mapping (*DM*) matrix, and for some counterfeit types, the probability of finding any defects are extremely small, which is captured in decision index (*DI*). For example, overproduced parts may be as good as new authentic parts and be free from any counterfeit defects. As such, the detection of defects does not necessarily provide the correct test coverage, *CDC*, which was introduced in Section 2.3.1. We will now introduce, counterfeit type coverage (*CTC*) to represent the test coverage for individual counterfeit types by a set of test methods.

CTC is defined as a measure to detect a counterfeit type given the test methods per-

formed. CTC can be expressed as:

$$CTC = DI \times \frac{\text{Probable Detectable Defects for a Counterfeit Type}}{\text{Total Defects for a Counterfeit Type}} \times 100\% \tag{2.6}$$

where, DI represents the decision index.

CTC can be expressed as the total confidence of finding a part that belongs to a particular counterfeit type. Taking CTC for a counterfeit type, k becomes the ratio of the cumulative sum of the resultant confidence of all the defects detected by the test methods to the total number of defects belonging to that counterfeit type and expressed as:

$$\text{CTC}_k = \text{DI}_k \times \frac{\sum_{j=1}^{n}(x_{Rj} \times w_{jk})}{\sum_{j=1}^{n}(w_{jk})} \times 100\% \tag{2.7}$$

where,

CTC_k : CTC for counterfeit type k;

DI_k : DI for counterfeit type k;

x_{Rj} : Resultant confidence for defect j;

w_{jk} : The presence of defect j in counterfeit type k ($\in \{0,1\}$).

Equation 2.7 (shown above) treats all the defects as equally likely in the component supply chain. However, some defects are more frequent than others, so we need to incorporate defect frequency in the calculation of CTC. The modified equation of CTC becomes

$$\text{CTC}_k = \text{DI}_k \times \frac{\sum_{j=1}^{n}(x_{Rj} \times DF_j \times w_{jk})}{\sum_{j=1}^{n}(DF_j \times w_{jk})} \times 100\% \tag{2.8}$$

where, DF_j: Defect frequency of defect j.

2.3.3 Not-Covered Defects

A set of test methods will not necessarily detect a particular counterfeit defect. A defect is called as a not-covered defect (*NCD*) when a set of test methods cannot detect it. A counterfeit defect j will be a *NCD* if

$$x_{Rj} = 0 \tag{2.9}$$

2.3.4 Under-Covered Defects

A defect is called an under-covered defect (UCD) when a set of test methods cannot provide the desired confidence level. Defects belong to this category when the resultant confidence level for detecting a defect is less than the target defect confidence level. A defect j will be a UCD if

$$x_{Rj} < TC \tag{2.10}$$

2.4 Assessment Framework

Different sequences of test methods have been developed by organizations for the detection of counterfeit parts. The assessment framework evaluates the effectiveness of a sequence of test methods used to screen for counterfeit parts. This framework works in two different modes. In the static assessment, it performs the assessment of a sequence of tests under evaluation. The output of this mode produces the test metrics (*CDC*, CTC, *NCD*, and UCD). In the dynamic assessment, the framework receives all the current available test methods as input and recommends (i) the best set of tests and (ii) an optimum set of tests that provides maximum coverage

within a certain test time and cost budget. Then the assessment is done on the basis of the same test metrics.

2.4.1 Static Assessment

The static assessment provides the test labs with an evaluation of the effectiveness of a specified test plan consisting of a sequence of tests, as it is necessary to evaluate the capability of the test labs. The term "static" suggests that, in this kind of assessment, the test methods put into this framework do not change, and the assessment is performed on the whole set of test methods.

Assessment of Test Methods

Algorithm 1 shows the flow of this assessment framework. The user specified test methods are provided to this framework as inputs. It selects the target confidence from the user specified risk category (tier level breakpoints). It then reads the confidence level matrix (*CL*), decision index (*DI*), and defects mapping matrix (*DM*) from a secured database. The function CALCULATE() in line 2, calculates the resultant confidence level for all the defects. The CALCULATE() functions in lines 4-7, calculate *CDC*, *CTC*, *NCDs* and *UCDs*.

Example 1

Let us assume that we want to assess five test methods for critical tier level (tier 4, described in Table 2.1). We also assume that there are five test methods ($\{$M1, M2, M3, M4, M5$\}$) present for counterfeit detection and five counterfeit defects ($\{$D1, D2, D3, D4, D5$\}$) present in the counterfeit parts with a given confidence level (*CL*) matrix and defect frequency vector of

Algorithm 1: Procedure *Static_Assessment*

Report test metrics for a preexisting test plan.

input : User specified test methods (M^S), confidence level matrix (CL), Tier Level (TL), decision index (DI), and defects mapping matrix (DM)

output: Report CDC, CTC, $NCDs$ and $UCDs$

1 **for** $j := 1$ **to** n *in* D **do**

2 Calculate x_{Rj}, $x_{Rj} \leftarrow$ CALCULATE (X, M^S) ;

3 **end**

4 Calculate counterfeit defect coverage, $CDC \leftarrow$ CALCULATE (x_R, DF) ;

5 Calculate counterfeit type coverage, $CTC \leftarrow$ CALCULATE (x_R, DF, DI, DM) ;

6 Calculate not-covered defects, $NCDs \leftarrow$ CALCULATE (x_R) ;

7 Calculate under-covered defects, $UCDs \leftarrow$ CALCULATE (x_R, TC) ;

$$CL = \begin{array}{c} \\ M1 \\ M2 \\ M3 \\ M4 \\ M5 \end{array} \begin{array}{c} \begin{array}{ccccc} D1 & D2 & D3 & D4 & D5 \end{array} \\ \begin{bmatrix} 0.9 & 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.9 & 0.0 & 0.5 \\ 0.0 & 0.9 & 0.0 & 0.0 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.9 & 0.0 & 0.0 \end{bmatrix} \end{array}, \quad \text{and} \quad DF = \begin{array}{c} \\ D1 \\ D2 \\ D3 \\ D4 \\ D5 \end{array} \begin{array}{c} \\ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{array}$$

This matrix can be interpreted as follows: Each row represents a test method (e.g., the first row represents the test method M1, the second row represents the test method M2, and so on). Each column represents a defect (e.g., the first column represents defect a, the second column represents defect b, and so on). Each entry denotes the confidence of detecting a defect

using a test method. This means that test M1 has a 0.9 probability of detecting defect D1, a 0.5 probability of detecting defect D2, and a 0 probability of detecting defects D3, D4, and D5.

We also assume that there are three counterfeit types ({x,y,z}) with defect mapping (*DM*) matrix and decision index (*DI*) vectors of

$$DM = \begin{array}{c} \\ D1 \\ D2 \\ D3 \\ D4 \\ D5 \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}, \quad \text{and} \quad DI = \begin{array}{c} \\ x \\ y \\ z \end{array} \begin{bmatrix} 0.9 \\ 0.5 \\ 0.1 \end{bmatrix}$$

Table 2.4 summarizes the assessment process. The *CDC* is 68.8%, whereas the *CTC* for counterfeits x, y, and z is 55.8%, 24.8%, and 8.1%, respectively. The lower value of *CTC* for counterfeits y and z points to the fact that the defects related to those counterfeit types are not visible as we can see from the *DI* vector that the probability of finding any counterfeit defect is 0.5 and 0.1, respectively.

2.4.2 Dynamic Assessment

We need to identify a set of test methods targeting critical risk applications in order to maximize the test confidence in our ability to detect counterfeit parts. At the same time, when the risk category level is low, the user does not need to engage in exhaustive testing. In this case, test time and cost is more important, and we need to find the best set of tests that can give the maximum coverage under these test time and cost constraints. In the following, we will first

Table 2.4: Example for the Static Assessment.

Step	Name	Description
	Read Inputs	Read M^S, TL, CL, DI, and DM
Line 1-3	Compute resultant confidence (x_R) using Equation 2.1	$x_{RD1} = 1-\{(1-0.9)(1-0)(1-0)(1-0.9)(1-0)\} = 0.99$ $x_{RD2} = 1-\{(1-0.5)(1-0)(1-0.9)(1-0)(1-0)\} = 0.95$ $x_{RD3} = 1-\{(1-0)(1-0.9)(1-0)(1-0)(1-0.9)\} = 0.99$ $x_{RD4} = 1-\{(1-0)(1-0)(1-0)(1-0)(1-0)\} = 0.00$ $x_{RD5} = 1-\{(1-0)(1-0.5)(1-0)(1-0)(1-0)\} = 0.50$
Line 4	Compute CDC using Equation 2.5	$CDC = 100 * \frac{1*0.99+1*0.95+1*0.99+1*0.00+1*0.50}{1+1+1+1+1+1}\% = 68.6\%$
Line 5	Compute CTC using Equation 2.8	$CTC_x = 0.9 * \frac{1*0.99+0*0.95+1*0.99+1*0.00+1*0.5}{1+0+1+1+1} \times 100\% = 55.8\%$ $CTC_y = 0.5 * \frac{0*0.99+0*0.95+1*0.99+1*0.00+1*0.5}{0+0+1+1+1} \times 100\% = 24.8\%$ $CTC_z = 0.5 * \frac{1*0.99+1*0.95+0*0.99+0*0.00+1*0.5}{1+1+0+0+1} \times 100\% = 8.1\%$
Line 6	Compute $NCDs$ using Equation 2.9	NCD: Defect D4 as $x_{Rd} = 0$
Line 7	Compute $UCDs$ using Equation 2.10	UCD: Defect D5 as $x_{Re} < TC$ $(0.50 < 0.90)$)

present the method selection technique and then we will assess those selected test methods.

Selection of Test Methods

The objective of the method selection algorithm is to find an optimum set of methods to maximize counterfeit defect coverage while also allowing for a consideration of the test time, cost, and risk category constraints for certain applications. A counterfeit defect can be detected by multiple methods with different levels of confidence. Thus, the problem becomes that of se-

lecting the most suitable methods for achieving the highest *CDC* possible given the presence of practical constraints.

The problem can be formulated as follows:

Select a set of methods $M^S \subset M$ *to Maximize CDC*

Subjected to:

$$x_{Rj} \geq TC,\ \forall\ j \in \{1:n\} \qquad \textit{for critical applications}$$

or

$$\begin{cases} x_{Rj} \geq TC,\ \forall\ j \in \{1:n\} \\ M_1C_1 + M_2C_2 + \ldots + M_mC_m \leq C_{user} \qquad \textit{for non-critical applications} \\ M_1T_1 + M_2T_2 + \ldots + M_mT_m \leq T_{user} \end{cases}$$

where,

x_{Rj}: Resultant confidence for defect j;

TC: Target confidence;

M_i: Test method i, $M_i \in \{0,1\} = \{\text{Not Selected, Selected}\}$;

C_i: Test cost for test method i;

T_i: Test time for test method i;

m: Number of test methods;

n: Number of defects;

C_{user}: User specified total test cost;

T_{user}: User specified total test time;

Algorithm 2 describes the selection of the test methods. It starts by initializing the recommended test set to null. It then gets the defect frequency (DF) and the target confidence level (TC) for each defect. Next, it prioritizes the defects by sorting them according to DF, as we want to capture high-frequency defects first to achieve a higher CDC.

For critical risk applications, our primary objective is to obtain the maximum CDC regardless of test cost and time. On the other hand, for low and very low risk applications, test time and cost are more important than getting the maximum CDC. For medium- and high-risk applications, we can get a higher confidence level by setting a higher test time and cost limit. For critical applications, the SORT() function (line 7) takes M and CL as arguments and sorts them according to x_{ij} and discards the method i when $x_{ij} = 0$. Equation 2.1 has been implemented by the CALCULATE() function (line 8). The SELECTMETHODS() function (line 10) takes x_{Rj} and TC as arguments and selects methods until the condition, $x_{Rj} \geq TC$, is met. If this condition is not met after iterating all the methods, then the defects belong to the $UCDs$. If $x_{Rj} = 0$, then the defects become $NCDs$.

For other applications, the SORT() function (line 15) takes M, T, and C as arguments and sorts according to linear combinations of t_i and c_i ($0.5t_i$+$0.5c_i$) and discards the method i when $x_{ij} = 0$. The resultant confidence level has been calculated by the CALCULATE() function (line 16) through the implementation of Equation 2.1. The SELECTMETHODS() function (line 18) takes x_{Rj}, TC, t_{user}, and c_{user} as arguments and selects the methods that require the minimum test time and cost to achieve $x_{Rj} \geq TC$. If this condition is not met after iterating all the methods (as was the case for the critical applications, as well), then the defects belong to the $UCDs$ and, if $x_{Rj} = 0$, then the defects become $NCDs$.

Algorithm 2: Procedure *Method_Selection*

Select a sequence of test methods to maximize test coverage.

input : Test methods (M), confidence level matrix (CL), Tier Level (TL), decision index (DI), and defects mapping matrix (DM)

output: Report a sequence of test methods M^S

```
1  Initialize selected methods, M^S ← {φ} ;
2  Specify cost limit set by the user c_user except for critical risk applications ;
3  Specify test time limit set by the user t_user except for critical risk applications ;
4  Sort defects according to defect frequency, D ← SORT (DF) ;
5  if (TL == critical) then
6      for j := 1 to n in D do
7          Sort methods according to x_ij, M' ← SORT (M, CL) ;
8          Calculate x_Rj, x_Rj ← CALCULATE (CL, M') ;
9          for i := 1 to m in M' do
10             SELECTMETHODS (CL, M', x_Rj, TC) ;
11         end
12     end
13 else
14     for j := 1 to n in D do
15         Sort methods according to test time and cost, M' ← SORT (M, T, C) ;
16         Calculate x_Rj, x_Rj ← CALCULATE (CL, M') ;
17         for i := 1 to m in M' do
18             SELECTMETHODS (CL, M', x_Rj, TC, t_user, c_user) ;
19         end
20     end
21 end
```

Assessment of Test Methods

After the selection of the test methods, the assessment is done on those methods. It invokes Algorithm 1 with *DI*, *DM*, and selected methods by Algorithm 2 as inputs to compute *CDC*, *CTC*, *NCDs*, and *UCDs*.

Example 2

Let us now start with an example to explain the dynamic assessment. All the data used for this example are the same as the data used in Example 1. In this example, we will consider low risk categories. The target confidence (TC) corresponding to this risk category is 0.5 (described in Table 2.1). For simplicity's sake, we are not considering test time and cost in this example.

2.5 Results

We have developed a web-based tool, *Assessment Framework*, to perform the assessment of the efficiency of a test plan. The tool is currently deployed in the server of University of Connecticut's CHASE Center. This tool can be accessed at $http://ece-chaseweb.engr.uconn.edu/cdc-site/index.php$. The user of this tool needs to have the required authentication to access the tool.

2.5.1 Static Assessment

In static assessment, the evaluation of a preexisting test plan is performed. For example, a test lab wants to assess a test plan, which consists of 11 test methods (see Table 2.6) for moderate risk applications. Column 3 of Table 2.6 represents the *CDC*. General EVI alone contributes the coverage of 17.6%. General EVI and detailed EVI contribute a combined coverage of 41.34%.

Table 2.5: Example for the Dynamic Assessment.

	Step	Name	Description
Selection (Algorithm 2)		Read Inputs	Read TL, CL, and DF
	Line 4	Sort DF	No sort needed as all the defects are treated equally
	Line 14-20	Select Methods	Defect D1: Select method M1 Defect D2: Method M1 already selected and $x_{RD2} = TC$, No extra methods are necessary. Defect D3: Select M2 Defect D4: No test methods can detect D4. Defect D5: Method M2 already selected and $x_{RD5} = TC$, No extra methods are necessary. Selected methods are M1 and M2.
Assessment (Algorithm 1)		Read Inputs	Read DI, and DM
	Line 1-3	Compute resultant confidence (x_R) using Equation 2.1	$x_{RD1} = 1-\{(1-0.9)(1-0)\} = 0.9$ $x_{RD2} = 1-\{(1-0.5)(1-0)\} = 0.5$ $x_{RD3} = 1-\{(1-0)(1-0.9)\} = 0.9$ $x_{RD4} = 1-\{(1-0)(1-0)\} = 0.0$ $x_{RD5} = 1-\{(1-0)(1-0.5)\} = 0.50$
	Line 4	Compute CDC using Equation 2.5	$CDC = 100 * \frac{1*0.9+1*0.5+1*0.9+1*0.0+1*0.5}{1+1+1+1+1} \% = 56\%$
	Line 5	Compute CTC using Equation 2.8	$CTC_x = 0.9 * \frac{1*0.9+0*0.5+1*0.9+1*0.0+1*0.5}{1+0+1+1+1} \times 100\% = 51.75\%$ $CTC_y = 0.5 * \frac{0*0.9+0*0.5+1*0.9+1*0.0+1*0.5}{0+0+1+1+1} \times 100\% = 23.3\%$ $CTC_z = 0.5 * \frac{1*0.9+1*0.5+0*0.9+0*0.0+1*0.5}{1+1+0+0+1} \times 100\% = 6.3\%$
	Line 6	Compute $NCDs$ using Equation 2.9	NCD: Defect D4 as $x_{Rd} = 0$
	Line 7	Compute $UCDs$ using Equation 2.10	None

The combined coverage of the total 11 test methods gives a final CDC of 66.48%. This signifies the fact that we are 66.48% confident of finding a part as counterfeit after performing these

test methods. There are 6 defects that cannot be detected by these tests and become *NCDs*. In addition, there are 16 defects that are properly covered (target confidence for moderate risk category is 65%) and become *UCDs*.

Table 2.6: Static Assessment of Test Methods for Moderate Risk Category (CDC, NCDs, and UCDs)

#	**Test Method**	**CDC (%)**	**NCDs**	**UCDs**
1	General EVI	17.6	6	16
2	Detailed EVI	41.34		
3	Testing for Remarking (EVI)	41.96		
4	Testing for Resurfacing (EVI)	42.51		
5	Lead Finish Analysis (XRF)	44.11		
6	Lead Finish Thickness (XRF)	44.15		
7	Material Composition (XRF)	45.33		
8	Internal Inspection (DDPA)	58.24		
9	2D Radiological Inspection (RI)	62.88		
10	PEMS-External Only (AM)	63.98		
11	DC Test at ambient temperature	66.48		

Table 2.7 shows the *CTCs* for all the counterfeit types. As we explained before, detecting defects can help us identify a component as counterfeit. However, this cannot provide the necessary confidence that the counterfeit component belongs to a particular type. *CTC* gives what is desired for finding a counterfeit type. The *CTCs* for recycled and remarked types are close to *CDC*, as the probability of finding any counterfeit defects is close to 1 (i.e., 0.98 and 0.9 for recycled and remarked types indicated in the *DI* vector in Table 2.2). However, in overproduced and cloned types, the *CTCs* are quite small and are 1.4%, and 5.71%, respectively.

The probability of finding counterfeit defects in these counterfeit types is extremely small. This signifies that we need a different set of measures (designed for anti-counterfeit, DFAC) to detect these counterfeit types. We use the term "not applicable" (N/A) in the *CTC* field for tampered types in Table 2.7.

Table 2.7: Static Assessment of Test Methods for Moderate Risk Category (CTC).

#	Counterfeit Type	CTC
1	Recycled	65.81
2	Remarked	61.21
3	Overproduced	1.4
4	Out-of-Spec./Defective	27.28
5	Cloned	5.71
6	Forged Documentation	60.58
7	Tampered	N/A

2.5.2 Dynamic Assessment

In dynamic assessment, the best set of test methods are determined based on the user specified test time and test cost. For example, a test lab wants to find the best set of tests for moderate risk applications. Table 2.8 shows the dynamic assessment of the test methods for the moderate risk category. Column 2 shows the recommended test methods. The test cost and time budgets are not mentioned here due to the confidentiality agreement between the CHASE Center and the G-19A group. However, the total test cost and time for these six recommended test methods are well below compared to the static assessment. Columns 3, 4, and 5 represent the *CDC*, *NCDs*, and *UCDs*, respectively.

The first recommended test method, general EVI, contributes a coverage of 17.6%, as before. The second recommended test method, internal inspection combined with general EVI, contributes 41.34% coverage. The combined coverage of the total 7 test methods provides a final *CDC* of 66.4%. We can see that there is a significant reduction in the total number of test methods (11 to 7) in the dynamic assessment. The *NCD* or *UCD* values are comparable for both the assessments.

Table 2.8: Dynamic Assessment of Test Methods for Moderate Risk Category (CDC, NCDs, and UCDs).

#	Test Method	CDC (%)	NCDs	UCDs
1	General EVI	17.6	8	13
2	Detailed EVI	41.34		
3	Part Dimensions	43.24		
4	Internal Inspection (DDPA)	56.22		
5	Bond Pull (DDPA)	57.72		
6	2D Radiological Inspection (RI)	62.5		
7	RAMAN	66.4		

Table 2.9 shows the *CTCs* for all the counterfeit types in the low risk category. Here we can observe the similar *CTC* values for the dynamic and static assessments for all the counterfeit types, as both assessments provide similar *CDCs*.

2.6 Summary

In this chapter, we have developed a comprehensive framework for assessing currently available test methods by introducing test metrics such as counterfeit defect coverage (*CDC*), counterfeit type coverage (*CTC*), under-covered defects (*UCD*) and not-covered defects (*NCD*). The

Table 2.9: Dynamic Assessment of Test Methods for Moderate Risk Category (CTC).

#	Counterfeit Type	CTC
1	Recycled	65.27
2	Remarked	57.04
3	Overproduced	1.4
4	Out-of-Spec./Defective	27.33
5	Cloned	5.76
6	Forged Documentation	59.28
7	Tampered	N/A

framework provides two types of assessments: static assessment that helps in the evaluation of test methods based on the aforementioned metrics, and dynamic assessment for selecting an optimum set of test methods to maximize the test coverage. Static assessment can be used to estimate the counterfeit detection capabilities of a test lab based on their equipment and test methods. The dynamic assessment can be used by test labs to determine how much they can improve their capabilities by adding different equipment and test capabilities. It can also illustrate the trade-off between test time, cost, and counterfeit coverage. Both types of assessment can determine what counterfeit defects are partially covered or missed, what counterfeit types are not well covered, etc. This information can be used to guide in the development of new test methods for counterfeit detection.

Chapter 3

Combating Die and IC Recycling

The technologies (ECID, PUFs, HM, and SST) discussed in Chapter 1 Section 1.6, are not suitable for detecting recycled ICs as long as the counterfeiters maintain the same grade (e.g., commercial grade component remains same). In addition, many of these technologies cannot be implemented on small parts because of their large area overhead. They are also inapplicable on analog and mixed-signal components due to the difference in technologies. DNA and NR have their own challenges for use in IC authentication. In this chapter, we present very low-cost structures that can be implemented in the full spectrum of components to detect recycled and remarked types. These technologies are added to the die, making them suitable for new components.

Along with DfAC measures, several approaches have been proposed to detect recycled ICs. Zhang et al. proposed path-delay fingerprinting, where used components can be differentiated from their genuine counterparts as their path delay distribution changes [85]. This technique, however, presents several shortcomings, one of which is that it requires data from genuine ICs and cannot be easily applied to analog/RF/mixed-signal devices. Huang et al. presented a statistical approach to distinguish recycled ICs by measuring electrical parameters and using a one-class support vector machine (SVM) [86]. Like path-delay fingerprinting, this technique requires a large number of genuine samples for SVM training. This may not be feasible as there are thousands of different types of components available in the supply chain, making it difficult to find large numbers of genuine samples. Thus, it is of utmost importance that we develop a new, practical DFAC structures that will enable easy counterfeit detection without the need for existing expensive test methods and/or genuine ICs. Zheng et al. utilized dynamic current analysis to determine the aging difference between high-activity and low-activity por-

tions of symmetric structures [87]. However, this approach requires at least a year of aging for reliable detection of recycled ICs.

The above discussion highlights the major challenges that must be overcome in order to realize more effective DfAC measures. In this chapter, we address the shortcomings of prior work by *(i)* developing separate measures for analog and digital components as they are of different sizes and use different manufacturing technologies; *(ii)* keeping the cost/overheads of adding the DfAC measures as low as possible; and *(iii)* enabling fast authentication with low-cost test devices that do not require genuine ICs for the purpose of comparison.

We meet these objectives by presenting several new combating die and IC recycling (CDIR) structures. In Section 3.1, we present a lightweight ring-oscillator-based CDIR structure suitable for both large and small digital ICs. The CDIR was first presented by Zhang et al. in [1] and we call this as simple RO-CDIR [1]. We have proposed an improved version and we call this as NBTI-aware RO-CDIR [15]. Our proposed NBTI-aware RO-CDIR exploits aging much better than the simple RO-CDIR so that it is able to capture very short usage time for a chip. In Section 3.2, we present two fuse-based CDIR (F-CDIR) structures primarily aimed at analog and small ICs. A very low-cost measurement device such as a multimeter can authenticate the component with these F-CDIRs.

Depending on the size of the chip and the accuracy required in measuring the IC usage, one can select one or a combination of these CDIR structures for recycled IC detection. Note that in this chapter, we only address the remarking of recycled ICs, not the remarking of new ICs.

3.1 RO-Based CDIR Sensor

The first set of avoidance measures are to be taken by placing ring-oscillator-based CDIRs (RO-CDIRs) in the digital ICs. This simple elegant structure utilizes aging efficiently to authenticate ICs as counterfeit or not. In the following, we will describe aging phenomenon in detail, and then present two different versions of RO-CDIR.

Recycled ICs are characterized by aging, i.e., prior usage has taken its toll on the components' life and performance. A shift in the components' parameters due to aging will occur

when they are used in the field for some time, which leads to the development of parametric defects and anomalies in the component. Aging of a component used in the field can be attributed to two major, distinct phenomena (which are becoming more prevalent as the technology scales down). They are negative-bias temperature instability (NBTI) and hot carrier injection (HCI) which are prominent in PMOS and NMOS devices, respectively. NBTI occurs in p-channel MOS devices stressed with negative gate voltages and elevated temperatures due to the generation of interface traps at the $Si - SiO_2$ interface. Removal of the stress can anneal some of the interface traps, but not completely. As a result, it manifests as the increase in threshold voltage (V_{th}) and absolute off current (I_{off}) and the decrease of absolute drain current (I_{DSat}) and transconductance (g_m). HCI occurs in NMOS devices caused by the trapped interface charge at $Si - SiO_2$ surface near the drain end during switching. It results in non-recoverable V_{th} degradation. These two aging mechanisms lead to the increased delay in the components' internal paths, which ultimately reduces the component's operating speed. Now the obvious question is *can aging help us to detect recycled ICs?* And, the answer is *yes*!

Prior approaches [85] [86] for the detection of recycled ICs, have exploited this aging phenomenon. These approaches require that the performance measurements of fresh chips be collected and analyzed, a challenge for legacy parts when golden ICs may not be available. Furthermore, large process variations in lower technology nodes can make it very difficult to separate recycled ICs from a batch when the process variation outpaces aging degradation.

3.1.1 Simple RO-CDIR

A different approach was proposed in [1] based on ring oscillators (ROs) that avoided the data collection altogether and applied a "self–referencing" concept to the measurement of use time. Specifically, [1] embeds two ROs within the chip and compares them to detect prior IC usage. The first RO is called the *reference RO* and is designed to age at a slow rate. The second RO is referred to as the *stressed RO*, and it is designed to age at a much faster rate than the reference RO. As the IC is used in the field, the stressed RO's rapid aging reduces its oscillation frequency while the reference RO's oscillation frequency remains largely static over the chip's lifetime. Thus, a large disparity between the two ROs' frequencies implies that the chip has been used. To overcome global and local process variations, the two ROs are placed physically very close

together so that the process and environmental variations between them are negligible.

Figure 3.1 shows the structure of this simple RO-CDIR, 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 time period controlled by the timer. The 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 signal. The inverters in the ROs can 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-CDIR significantly, according to prior analysis in [1]. In 90nm technology, a 16-bit counter can operate at a frequency of up to 1GHz, which means that an inverter-based RO must be composed of at least 21 stages [1].

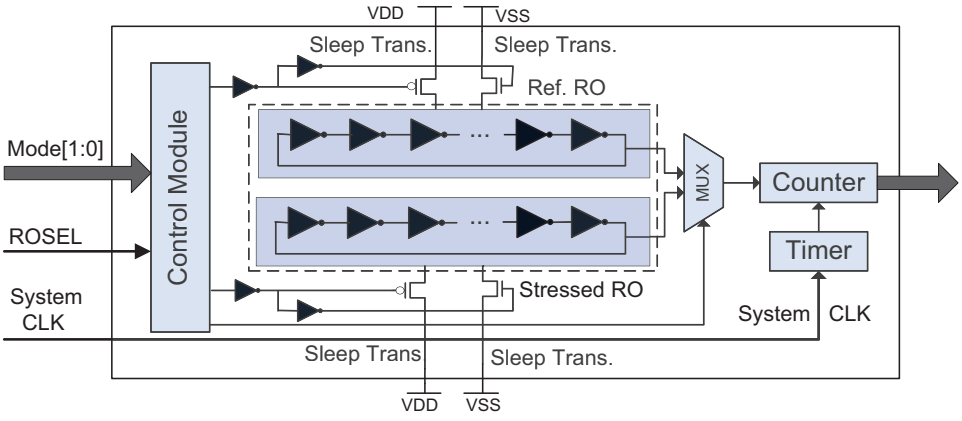
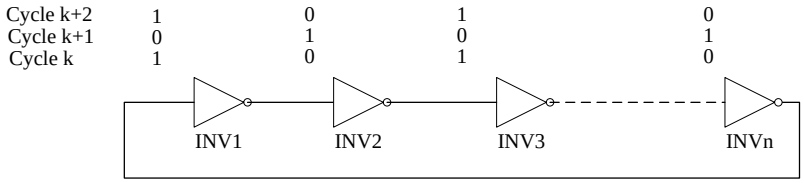


Fig. 3.1: Simple RO-CDIR sensor [1].

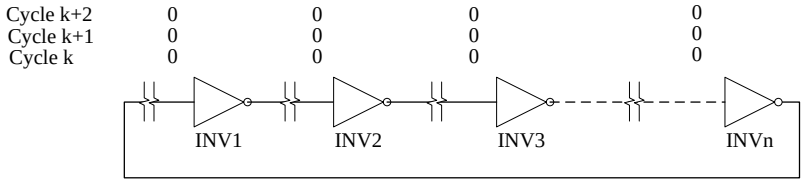
3.1.2 Limitations of Simple RO-CDIR

Given the objective for designing RO-CDIR, the best RO-CDIR sensor (i.e., the one that detects recycled ICs most accurately) should possess minimal aging for the reference RO and maximum aging for the stressed RO. This cannot be achieved by the RO-CDIR proposed in [1] because, in the RO-CDIR design shown in Figure 3.1, only half of the inverters in the stressed RO are negative-bias temperature instability (NBTI) stressed in one oscillation cycle as shown in Figure 3.2(a). This implies that half of the inverters age while the other half recover some of their aging. For example, at cycle time k, the even number inverters (e.g., inverters 2, 4, 6, ...) are stressed as

they receive zero at their inputs (zero causes the PMOS to age) while odd number inverters (e.g., inverters 1, 3, 5, ...) recover their aging. At cycle time $k+1$, the even number inverters recover and odd number inverters age. This process continues during normal operations and results in a slower aging for the stressed RO because the PMOS transistors partially recover every other cycle. Hot carrier injection (HCI) will not contribute as much to the total degradation of this sensor in the field since the sensors are kept in non-oscillatory mode. More details on the aging and recovery process can be found in [88] [89].



(a) Stressed RO in RO-CDIR sensor in stress mode.



(b) Stressed RO in our proposed NBTI-Aware RO-CDIR sensor in stress mode.

Fig. 3.2: NBTI stress on stressed ROs.

This problem is overcome in [15] where all the inverters are NBTI stressed during the entire operation of an IC where the RO-CDIR is deployed. Figure 3.2(b) shows the proposed solution where all the inverters are stressed during normal operations. This is achieved by breaking the connection of each inverter to its prior one and pulling down their inputs to ground. NBTI stress occurs when the gate of a PMOS transistor is pulled down to ground. Thus, all the inverters of the stressed RO are NBTI stressed during the entire time of the operation. As a result, the aging recovery cannot take place. However, if the chip is completely powered off, a partial recovery may occur. Nevertheless, the permanent degradation is proven to be much larger than the recovery [90].

3.1.3 Design and Operation of NBTI-aware RO-CDIR

Figure 3.3 shows the design of the NBTI-Aware RO-CDIR sensor [15]. The stressed RO is modified in such a way that all the inverters are stressed constantly during normal operation, as explained above. To achieve this, a pass transistor is introduced in between every pair of inverters, and the inputs of all the inverters are pulled down to ground using an NMOS network. To match all the internal parameters (node capacitance, resistance, etc.), the same pass transistor and NMOS are mimicked in the reference RO. This is to ensure that at time 0, when there is no aging, the difference between the two ROs is minimal and is mainly impacted by the manufacturing process variations present between the two ROs. A decoder is introduced to generate all the internal signals for a specific mode. When $EN = 0$, both ROs oscillate while the sleep transistors are ON. The signals EN and SRO_EN can never be "1" simultaneously as they would create a short circuit in the design. Similar to the design described in Figure 3.1, the NBTI-aware RO-CDIR also has a MUX, a counter, and a timer to select the ROs and measure their frequencies during authentication. Also, sleep transistors are used to connect the ROs to the power supply in the RO-based sensor as before. PMOS sleep transistors control the connection between VDD and the inverters and NMOS sleep transistors control the connection between VSS and the inverters.

Table 3.1 highlights the four distinct modes of operation. In the manufacturing and burn-in tests, our objective is to protect both ROs from aging. In this mode, both ROs enter sleep mode by being cut off from the power and ground lines. R_SLEEP and S_SLEEP are assigned to "0" during this entire operation. In normal operation, the reference RO remains in the sleep mode while the stressed RO is in the stressed mode. All the inverters in the stressed RO are given a DC stress by pulling their inputs to ground. In authentication mode, the reference RO is activated to measure its frequency (RO_SEL to 0), which corresponds to the RO frequency of a new IC. Then, the stressed RO is activated (SRO_EN to 0 and EN to 1) and its frequency (RO_SEL to 1) is measured.

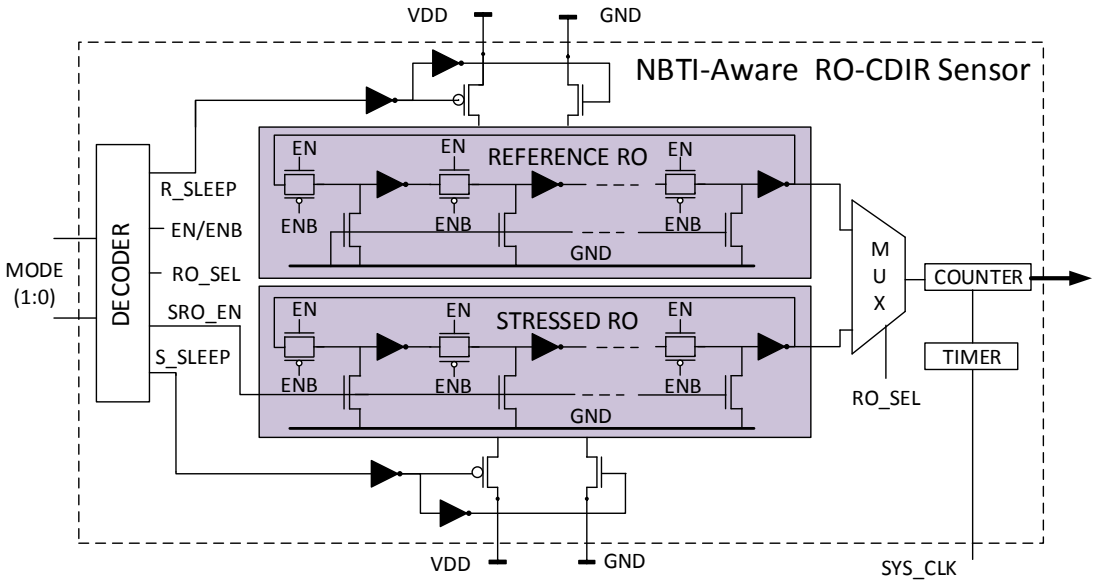


Fig. 3.3: The proposed NBTI-Aware RO-CDIR sensor.

Table 3.1: Modes of operation.

MODE	Signals					Description
	R_SLEEP	*EN*	*RO_SEL*	*SRO_EN*	*S_SLEEP*	
00	0	X	X	X	0	Manufacturing and Burn-In Tests: Both ROs are in sleep mode.
01	0	0	X	1	1	Normal Operation: Reference RO in sleep mode and Stressed RO in stressed mode (inverter input GND)
10	1	1	0	0	1	Authentication Mode: Measure frequency of Reference RO
11	0	1	1	0	1	Authentication Mode: Measure frequency of Stressed RO

3.1.4 Δf distribution versus ROs stages

Let us consider two n-stage reference and stressed ROs. The frequency of an RO becomes: $f = \frac{1}{2\sum t_{di}}$, where t_{di} is the delay for the i^{th} stage. We can express t_{di} as $t_{di} = t_{d0} + \Delta_i$, where t_{d0}

is the fixed delay for all the inverters and Δ_i is the variable delay due to process variation. Thus, the frequency becomes:

$$f = \frac{1}{2nt_{d0} + 2\sum \Delta_i}$$

$$\begin{aligned} \text{Now, } \Delta f &= f_R - f_S \\ &= \frac{1}{2nt_{d0} + 2\sum_R \Delta_i} - \frac{1}{2nt_{d0} + 2\sum_S \Delta_i} \\ &= \frac{(\sum_S \Delta_i - \sum_R \Delta_i)}{(nt_{d0} + \sum_R \Delta_i)(nt_{d0} + \sum_S \Delta_i)} \end{aligned} \tag{3.1}$$

From Equation 3.1, it can be inferred that Δf $(f_R - f_S)$ tends to be near the mean (0) of Δf distribution as n increases due to the numerator increases at the order of n, whereas the denominator increases at the order of n^2. This results in the reduction of the spread of Δf distribution for a 51-stage RO and thus increase in the accuracy.

3.1.5 Registration and Authentication Flow

Figure 3.4 shows the registration and authentication flow. The objective of the registration process is to determine a threshold (Δf_{th}), which will be used during the authentication process. Here, $\Delta f = f_R - f_S$ is the frequency difference between reference RO (f_R) and stressed RO (f_S). Δf of an IC are measured during authentication. If Δf of an IC is greater than Δf_{th}, then the IC will be treated as recycled, otherwise, it will be marked as new.

During registration phase, a number of new ICs are used to generate the distributions to determine the threshold (Δf_{th}) after the manufacturing test process at the foundry. It is recommended to select the samples from different wafers and lots to capture within-die and within-wafer process variations. The larger this sample space is, the more accurate the Δf_{th} will be. In the following, we will describe the Δf_{th} determination process.

Due to the process variations, the difference between the reference and stressed RO frequencies (Δf) will not be zero even though we place these ROs very close to each other in the circuit layout. We observe a Gaussian distribution of Δf when we perform a Monte Carlo

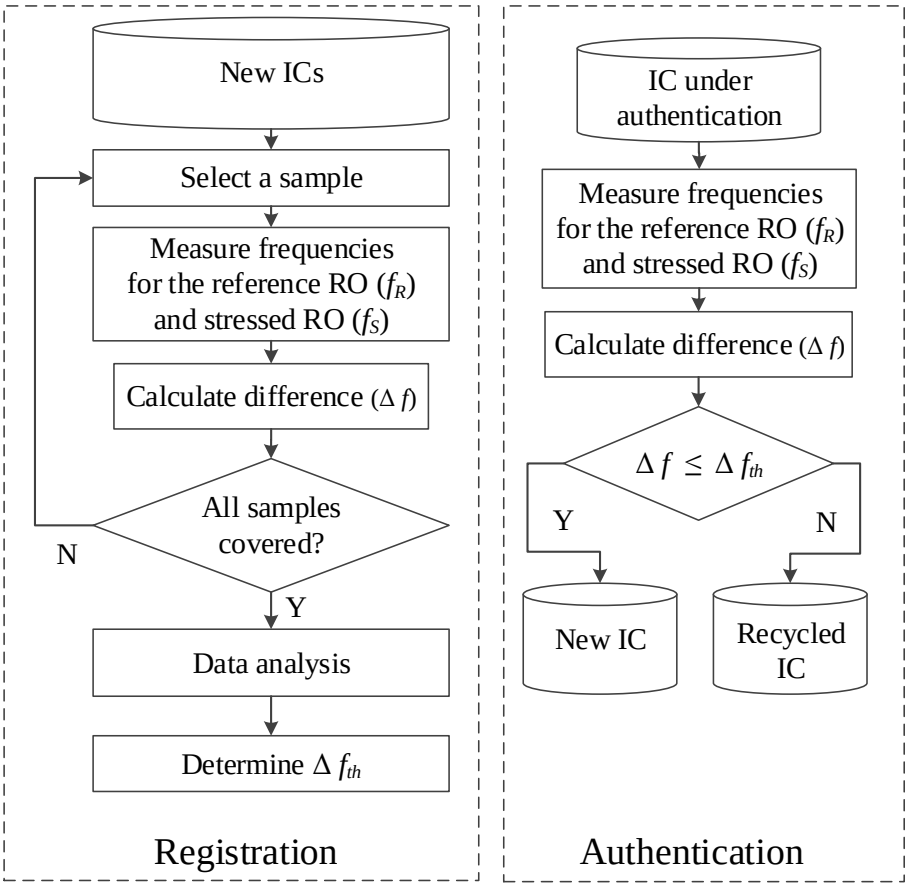


Fig. 3.4: Registration and authentication flow for N-CDIR.

simulation with 1000 samples (see Section 3.1.7). Figure 3.5 shows a simplified representation of the two distributions (probability density functions) of Δf at times 0 and t ($g_0(\Delta f)$ and $g_t(\Delta f)$). The x-axis represents the frequency differences between the two ROs (Δf) and the y-axis represents the corresponding distribution function. The overlapping area represents the misprediction of identifying new or recycled ICs. The red area represents the probability of identifying recycled ICs as new whereas the green area denotes the probability of identifying new ICs as recycled. These areas (θ_1 and θ_2) are represented by:

$$\theta_1 = \int_{-\infty}^{\Delta f_{th}} g_t(\Delta f)\, d\Delta f, \;\; and \;\; \theta_2 = \int_{\Delta f_{th}}^{\infty} g_0(\Delta f)\, d\Delta f \tag{3.2}$$

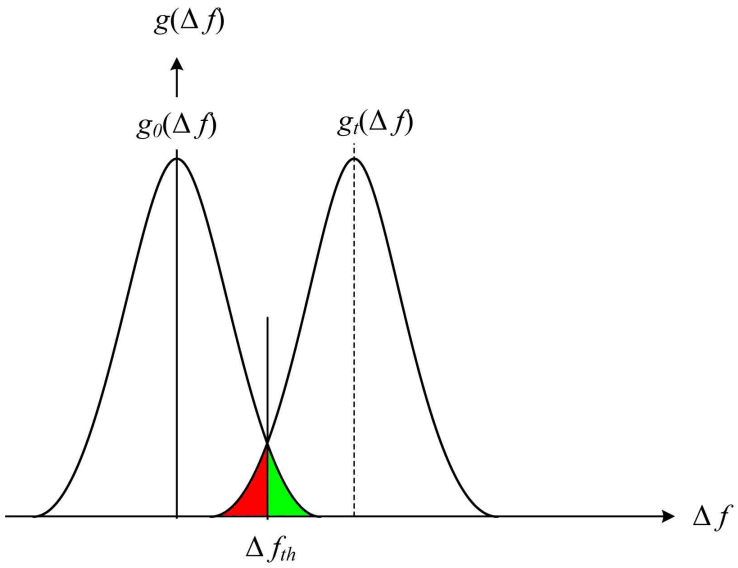


Fig. 3.5: Probability density function of frequency differences (Δf) between reference and stressed ROs.

where, $g_0(\Delta f)$, and $g_t(\Delta f)$ correspond to the distribution of frequency differences for new and ICs with t amount of usage, respectively. The decision threshold should be the point (Δf_{th}) where both distributions intersect each other. This represents the frequency difference that minimizes the total probability of error ($\theta_1 + \theta_2$).

3.1.6 Overhead Analysis

The area overhead of both the RO-CDIRs is negligible for modern designs. The area overhead mostly comes from the size of the counter and timer. The area of the remaining parts is negligible. Thus both the original and NBTI-aware designs offer similar area overhead. We can also remove the timer and counter from the RO-CDIRs and measure the frequencies off-chip making the area overhead even smaller.

Table 4.4 shows the area overhead analysis of the RO-CDIRs. We define area overhead as the ratio of the size (area) of the RO-CDIR with the size (area) of the benchmark. Here, the IWLS 2005 benchmarks are arranged from low to high sizes to compute the area overhead. The timer and counter are excluded during the computation, as we assume the frequency measure-

ment can be performed off-chip. As seen, the overhead is more than 1% for small benchmarks (*i2c, spi, and b14*) for 51-stage NBTI-Aware RO-CDIR that could make it challenging to use them in small designs. The area overhead for the 51-stage RO-CDIR is less than the 51-stage NBTI-Aware RO-CDIR. The area overhead is comparably lower for the 21-stage RO-CDIRs. For large designs, however, it hardly impacts the overall area overhead.

Table 3.2: Area overhead analysis of RO-CDIRs.

Benchmark	**Size (# Gates)**	**Area Overhead**			
		Simple 21-stage RO-CDIR	**NBTI-Aware 21-stage RO-CDIR**	**Simple 51-stage RO-CDIR**	**NBTI-Aware 51-stage RO-CDIR**
i2c	1124	2.89 %	4.73 %	5.52 %	9.98 %
spi	3277	1.01 %	1.65 %	1.92 %	3.48 %
b14	8679	0.38 %	0.62 %	0.73 %	1.31 %
b15	12562	0.26 %	0.43 %	0.50 %	0.91 %
DMA	19118	0.17 %	0.28 %	0.33 %	0.6 %
DSP	32436	0.10 %	0.17 %	0.19 %	0.35 %
ethernet	46771	0.07 %	0.115 %	0.135 %	0.244 %
vga_lcd	124031	0.03 %	0.044 %	0.051 %	0.092 %
leon2	780456	0.004 %	0.007 %	0.008 %	0.015 %

The power consumption of the NBTI-Aware RO-CDIR is lower compared to the simple RO-CDIR, as there is no switching during the normal operation due to the fact that all inputs of the inverters in the stressed RO are pulled down to ground. However, both of them provide negligible power overhead when they are placed in modern industrial designs. As shown in Figure 1.12, RO-CDIRs are suitable for large digital ICs such as microprocessors, microcontrollers, digital signal processors, ASICs, programmable logic devices, and memories. Such sensors can

also be used in smaller digital ICs if the area overhead is acceptable.

3.1.7 Simulation of the NBTI-Aware RO-CDIR

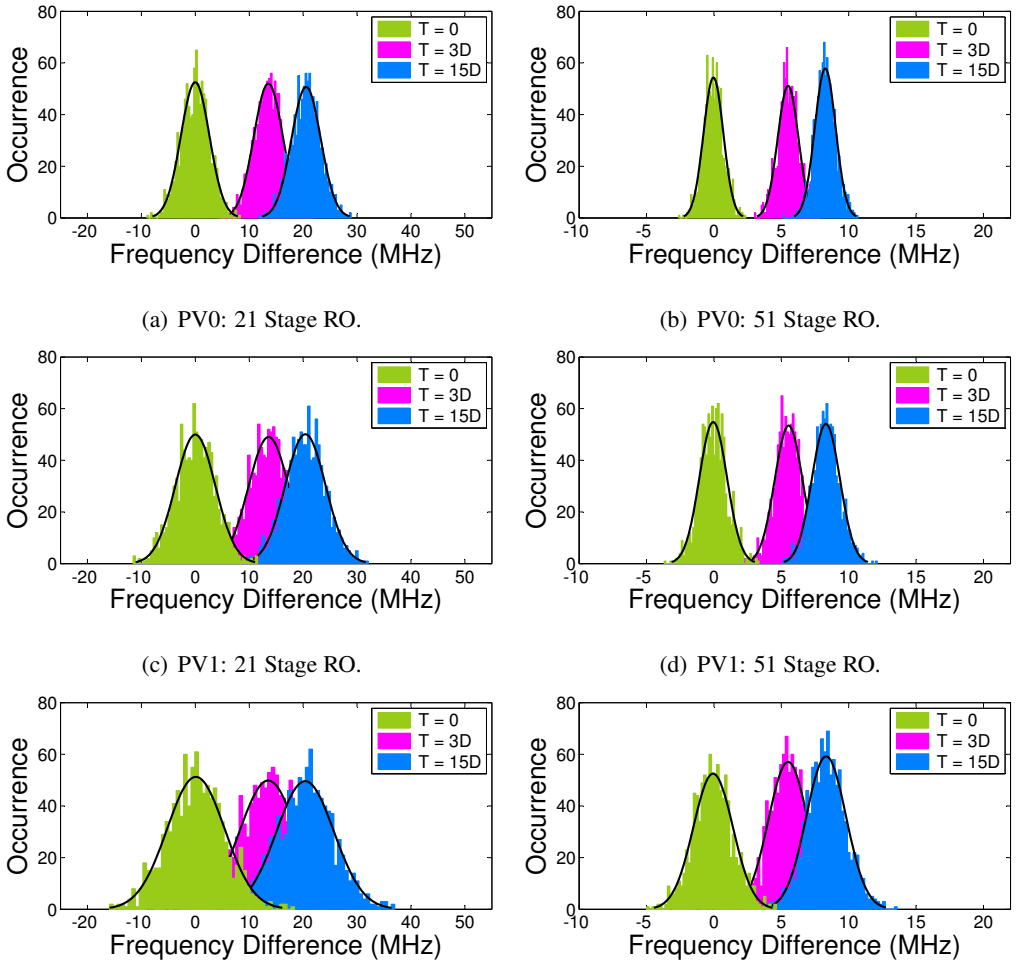
In order to verify the effectiveness of the NBTI-Aware RO-CDIR, the design is implemented and simulated using the 90nm technology node [91]. HSPICE MOSRA from Synopsys is used to simulate and measure the impact of aging on this RO-CDIR. The nominal supply voltage is 1.2V. In this simulation, we select 21-stage and 51-stage ROs to compare the results. To model the variation, Monte Carlo (MC) simulation is performed with 1000 samples of the NBTI-Aware RO-CDIR in HSPICE. Here, we were mostly concerned with detecting ICs used in the field for a very short period of time, so we set, the total aging time at 15 days in the increment of 3 days. Larger usage times would be easily detected using this sensor.

Table 3.3: Process variations.

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%

Three different process variations are considered to investigate the impact of variation on the detection of recycled ICs. Table 3.3 shows the different process variations used in the simulation. Moving from PV0 to PV2, inter-die and intra-die variations both become larger. That is because, as feature size decreases and die size increases, the complex semiconductor manufacturing processes cause variations to the device parameters significantly. However, we acknowledge that the impact of process variation on ROs will be minimal as they are placed physically near to each other. PV0 represents the expected process variation between ROs while the other two are the worst-case scenarios.

Figure 3.6 shows the simulation results of our proposed NBTI-Aware RO-CDIR sensor. The x-axis represents the frequency difference (Δf) between the reference RO and stressed RO.



(a) PV0: 21 Stage RO.

(b) PV0: 51 Stage RO.

(c) PV1: 21 Stage RO.

(d) PV1: 51 Stage RO.

(e) PV2: 21 Stage RO.

(f) PV2: 51 Stage RO.

Fig. 3.6: The distribution of frequency differences between the reference RO and the stressed RO with different process variations, PV0, PV1, and PV2.

The y-axis represents the frequency of occurrence (i.e., # of Monte Carlo samples). The legend in the figures denotes the aging time (for example, T = 3D denotes the RO-CDIR is aged for 3 days). The green (0D aging) distribution for Δf is centered at 0Hz while the pink and blue (3D and 15D aging) distributions shift to the right. This is because the stressed RO has aged and become slower resulting in larger Δf.

We can clearly identify recycled ICs when the two distributions (T = 0 and T = 3D/15D) do not overlap with each other. The percentage of misprediction (new ICs detected as counterfeit and vice versa) can be estimated as the area of overlap between these two distributions.

We apply Gaussian fit to find the mean and variance of the distributions and then calculate the overlapped area. We can certainly identify recycled ICs with aging more than 15 days in almost all cases. Based on the figure, we expect a higher misprediction rate (i) as the process variations increase and (ii) when the 21-stage RO is used rather than the 51-stage RO (see Appendix B). As process variations increase, the variance in Δf grows resulting in larger overlap between 0D and 3D/15D distributions. Similarly, since the 21-stage RO distributions have a larger spread than the 51-stage RO (see Section 3.1.4 for the proof), we should also expect higher misprediction rate. The best case scenario occurs for the 51-stage RO with PV0 where we can detect recycled ICs in 3 days with negligible penalty of misprediction. *This represents a substantial improvement over the prior work [1] which required at least one month of aging to identify recycled ICs*. As we described in Section 3.1, the design in [1] only ages 50% of inverters in each oscillation cycle while the other half of inverters recover. This results in a slower aging of the stressed RO. In contrast, our proposed design ages all the inverters in the stressed RO constantly (without recovering) during normal operation. Thus, we expect higher aging for the stressed RO which allows our NBTI-aware RO-CDIR to detect recycled ICs used much less than one month (as little as 3 days).

3.1.8 Misprediction Rate Analysis

In order to find the effectiveness of RO-CDIR, we present the misprediction rate analysis. We define misprediction rate as recycled ICs identified as new (θ_1), and new ICs identified as recycled (θ_2). Here we will only present the results for NBTI-Aware RO-CDIR.

Table 3.4: Misprediction Rate.

	θ_1 (%)						**θ_2 (%)**					
	3 Days			**15 Days**			**3 Days**			**15 Days**		
	PV0	**PV1**	**PV2**	**PV0**	**PV1**	**PV2**	**PV0**	**PV1**	**PV2**	**PV0**	**PV1**	**PV2**
21-stage RO	0.6	3.53	10.19	0	0.31	2.84	0.45	3.16	10.54	0	0.25	2.87
51-stage RO	0	0.32	2.79	0	0	0.21	0	0.3	2.85	0	0	0.18

Table 3.4 shows the misprediction rate i.e., recycled ICs identified as new (θ_1) and new

ICs identified as recycled (θ_2) for 21-stage and 51-stage NBTI-Aware CDIR sensors, with process variations mentioned in Table 3.3. The rate is higher in PV2 as stressed and reference ROs frequencies differ significantly between two samples due to higher process variations. This results in a higher overlapped area between two distributions. However, we obtain significantly lower θ for 51-stage RO. θ_1 is 2.79% and 0.21% when the NBTI-Aware RO-CDIR has aged 3 days and 15, respectively. For PV1, it is 0.32% and 0% for same use times. We can predict all the samples as recycled or new when they are aged only 3 days. As we described earlier, with these two ROs placed very close to each other, the variation will be well below PV1. Under different cases, we also observe similar misprediction rate (θ_2) of identifying new ICs as recycled. In both these cases, 51-stage RO outperforms the 21-stage RO.

In the simulation, we have only considered process variation. We did not include any results for temperature and power supply variation. As the two ROs are placed very close in the circuit layout and the temperature variation is a global phenomenon, the temperature variation between the two is practically negligible ($\triangle T = 0$). At higher temperatures, we would also expect more rapid aging in the stressed RO, which should only improve our results. A similar argument can be made for power supply variation.

3.1.9 Workload Analysis

It is also important to analyze different workloads that impact the detection of recycled ICs. We define workload as the percentage of time per day that the IC is in use. The workload/usage depends on the type of application being run. For example, the ICs used in – (i) mobile phones may remain on during the entire day (workload may be 100%), or, (ii) televisions or laptops may be ON for a fraction of day (workload may be well below 100%). *We have considered 100% workload for all the simulations unless specified otherwise.* Table 3.5 shows the minimum usage time of ICs under various workloads required for proper identification. Note we have shown the results only for the 51-stage NBTI-Aware RO-CDIR, as it provides minimum misprediction. The results show that the length of time required to detect the recycled IC increases as the workload decreases. For example, a workload of 10% and 1% requires the IC be used for 30 days and 300 days respectively. With reduced workload, we can only identify ICs as recycled if the system is used over a longer period of time because when the system is off (i.e., not in

use), time passes, but the stressed RO does not age at all. Note that the impact of low-workload environment would be similar for all prior approaches based on aging [85] [86] [1]. Hence, the NBTI-aware RO-CDIR will outperform all other aging-based methods at any workload.

Table 3.5: Workload analysis.

	Workload				
	100%	**75 %**	**50 %**	**10 %**	**1 %**
51-Stage RO-CDIR	3 Days	4 Days	6 Days	30 Days	300 Days

3.1.10 Attack Analysis

As we all know, counterfeiting is an evolving problem. The counterfeiters are continuously improving their techniques through experience. We believe that this trend will continue and the counterfeiters will continue to evolve and adapt their techniques to new detection and protection methods. Thus, it is of the utmost importance to analyze all of the possible attacks on these RO-CDIRs and their vulnerabilities in order to examine their robustness. There may be two types of attacks possible on RO-CDIRs, and they are as follows:

- *Removal/Tampering*: The first attack on RO-CDIRs could be removal/tampering attacks. However, it is fairly impossible for the counterfeiter to replace the stressed RO with a new one or to tamper with the stressed/reference RO in order to match their frequency. If we assume that a removal or tampering attack is possible, then the counterfeiter must remove the old package and then again repackage and remark it according to its original specifications. This removal and then repackaging may not be cost effective to the counterfeiters. Hence, it is unlikely to be used in practice.

- *Age Reference RO*: In this attack scenario, the counterfeiter may try to intentionally age the Reference RO to mask the difference between the ROs. The counterfeiter might attempt to force the RO-CDIR to work in authentication mode (MODE 10, in Table 3.1) for a period of time under accelerated stress conditions. With the accelerated aging at

the same time, the frequency difference between the Stressed RO and the Reference RO would shrink since both of them could asymptotically approach maximum degradation.

As we all know, burn-in is a very expensive process and the counterfeiter must have an expensive setup for that. The primary incentive for counterfeiting is cheap recycling, not adding extra cost to the components. There might not be any motivation left for the counterfeiters when they are forced to add burn-in to their recycling process. As a result, this attack might not be feasible as there is no cost incentive.

3.2 Fuse-Based CDIR

The RO-CDIR structures describe above, are most suitable for large digital ICs due to the area required to implement them. However, the majority of components on the market today are smaller analog, digital, and mixed-signal types. In this section, we are presenting an alternative, low-cost structure that is based on semiconductor fuses [92] [93] and can be implemented into almost any design, with the exception of discrete components, such as diodes, transistors, and passive components. This structure can be fabricated along with the original design, and it does not require the modification or addition of any steps to the manufacturing process.

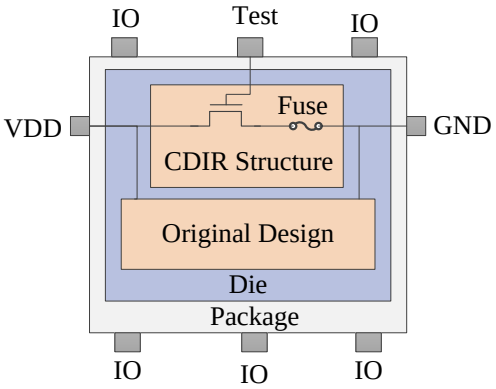


Fig. 3.7: F-CDIR: version I.

Figure 3.7 presents the design for the fuse-based CDIR structure. The structure consists of a switch and a fuse. It is a three-terminal structure, having two terminals that are connected to VDD and GND pins. The third terminal, the control terminal, is regulated by *Test* pin on the

IC. In this design, the MOSFET acts as the switch. The design overhead is only one transistor and a fuse. The design works as follows: During the manufacturing and burn-in test modes *Test* pin will always be "0" which will provide no current flow through this structure. When the component is placed in the printed circuit board (PCB) for normal operation, *Test* pin will be connected to VDD. The MOS will be ON and a current will flow through the fuse, which will result in an open circuit inside the structure. The device will then operate normally.

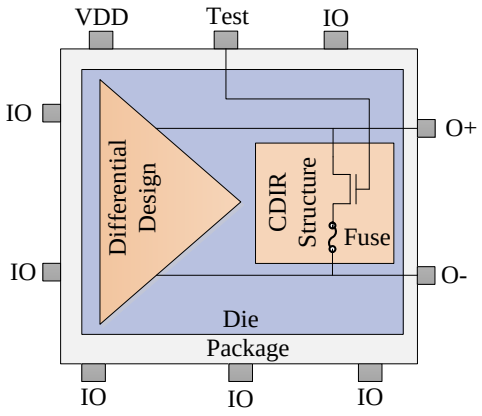


Fig. 3.8: F-CDIR version I implemented in differential designs.

The detection of counterfeit (used in the field) components will be the measurement of resistance between VDD and GND pins while setting *Test* pin to VDD. The measured resistance between VDD and GND should be negligible for new component. If the component has been used in the field, the measured resistance will be high (infinite). Here we are assuming the users of the component are trusted and they design the PCB with *Test* treated as VDD. For the added security, the *Test* pin can be named as VDD.

Figure 3.8 shows the implementation of this structure in differential designs. The structure is placed in between the differential output, $O+$ and $O-$, pins. The control pin is connected to the *Test* pin. For the proper burning of the fuse, the differential design must provide the necessary current to the fuse. During the manufacturing and burn-in tests mode, *Test* pin will be assigned to "0" which makes the MOS off and the fuse remains intact. When the device operates in field for the first time, the fuse will be burnt because of a current flowing through it. The design will then operate according to its normal specifications. The measured resistance

between $O+$ and $O-$ should be negligible for new components, and it will be high (infinite) for counterfeit components.

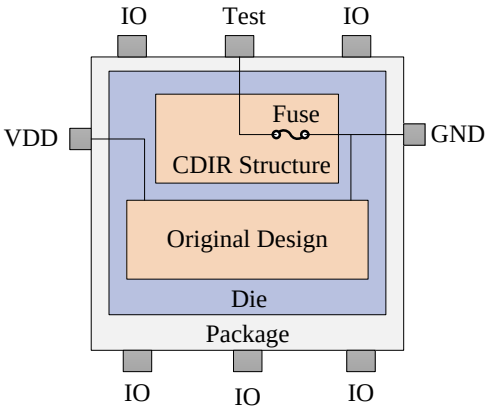


Fig. 3.9: F-CDIR: version II.

Figure 3.9 presents a different version of the CDIR structure. The design consists of only one semiconductor fuse. The terminals of the sensor are connected to *Test* and *GND* pins. The fuse is isolated from the rest of the design. During the manufacturing and burn-in tests mode, the *Test* pin will always be "0". The fuse will be intact during these modes, as there is no current flowing through it. In normal operation, this pin will be assigned to VDD. When the chip operates in the field for the first time, the current will flow through the sensor and the fuse will be burnt. The detection of used components will be based on measuring the resistance value between the *Test* and *GND* pins. A simple multimeter can authenticate the components. A component will be treated as counterfeit if this measured resistance value is high (infinite) and new if this value is low.

Note that the successful implementation of F-CDIR relies on the trusted system integrator as the burning of the fuse will only be asserted if VDD gets applied to the Test pin. If the system integrator does not apply VDD to the *Test* pin in their systems, the fuse will then be intact. In that case, we cannot identify recycled ICs by simply measuring the resistance.

3.2.1 Area overhead analysis

Table 3.6 shows the approximate area overhead of the F-CDIRs. We have selected ITC'97 benchmark [94] for analog and mixed signal circuits. We have calculated the approximate area overhead by the ratio of components used in the F-CDIR with benchmark circuits. For small analog circuits, the overhead is about 20% for the F-CDIR Version I, whereas it is considerably lower for the F-CDIR Version II. The F-CDIR II consists of only one component (fuse) compared to two components (a fuse and a transistor) for the F-CDIR I. As mentioned earlier, both F-CDIR structures also require one extra test pin. This might prohibit their use in cases where the number of IO pins are limited as they are in smaller ICs. As for digital circuits (such as the benchmarks used in Table 4.4), F-CDIR structures require virtually negligible area overhead and the one extra pin may not be an issue either.

Table 3.6: Area overhead of F-CDIR.

Benchmark	Components	Area Overhead	
		F-CDIR I	F-CDIR II
Operational Amplifier #1	11	18.18 %	9.09%
Continuous-Time State-Variable Filter	42	4.76 %	2.38 %
Operational Amplifier #2	10	20.00 %	10.00 %
Leapfrog Filter	77	2.60 %	1.30 %
Digital-to-Analog Converter	44	4.54 %	2.27 %

3.2.2 Attack Analysis

The design for the F-CDIRs is the simplest among the all three CDIRs, as it consists of only one fuse (F-CDIR version II) or one fuse and one transistor (F-CDIR version I). However, this design is also resistant to tampering. The possible attacks are as follows:

- *Trust on System Integrator*: For the proper operation of the F-CDIR, burning the fuse is necessary, and this can only be done when VDD gets applied to the *Test* pin. Thus the

successful implementation of the F-CDIR relies on the trusted system integrator.

- *Tampering*: The state of the fuse could be modified. However, a separate metal deposition is necessary to make the fuse. This would require the decapsulation of the package and then metal deposition. This is indeed a very costly process. Thus there should not be any cost incentive for the counterfeiters to perform this process for every IC. The counterfeiters would not get any benefit, as these structures would be placed in very low-cost analog and mixed-signal ICs.

3.3 Summary

In this chapter, we have presented two DFAC structures, RO-CDIRs and F-CDIRs, to detect recycled and remarked ICs of different types and sizes. The NBTI-Aware RO-CDIR structure can be implemented in any digital IC with new technology nodes as it takes the advantage of higher aging in newer technology nodes. It can be placed even in smaller digital ICs with few thousand gates, due to the low area overhead. The simple RO-CDIR requires three test pins whereas the NBTI-Aware RO-CDIR needs two additional pins while also achieving better performance. F-CDIRs can be implemented in any components (small, or large, and analog, or digital) and any technology node. These CDIRs can authenticate ICs very effectively and require a very low cost multimeter. F-CDIRs require only one test pin for IC authentication. Finally, all these CDIRs are resistant to all types of known attacks. Together, these structures provide excellent coverage for the full range of recycled ICs.

Chapter 4

Combating Die and IC Recycling with Multiple RO-Pairs

The NBTI-aware CDIR (N-CDIR) proposed in Chapter 3, can efficiently detect recycled ICs with little misprediction when the chips are aged for a short period of time. However, when the workload decreases (i.e., the chip is used less frequently in case of mobile and automotive applications), we require the chips to be used much longer for detection, which eventually results in a higher rate of misprediction. In addition, there may be a recycling activity from the overstock of electronic systems where the recyclers extract components from never used systems. The detection of these components can be performed with a CDIR that can detect a small amount of aging (for example, the amount of aging caused during the test of a system). When the application risk is critical [9] [12] [13], we do not have the luxury for any test escapes as the system failure due to using recycled chips could cause significant financial loss, as well as risks to safety and security. Thus, this necessitates further improvements of the N-CDIR.

To address these challenges, we present two different CDIR structures based on multiple ring oscillators [16]. We propose a new N-CDIR with multiple reference and stressed RO-pairs. We introduce an averaging approach to reduce the impact of process variations during the estimation of a threshold which will be used in the authentication process (see Figure 3.4). We call this design multiple-pair NBTI-aware RO CDIR with averaging (AN-CDIR). This design provides a much better detection for ICs used for only few hours in the field with the cost of small misprediction. In addition, we propose another modified design of N-CDIR by adding multiple NBTI-aware reference and stressed RO-pairs like AN-CDIR. However, in this case we propose a selection algorithm (see Figure 4.5) to find the best reference and stressed RO-pair. We call this design multiple-pair NBTI-aware RO CDIR with selection (SN-CDIR). This CDIR with the best selected RO-pair, provides even better detection (no misprediction) of recycled

ICs, even if they have been used only for a few hours, unlike the AN-CDIR.

4.1 CDIR with Multiple RO-pairs

Figure 4.1 shows the process of reducing the rate of misprediction for identifying a recycled IC as new and vice versa. Misprediction arises from the overlap of the reference and stressed ROs frequency difference distribution at time 0 ($g_0(\Delta f)$) and the distribution at time t ($g_t(\Delta f)$), which is the aged replica of $g_0(\Delta f)$. This overlapping area can be reduced by:

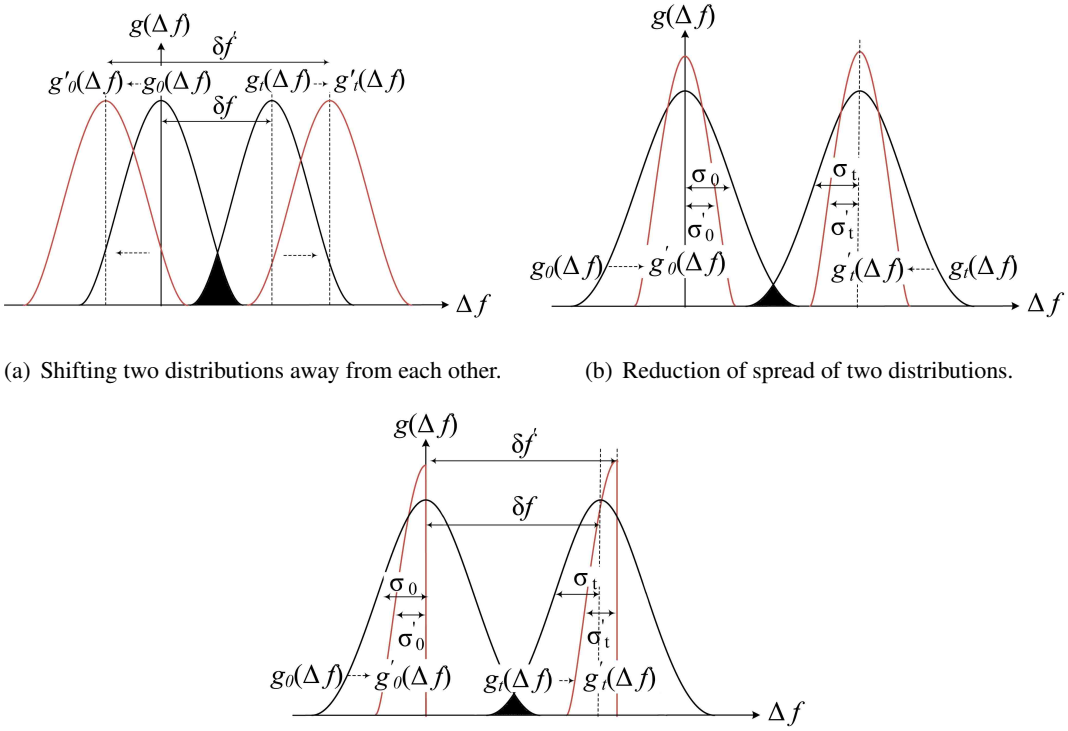
i) Increasing the separation of these two distributions. This separation (aging degradation, δf) can be increased by shifting the distribution $g_0(\Delta f)$ to the left ($g_0^{'}(\Delta f)$) or shifting the distribution $g_t(\Delta f)$ to the right $g_t^{'}(\Delta f)$, or by doing both simultaneously (see Figure 4.1(a)). Our proposed N-CDIR provides better detection of recycled ICs by shifting the distribution $g_t(\Delta f)$ to the right as compared to O-CDIR.

ii) Reducing the spread of these two distributions. This spread results from their variances (σ_0^2 and σ_t^2). Figure 4.1(b) shows no overlap between $g_0^{'}(\Delta f)$ and $g_t^{'}(\Delta f)$ ($\sigma_0^{'} < \sigma_0$ and $\sigma_t^{'} < \sigma_t$). Our proposed AN-CDIR utilizes this technique to reduce misprediction rate.

iii) Simultaneously reducing the spread and increasing the separation of two distributions. Figure 4.1(c) shows such case. The overall spread can be reduced by discarding the right hand side, and reducing the left hand side spread of $g_0(\Delta f)$. The separation can be increased by shifting $g_t(\Delta f)$ to the right. Our proposed SN-CDIR utilizes this technique and provides the best detection of recycled ICs.

By introducing multiple RO-pairs in a CDIR, it becomes possible to achieve 1, 2, and/or 3, thereby reducing misprediction when the ICs are used only for a short period of time. In the following, we will describe two different architectures utilizing multiple RO-pairs to minimize the misprediction. It is also analytically proven that both approaches are better than the single N-CDIR. In Section 5.4, simulations for all sensors support our conclusions.

To eliminate the confusion among different symbols, which will be introduced shortly, we summarize the notations in Table 4.1.



(a) Shifting two distributions away from each other.

(b) Reduction of spread of two distributions.

(c) Reduction of spread and shifting of mean of two distributions.

Fig. 4.1: Reduction of misprediction (overlapped area).

4.2 CDIR Sensor with multiple RO-pairs and Averaging Approach (AN-CDIR)

The AN-CDIR works based on the averaging of reference and stressed RO frequencies. In the following, we will provide proof that the spread (σ) of $g_0(\Delta f)$ (see Figure 4.1(b)) is reduced significantly after averaging. In this method, one must measure all the frequencies of the stressed and reference ROs consecutively and then take the average of the reference RO and stressed RO frequencies.

4.2.1 Averaging to Reduce Spread

Let us assume that there are n ROs present in the reference and stressed block of an AN-CDIR. We treat the frequencies of the reference ROs and stressed ROs as random variables, denoted by $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}$, and $\mathbf{Y_1}, \mathbf{Y_2}, \ldots, \mathbf{Y_n}$, respectively.

Table 4.1: Notations and their descriptions.

Notation[1–3]	**Equation**	**Description**
Δf	$\Delta f = f_R - f_S$	Δf is the frequency difference between reference RO (f_R) and stressed RO (f_S).
δf	$\begin{aligned}\delta f &= \Delta f_t - \Delta f_0 \\ &= (f_{tR} - f_{tS}) - \\ &\quad (f_{0R} - f_{0S}) \\ &= -(f_{0R} - f_{tR}) + \\ &\quad (f_{0S} - f_{tS}) \\ &= -\delta f_R + \delta f_S\end{aligned}$	δf is the aging degradation. Δf_0 and Δf_t are the frequency differences at time 0 and t. f_{0R} and f_{0S} are the frequencies of the reference and stressed ROs at time 0. f_{tR} and f_{tS} are the frequencies of the reference and stressed ROs at time t. δf_R and δf_S are the aging degradation of reference and stressed ROs.
∂f_S	$\partial f_S = \frac{f_{0S,V_{DD1}} - f_{0S,V_{DD2}}}{f_{0S,V_{DD2}}}$	∂f_S is the percentage frequency difference of the stressed RO with two different supply voltages ($V_{DD1} > V_{DD2}$) at time 0. $f_{0S,V_{DD1}}$ and $f_{0S,V_{DD2}}$ are the frequencies at supply voltages V_{DD1} and V_{DD2}, respectively.

[1] $\hat{}$ denotes minimum mean square error (MMSE) estimator [95].

[2] Boldface symbol denotes random variables.

[3] $\vec{}$ denotes vectors.

As the distribution of $g_0(\Delta f)$ is formed by the frequency differences of the reference and stressed RO frequencies, we construct the following random variables,

$$\mathbf{Z_i} = \mathbf{X_i} - \mathbf{Y_i}$$

$\mathbf{Z_i}$s are Gaussian as all $\mathbf{X_i}$s and $\mathbf{Y_i}$s are Gaussian. We also assume that these newly formed variables have the same mean (μ) and variance (σ^2), as all the ROs experience the same process variations.

The objective is to find the mean and variance of a newly formed random variable $\mathbf{W_n}$, where

$$\mathbf{W}_n = \frac{1}{n}\sum_{i=1}^{n}\mathbf{X_i} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{Y_i} \tag{4.1}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X_i} - \mathbf{Y_i}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{Z_i} \tag{4.2}$$

As all $\mathbf{Z_i}$s are Gaussian, the resultant random variable $\mathbf{W_n}$ will also be Gaussian and its statistics will be completely determined by the mean and variance, which can be formulated as:

$$E[\mathbf{W}_n] = E\Big[\frac{1}{n}\sum_{i=1}^{n}\mathbf{Z_i}\Big] = \frac{1}{n}\Big(E\Big[\sum_{i=1}^{n}\mathbf{Z_i}\Big]\Big)$$

$$= \frac{n\times\mu}{n} = \mu \tag{4.3}$$

$$var(\mathbf{W}_n) = var\Big(\frac{1}{n}\sum_{i=1}^{n}\mathbf{Z_i}\Big) = var\Big(\sum_{i=1}^{n}\frac{\mathbf{Z_i}}{n}\Big)$$

$$= \frac{1}{n^2}\sum_{i=1}^{n}var(\mathbf{Z_i}) + \frac{1}{n^2}\sum_{i\neq j}cov(\mathbf{Z_i},\mathbf{Z_j}) \tag{4.4}$$

In the above, $E[A]$ denotes the expected value of random variable A which is equivalent to the mean for a Gaussian random variable; $var(A)$ denotes the variance of random variable A; and $cov(A,B)$ denotes the covariance between random variables A and B. Let us assume that the frequencies of all the ROs are independent. This results in $\mathbf{Z_1},\mathbf{Z_2},\ldots,\mathbf{Z_n}$ being independent. Then all of the covariances in Equation 4.4 are zero.

$$var(\mathbf{W}_n) = \frac{1}{n^2}\sum_{i=1}^{n}var(\mathbf{Z_i}) = \frac{n\times\sigma^2}{n^2} = \frac{\sigma^2}{n} \tag{4.5}$$

Thus, the mean (μ) and standard deviation (σ) of $\mathbf{W_n}$ becomes:

$$\mu_{W_n} = \mu \tag{4.6}$$

$$\sigma_{W_n} = \frac{\sigma}{\sqrt{n}} \tag{4.7}$$

In Equations 4.3 and 4.6, the mean of the average difference W_n is unchanged when compared to each Z_i. However, the variance (spread) of W_n is a factor on $\sqrt{n}$ smaller (see Equation 4.7). A similar treatment can be performed for the distribution $g_t(.)$ at time t to estimate the resultant mean and variance. *This implies that the overlap between the two distributions can be made negligibly small by adding additional RO-pairs (as shown in Figure 4.1(b)).*

4.2.2 Architecture of AN-CDIR

Figure 4.2 shows our proposed architecture for the AN-CDIR. It consists of a reference RO block and a stressed RO block. Each block again consists of equal numbers of NBTI-aware reference and stressed ROs. The number of ROs in each block depends on the detection of recycled ICs used for a minimum amount of time (we label this time as "resolution"). For example, our results show that by placing four ROs in each block, we can detect recycled ICs that have been used for only one day with 100% workload (see Table 4.3). Larger numbers of ROs are necessary to achieve a superfine recycled IC detection resolution. The requisite number of ROs can be determined based on the available area on the chip.

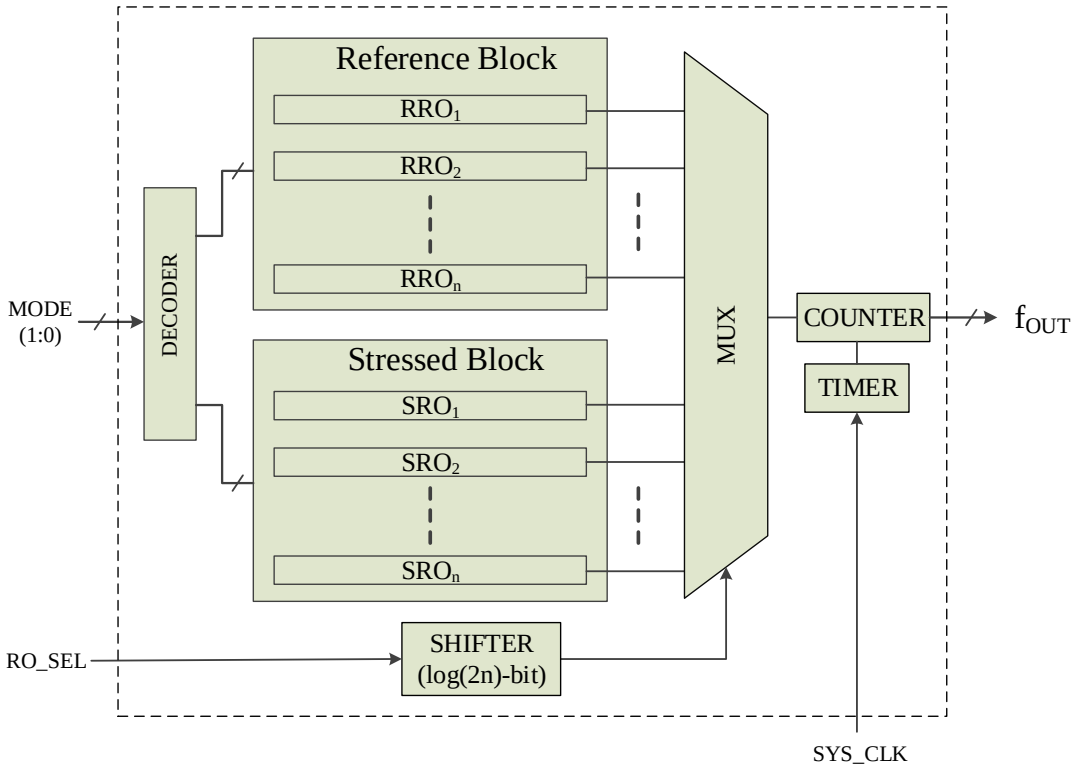


Fig. 4.2: The architecture of our proposed AN-CDIR.

All the ROs in the reference block and stressed blocks are fed to a multiplexer (MUX). The selection input of MUX is provided by a shift register of $log_2(2n)$ bit to minimize the I/O pin count for this CDIR. This register is loaded through a serial in RO_SEL pin. The rest of the design is similar to the N-CDIR described in Figure 3.3. The $DECODER$ generates all the

internal signals (see Table 3.1) for the reference and stressed RO blocks. It is not necessary to generate the control signals for each RO in the reference and stressed RO blocks. All the ROs in each block utilize the same internal signals generated by the *DECODER*. The *COUNTER* and *TIMER* operate as described in N-CDIR.

The registration and authentication flow are very similar to the one described in Figure 3.4. The only difference is the measurement of Δf, where it is the difference of the average of reference and stressed RO frequencies (see Equation 4.1).

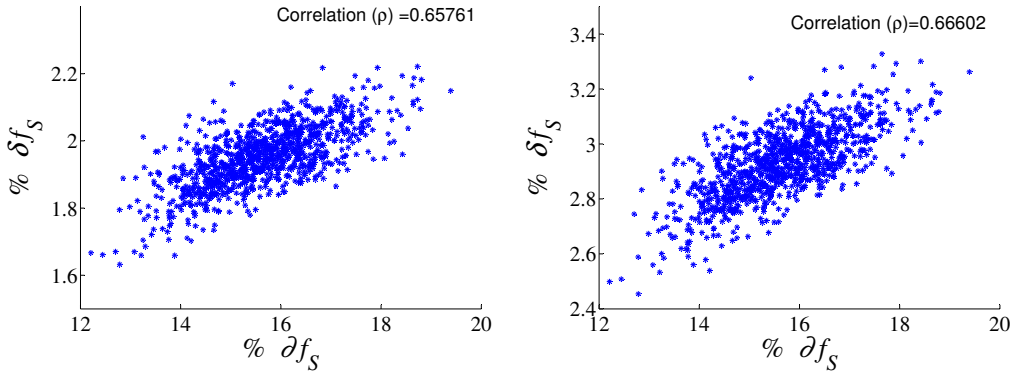
4.3 CDIR Sensor with multiple RO-pairs and Selection Approach (SN-CDIR)

SN-CDIR is based on the selection of the best RO pair that minimizes misprediction. Increasing the difference between the mean of two distributions (δf) for time t and 0 and reducing their spread (σ_0 and σ_t) are the key parameters for improving the detection of the recycled ICs. The selection of the best RO pair is the primary objective for minimizing the level of misprediction. As the reference RO remains quiet during normal operations, our objective is to find a stressed RO that degrades the most among all of the available stressed ROs. At the same time, we need to find a reference RO which is slower than the stressed RO at time 0. In the following, we will present a novel RO selection flow to select the best RO-pair for minimizing the misprediction. Let us start with a mathematical proof to find a maximum δf, the aging degradation.

4.3.1 Correlation between aging degradation (δf_S) and normalized frequency differences (∂f_S)

To find the maximum aging degradation (δf) of a CDIR, we have conducted an experiment to observe how δf varies with the percentage frequency differences (∂f_S) at different supply voltages. As the reference ROs remain quiet during normal operations, we have selected stressed ROs for this experiment. We have performed a Monte Carlo simulation with 1000 stressed RO samples implemented in PTM 90*nm* technology node [96] with two different supply voltages (1.2V and 1.4V). Figure 4.3 shows the scatter plot of δf_S versus ∂f_S at time t, where $\partial f_S = \frac{f_{0S,1.4V} - f_{0S,1.2V}}{f_{0S,1.2V}}$. Here, $f_{0S,1.4V}$ and $f_{0S,1.2V}$ are the frequencies of the stressed RO at 1.4V and 1.2V supply voltage. We have observed a positive correlation (ρ) for aging degradation

and normalized frequency differences (see Figure 4.3(a) and 4.3(b)). A theoretical proof will be presented below in Section 4.3.2.



(a) ICs aged for 3 days.

(b) ICs aged for 15 days.

Fig. 4.3: Scatter plot of percentage degradation ($\%\delta f_S$) versus percentage frequency differences ($\%\partial f_S$) of stressed ROs.

4.3.2 Proof of positive correlation between δf_S and ∂f_S

The amount of threshold voltage degradation (ΔV_{th}) due to voltage profiles experienced by the PMOS transistor can be represented by [97]

$$\Delta V_{th} \propto exp\Big(\frac{|V_{gs}| - |V_{tp}|}{E_0 t_{ox}}\Big)$$

where t_{ox} is the gate oxide thickness, V_{gs} is the gate-source voltage and $E_0 = 2.0$ MV/cm. Now, differentiating ΔV_{th} with respect to $|V_{tp}|$ results in

$$\frac{d\Delta V_{th}}{d|V_{tp}|} \propto \frac{-1}{E_0 t_{ox}} exp\Big(\frac{|V_{gs}| - |V_{tp}|}{E_0 t_{ox}}\Big) < 0$$

Thus, ΔV_{th} is a monotonic decreasing function with $|V_{th}|$ which results in higher degradation in at low V_{th} corner. This will result in a higher δf_S for low V_{th} corner. We can prove a positive correlation between δf_S and ∂f_S if we prove higher ∂f_S leads to the selection of a low $|V_{th}|$ PMOS transistor.

For simplicity, let us consider a simple ring oscillator (see Figure 3.2(a)) with n inverters consisting of one pMOS and one NMOS transistors. The frequency of that RO is $f = \frac{1}{2*n*t_d}$, where t_d is the delay of an inverter. Clearly, the frequency of an RO is inversely proportional to the delay of an inverter while assuming all the inverters are identical.

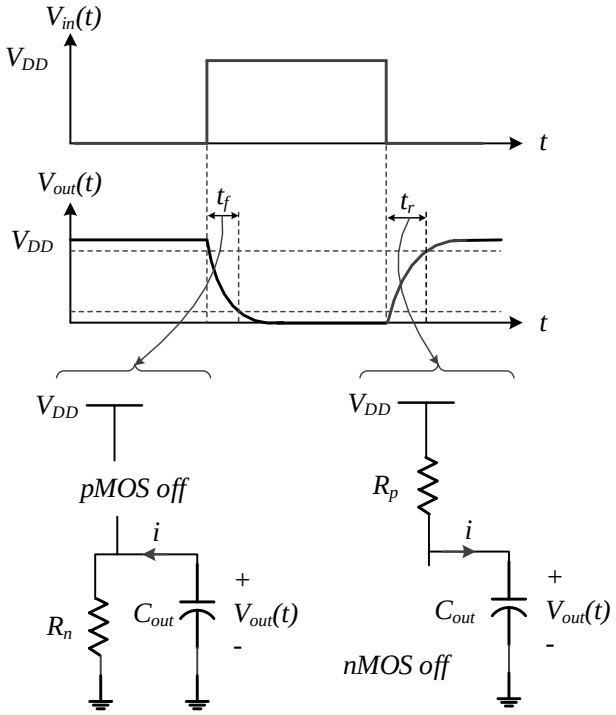


Fig. 4.4: Transient response of an CMOS inverter.

Figure 4.4 shows the transient response of an inverter and the charging and discharging circuits during its switching. The rise time t_r depends on the charging of the output capacitor C_{out} through R_p while the fall time t_f depends on the discharging of C_{out} through R_n. The *RC* time constants during charging and discharging are $\tau_p = R_p C_{out}$ and $\tau_n = R_n C_{out}$. The rise time and fall time are proportional to τ_p and τ_n respectively. Now, the propagation delay of an inverter

$$
\begin{aligned}
t_d &\propto (t_r + t_f) \propto (\tau_p + \tau_n) = (R_p + R_n) C_{out} \\
&\text{where } R_p = \frac{1}{\beta_p (V_{DD} + V_{tp})} \text{ and} \\
&R_n = \frac{1}{\beta_n (V_{DD} - V_{tn})} \\
\text{Thus, } f &\propto \frac{1}{R_p + R_n} = \frac{k}{R_p + R_n} \text{ where } k \text{ is a constant.}
\end{aligned}
$$

Now the percentage frequency differences,

$$
\begin{aligned}
\partial f_S &= \frac{f_{0S,V_{DD1}} - f_{0S,V_{DD2}}}{f_{0S,V_{DD2}}} = \frac{\frac{1}{R_{p1}+R_{n1}} - \frac{1}{R_{p2}+R_{n2}}}{\frac{1}{R_{p2}+R_{n2}}} \\
&= \left(\frac{R_{p2} + R_{n2}}{R_{p1} + R_{n1}} - 1 \right) \qquad (4.8)
\end{aligned}
$$

Differentiating ∂f_S with respect to $|V_{tp}|$,

$$
\begin{aligned}
&\frac{d}{dV_{tp}} (\partial f_S) = \frac{d}{dV_{tp}} \left(\frac{R_{p2} + R_{n2}}{R_{p1} + R_{n1}} - 1 \right) \\
= \; &\frac{(R_{p1} + R_{n1}) \frac{d}{dV_{tp}} (R_{p2}) - (R_{p2} + R_{n2}) \frac{d}{dV_{tp}} (R_{p1})}{(R_{p1} + R_{n1})^2} \\
&\text{as } R_{n1} \text{ and } R_{n2} \text{ are constants.}
\end{aligned}
$$

$$
\begin{aligned}
\text{Now,} \quad &\frac{(R_{p1} + R_{n1}) \frac{d}{dV_{tp}} (R_{p2})}{(R_{p2} + R_{n2}) \frac{d}{dV_{tp}} (R_{p1})} \\
= \; &\frac{(\frac{1}{V_{DD1}+V_{tp}} + \frac{1}{V_{DD1}-V_{tn}}) \times \left[- \frac{1}{(V_{DD2}+V_{tp})^2} \right]}{(\frac{1}{V_{DD2}+V_{tp}} + \frac{1}{V_{DD2}-V_{tn}}) \times \left[- \frac{1}{(V_{DD1}+V_{tp})^2} \right]} \\
= \; &\frac{V_{DD1} - V_t}{V_{DD2} - V_t}, \text{assuming } V_{tn} = -V_{tp} = V_t; \beta_n = \beta_p \\
> \; &1 \text{ as } V_{DD1} > V_{DD2}
\end{aligned}
$$

Thus, $\frac{d}{dV_{tp}} (\partial f_S) > 0$ signifies that ∂f_S is a monotonic increasing function with V_{th}. We can infer that the selection of a higher ∂f_S leads to the selection of a lower $|V_{th}|$ (higher V_{th} as it is negative) PMOS transistor. This results in a positive correlation (> 0) between δf_S and ∂f_S.

4.3.3 δf Versus ∂f_S

Let us assume that a stressed RO is used for about time t in the field. Due to aging, it slows down and the frequency f_{tS} at time t becomes lower than the frequency f_{0S} at time 0. Thus its aging degradation, δf_s, becomes

$$\delta f_S = f_{0S} - f_{tS}$$

The RO is operated at two different supply voltages at time 0 to calculate the percentage frequency differences

$$\partial f_S = \frac{f_{0S,V_{DD1}} - f_{0S,V_{DD2}}}{f_{0S,V_{DD2}}}$$

where, $V_{DD1} > V_{DD2}$.

There exists a positive correlation ρ (see Figure 4.3(a)) between δf_S and ∂f_S. Now our objective is to select a RO-pair that will maximize the aging degradation (δf) of the SN-CDIR based on the stressed RO percentage frequency differences (∂f_S) at different supply voltages.

$$\delta f \xleftarrow{\rho} \partial f_S$$

Here, the aging degradation for a CDIR is expressed as:

$$\delta f = \Delta f_t - \Delta f_0 \text{ where, } \Delta f_i = f_{iR} - f_{iS}$$

Note that δf and ∂f_S are random variables due to process variations.

Since we have shown above that a positive correlation exists between δf_S and ∂f_S, it is possible to find an optimal estimate $\hat{\delta f}$ for δf. Specifically, the minimum mean square error (MMSE) estimator [95] for the stressed RO degradation can be expressed as:

$$\hat{\delta f_S} = \rho \frac{\sigma_{\delta f_S}}{\sigma_{\partial f_S}} \left(\partial f_S - \mu_{\partial f_S} \right) + \mu_{\delta f_S}$$

where ρ represents the correlation between the δf_S and ∂f_S; $\sigma_{\delta f_S}$ and $\sigma_{\partial f_S}$ represent the standard deviations for δf_S and ∂f_S respectively; and $\mu_{\delta f_S}$ and $\mu_{\partial f_S}$ denote means for δf_S and ∂f_S respectively.

The MMSE estimator for the CDIR degradation (δf) can be written in terms of ∂f_S as follows:

$$\begin{aligned}
\hat{\delta f} &= \Delta\hat{f}_t - \Delta\hat{f}_0 = (\hat{f}_{tR} - \hat{f}_{tS}) - (\hat{f}_{0R} - \hat{f}_{0S}) \\
&= -(\hat{f}_{0R} - \hat{f}_{tR}) + (\hat{f}_{0S} - \hat{f}_{tS}) \\
&= \hat{f}_{0S} - \hat{f}_{tS}, \text{ Assuming } \hat{f}_{0R}=\hat{f}_{tR} \text{ as reference RO} \\
&\qquad\qquad \text{ages very little.} \\
&= \hat{\delta f_s} = \rho \frac{\sigma_{\delta f_S}}{\sigma_{\partial f_S}} \left(\partial f_S - \mu_{\partial f_S} \right) + \mu_{\delta f_S}
\end{aligned}$$

As ρ is positive, maximizing ∂f_S will maximize δf, which is the separation between the two distributions at $t = 0$ and $t = t$. *This implies that in SN-CDIR, where we have several RO-pairs to choose from, it is optimal to choose the one with the largest ∂f_S at $t = 0$. This will maximize the distance between the two distributions of frequency difference as depicted in Figure 4.1(a), resulting in lower probability of misprediction than single N-CDIR.*

4.3.4 Proposed Registration and Authentication Flow

Figure 4.5 shows the proposed registration flow of the SN-CDIR. The registration flow consists of the selection of the best reference and stressed RO-pair. During registration, a large number of new ICs are used to generate the distributions to determine the threshold after the manufacturing test process at the foundry. It is better to select the samples from different wafers and lots to capture the actual process variations. The larger this sample space is, the more accurate the process variations will be. In our simulation, we selected PV2 (mentioned in Table 3.3) as the extreme case, and we believe that any process variations will be well below PV2. The environmental conditions during measurement should be as uniform as possible to reduce measurement errors. However, we believe that the environmental variations should not impact the measurement as the reference and stressed ROs are placed close to each other so that environmental conditions will impact all of the ROs uniformly.

The objective of the registration phase is to find the best reference and stressed RO-pair. During this phase all the ROs in the CDIR are selected and their frequencies are captured. Let us assume that there are n reference and n stressed ROs in a CDIR. Two vectors

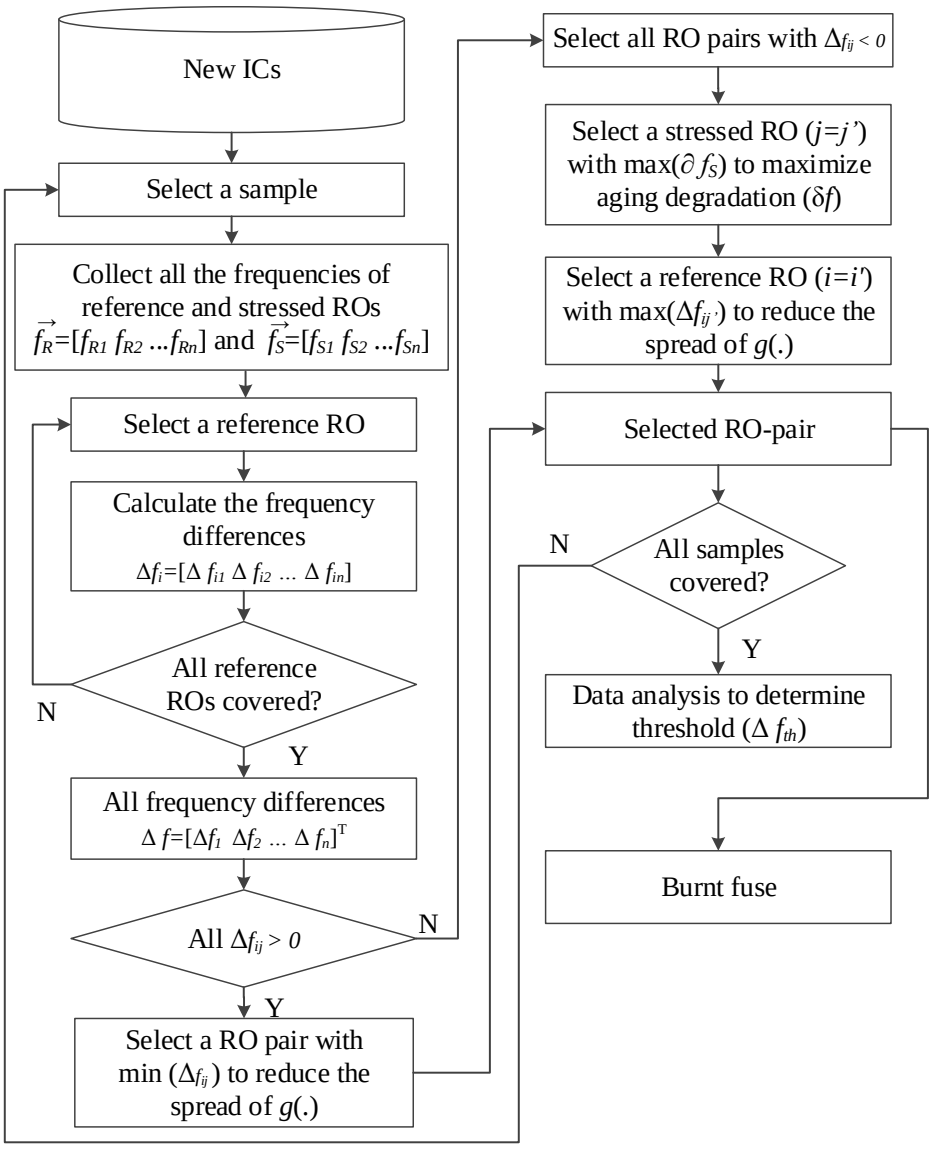


Fig. 4.5: Proposed registration flow for SN-CDIR.

$\vec{f}_R = [f_{R1}\ f_{R2} \ldots f_{Rn}]$ and $\vec{f}_S = [f_{S1}\ f_{S2} \ldots f_{Sn}]$ are formed to store all the reference and stressed RO frequencies. Now all the frequency differences are stored in a matrix $\Delta f = [\Delta f_{ij}]_{n \times n}$, where $\Delta f_{ij} = \vec{f}_R(i) - \vec{f}_S(j),\ \forall (i,j)$. If all the Δf_{ij} are positive, a reference-stressed RO pair is selected with $min(\Delta f_{ij})$, otherwise Δf is updated with only negative Δf_{ij} values. The resultant distribution $g_0^{'}(.)$ by applying these treatments can be visualized in Figure 4.1(c) where the spread has been reduced significantly.

It is now necessary to shift the distribution at time t, $g_t(.)$ to the right in order to increase δf even further (see Figure 4.1(a)). A stressed RO is selected which has maximum $\vec{\partial} f_S(j) = \frac{\vec{f}_{OS,V_{DD1}}(j) - \vec{f}_{OS,V_{DD2}}(j)}{\vec{f}_{OS,V_{DD1}}(j)}$. The corresponding reference RO is selected with $max(\Delta f_{ij})$ to reduce the spread of both $g_0(.)$ and $g_t(.)$ distributions. Once the best RO pair is selected, the frequency difference (Δf_{ij}) is stored to form the distribution ($g_0(.)$). A fuse block is used to select these two ROs permanently. All the ICs go through a similar treatment to find out the best RO pair. Finally the threshold is calculated, which will be used later for the detection of recycled ICs.

Note that in the worst case, there may not be a correlation present between δf_S and ∂f_S, as some researchers reported a very weak or zero correlation between the aging degradation with threshold voltage [98]. However, this will not impact the result significantly as the removal of right hand side and the reduction of the spread of the distribution at t=0 ($g_0(.)$).

The authentication flow is exactly same to the one described in Figure 3.4. The frequency differences (Δf) of the reference and stressed ROs of an IC under authentication are measured and compared with the threshold (Δf_{th}) to determine whether the IC is recycled or not. The CDIR will experience more degradation once it has been used longer in the field, and Δf will be much larger than Δf_{th} making it easier to identify.

4.3.5 Proposed Architecture of SN-CDIR

Figure 4.6 shows our proposed architecture for the SN-CDIR. It also consists of reference RO and stressed RO blocks like AN-CDIR. Each block consists of an equal number of NBTI-aware ROs. The number of ROs in each block depends on the recycled IC detection resolution. For example, it is required to place four ROs in each block when we want to detect recycled ICs aged for only 12 hours with a 100% workload (see Table 4.3). Larger numbers of ROs are necessary to achieve a superfine recycled IC detection resolution. Like AN-CDIR, the required number of ROs can be determined after observing the available area on a chip.

All the ROs in the reference and stressed blocks are fed to two different multiplexers (MUX_R and MUX_S) respectively. The selection input of MUX_R and MUX_S are provided by the LSBs and MSBs of a shift register. If there are n ROs in each block, the selection input of each multiplexer will be $log_2(n)$. Thus the size of the shift register will be $2log_2(n)$. This shift register can accept data from the MUX_SEL pin or a $2log_2(n)$ bit fuse/antifuse block. During reg-

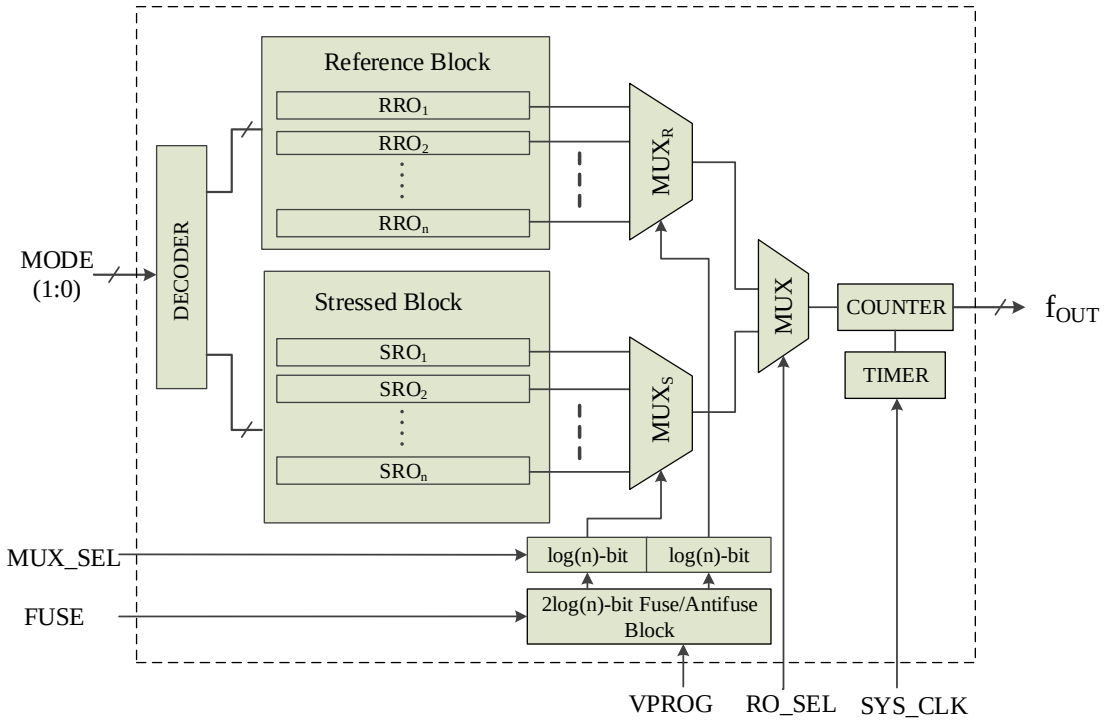


Fig. 4.6: The architecture of our proposed AN-CDIR.

istration phase, all the ROs are selected to measure their frequencies. In this phase, MUX_SEL selects each RO. At the end of the registration phase, the best RO-pair is determined. The selection bits corresponding to these ROs are programmed in a $2log_2(n)$ fuse/antifuse block. $VPROG$ provides the programming voltage to the fuse/antifuse block.

The rest of the design is similar to the N-CDIR described in Figure 3.3. The $DECODER$ generates all the internal signals (see Table 3.1) for the reference and stressed RO blocks. It is not necessary to generate the control signals for each RO in the reference and stressed RO blocks. All the ROs in each block utilize the same internal signals generated by the $DECODER$. The MUX, $COUNTER$, and $TIMER$ operate as described before.

4.4 Simulation Results and Analysis

In this section, we will present the simulation results for the AN-CDIR and SN-CDIR structures. We present the simulation results for the process variations PV2 (see Table 3.3) to evaluate the

performance of the CDIRs in the most extreme cases. We believe that these CDIRs will perform better than standard manufacturing processes, as these ROs are placed very close to each other. As in the case of the N-CDIR, we have implemented and simulated these CDIRs using the 90nm technology node [96]. HSPICE MOSRA from Synopsys is used to simulate and measure the impact of aging on this CDIR sensor. The nominal supply voltage is 1.2V. In this simulation, we selected 51-stage ROs as they outperform the 21-stage RO (see Table 3.4). To model the variation, the Monte Carlo (MC) simulation was performed with 1000 samples of the CDIRs. First we will present the results when the CDIRs are aged for only 3 days with a 100% workload. Larger usage time than 3 days would be easily detected using these sensors.

Figure 4.7 shows the histogram plot of the average frequency difference between the reference and stressed RO-pairs for different number of RO-pairs in an AN-CDIR. We have observed that the spread of the distributions at time 0 and 3 days reduced significantly while increasing the RO-pairs in the CDIR. However, the separation between the two distributions remains the same. The threshold (Δf_{th}) for determining whether the ICs under tests are new or recycled is the same for all the different RO-pairs in a CDIR and is $\frac{\mu_{t=0}+\mu_{t=3D}}{2} = 2.77MHz$. Figure 4.8 also reveals the same fact that to detect recycled ICs aged fewer than 3 days requires more than 2 RO pairs in a CDIR. The higher the number of RO pairs, the better the likelihood that we will be able to detect recycled ICs that have been used less in the field.

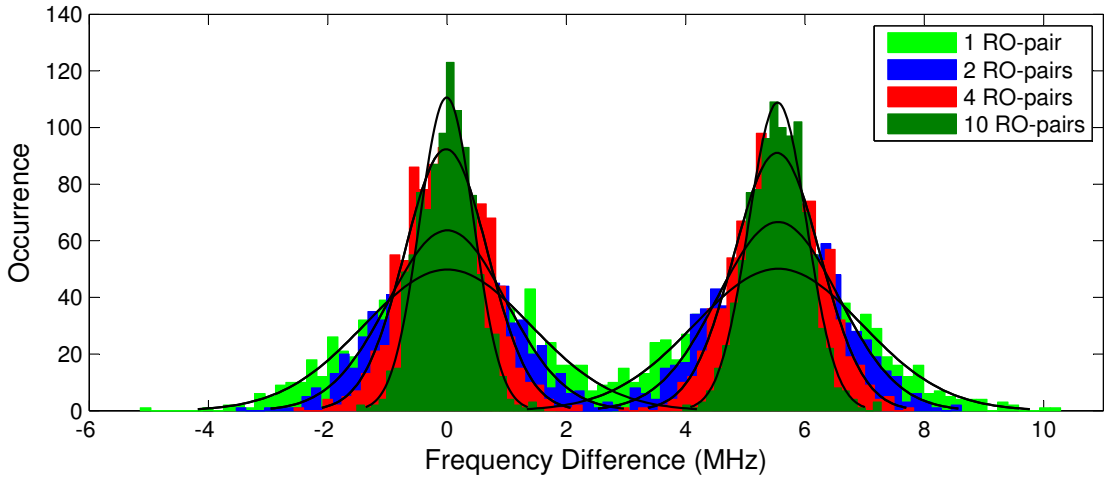


Fig. 4.7: The frequency difference distribution at PV2 of AN-CDIR with different number of RO-pairs.

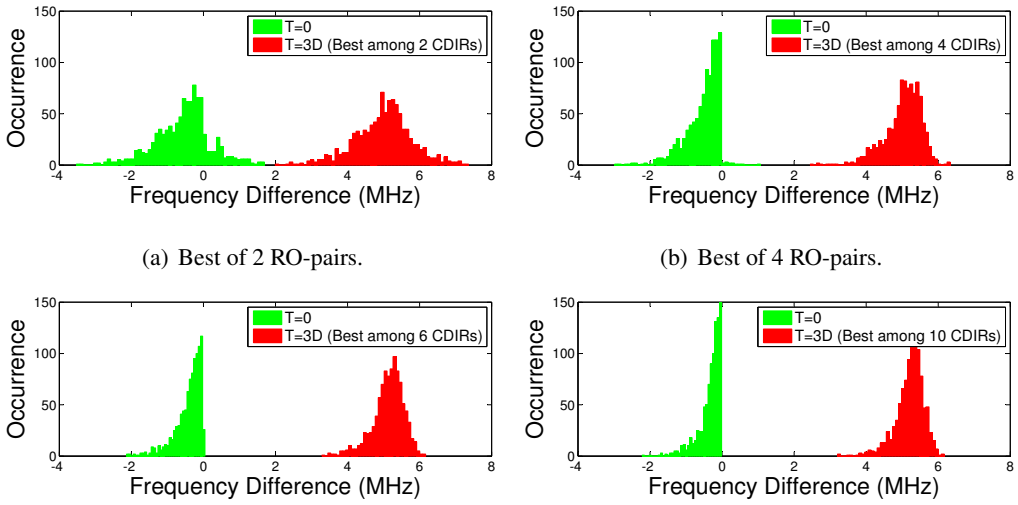
It is relevant to analyze how the mean (μ) and variance (σ^2) of the frequency difference distribution of the AN-CDIR changes with an increased number of RO pairs to estimate the misprediction accuracy and the number of RO-pairs required to achieve nearly a zero misprediction rate. By using the *normfit* MATLAB function [99], we measure the actual mean and variance (denoted as *Act*. in Columns 3, 5, 7, and 9 of Table 4.2) of different distributions with different numbers of RO pairs in a CDIR, and compare them (denoted as *Exp*. in Columns 2, 4, 6, and 8 of Table 4.2) with Equations 4.6 and 4.7 to measure the accuracy of our averaging algorithm. Table 4.2 shows the values for μ and σ. We have observed an error in the expected value (*Exp*.) compared to the actual value (*Act*.), which is less than 0.5% for μ and less than 9% for σ.

Table 4.2: Mean and variance f distribution of AN-CDIR.

# RO Pairs	$g_0(.)$				$g_{t=3D}(.)$			
	μ		σ		μ		σ	
	Exp.	Act.	Exp.	Act.	Exp.	Act.	Exp.	Act.
2	0.000	0.004	0.986	0.987	5.553	5.551	0.994	1.007
4	0.000	-0.016	0.697	0.695	5.553	5.533	0.703	0.722
6	0.000	0.003	0.569	0.580	5.553	5.551	0.574	0.608
10	0.000	-0.005	0.441	0.452	5.553	5.542	0.444	0.488

Now we will analyze the performance of the SN-CDIR. Figure 4.8 shows the histogram plot of the frequency difference between the selected best reference and stressed RO pair. We have observed that there is no overlap between the two distributions at time 0 and 3 days for all the figures, Figure 4.8(a) - 4.8(d). However, the separation between the two distributions increases as the number of RO-pairs increases. The threshold (Δf_{th}) (see Figure 4.5) for determining whether the ICs under tests are new or recycled is $2MHz$ for all the CDIRs with different RO pairs. Figure 4.8 reveals that, to detect a recycled IC aged for only 3 days with zero-misprection, does not require more than 2 RO pairs in a CDIR. However, there will be an inevitable overlap between the two distributions when the ICs are aged for fewer than 3 days.

In that case, a higher number of RO-pairs in a CDIR would be required.



(a) Best of 2 RO-pairs.

(b) Best of 4 RO-pairs.

(c) Best of 6 RO-pairs.

(d) Best of 10 RO-pairs.

Fig. 4.8: The frequency difference distribution at PV2 of SN-CDIR with different number of RO-pairs.

Table 4.3 shows the prediction accuracy of the AN-CDIR and SN-CDIR. The rate of misprediction (i.e. recycled ICs identified as new (θ_1) and new ICs identified as recycled (θ_2)) is also estimated by using Equation 3.2 for the AN-CDIR. We cannot apply this equation to estimate the rate of misprediction for the SN-CDIR, as the distributions are no longer Gaussian. These mispredictions are calculated as:

$$\theta_1 = \frac{\#samples\ with\ \Delta f < \Delta f_{th}}{Total\ samples} \times 100\%$$

$$\theta_2 = \frac{\#samples\ with\ \Delta f > \Delta f_{th}}{Total\ samples} \times 100\%$$

Column 1 represents the duration for the two CDIRs that are aged. Column 2 represents the number of RO-pairs in a CDIR. In this simulation, we have implemented a maximum of 10 RO-pairs in a CDIR. We observe from this table that a larger number of RO-pairs are required to detect ICs that have been aged less amount of time. We can identify all ICs (recycled or new) without any error for both CDIRs with 2 RO-pairs when the aging duration is more than 15 days. All ICs with 3 days of prior usage can be detected by using SN-CDIR with 2 RO-pairs without any error whereas the prediction errors for AN-CDIR with 2 RO-pairs are 0.2938%

and 0.2511%. We require 4 RO-pairs for both CDIRs to identify ICs with only 1 day of aging. However, SN-CDIR provides better prediction accuracy. We require 6 and 4 RO-pairs for SN-CDIR and AN-CDIR to identify ICs with only 12 hours of aging. For SN-CDIR, we need to set the threshold (Δf_{th}) carefully such that θ_1 and θ_2 are of similar value.

If there is no overlap between the two distributions ($g_0(.)$ and $g_t(.)$), then one can select a threshold (Δf_{th}) greater than 0. For example, one can select $\Delta f_{th} = 2MHz$ for Figures 4.8(b) - 4.8(d). However, in this table, we mention $f_{th} = 0$ even though there is no overlap. When the ICs are aged with less amount of time the distribution ($g_t(.)$) shifts very little to the right and there might be a possible overlap by the tail of $g_t(.)$ with $g_0(.)$. Thus it is wise to select a threshold (Δf_{th}) near 0.

When the ICs are used for a very small amount of time, the performance of SN-CDIR outperforms AN-CDIR. For example, θ_1 and θ_2 are 13.07% and 12.85% for a AN-CDIR with 2 RO-pairs whereas they are 7.3% and 7.5% for a SN-CDIR. We can detect recycled ICs with 2 hours of aging by using a SN-CDIR with 6 RO-pairs whereas it requires a AN-CDIR with 10 RO-pairs. In addition, the misprediction rate is also higher for the AN-CDIR.

4.4.1 Area Overhead Analysis

Table 4.4 shows the area overhead analysis of all the CDIRs. We simulated several IWLS 2005 benchmarks ranging from low to high sizes to compute the area overhead. The area overhead is defined as the ratio of the size/area of the CDIR with the size/area of the benchmark. We have not considered the size of the timer and counter while calculating the area for the CDIRs as we assume the frequency measurement can be performed off-chip. We also assume that the area for the AN-CDIR and SN-CDIR with the same number of RO-pairs is almost same. The additional area for the SN-CDIR comes from the fuse/antifuse block, and we can neglect this for simplicity's sake.

As seen above, the overhead is more than 1% for small benchmarks (*i2c, spi, and b14*) for the 51-stage NBTI-Aware RO-CDIR (N-CDIR) that could make it challenging to use them in small designs. The area overhead for the 51-stage O-CDIR [1] is less than our 51-stage

Table 4.3: Misprediction Analysis of AN-CDIR and SN-CDIR.

Aging duration	**RO-pairs (n)**	**AN-CDIR**		**SN-CDIR**		
		$\theta_1(\%)$	$\theta_2(\%)$	Δf_{th}	$\theta_1(\%)$	$\theta_2(\%)$
2 Hrs	2	13.07	12.85	0.4	7.3	7.5
	4	5.46	5.51	0.1	0.8	1.0
	6	2.73	2.77	0	0.0	0.1
	10	0.62	0.81	0	0	0
4 Hrs	2	9.0	8.9	0.7	4.5	4.8
	4	2.8	2.79	0.2	0.5	0.7
	6	1.1	1.15	0	0	0
	10	0.13	0.2	0	0	0
8 Hrs	2	5.58	5.38	1.0	3.1	2.9
	4	1.24	1.12	0.5	0.2	0.2
	6	0.32	0.34	0	0	0
	10	0	0	0	0	0
12 Hrs	2	3.91	3.75	1.0	1.7	2.9
	4	0.64	0.58	0.5	0.1	0.2
	6	0.12	0.14	0	0	0
	≥ 8	0	0	0	0	0
1 Day	2	1.82	1.65	1.4	0.7	0.7
	4	0.14	0.14	1.0	0.0	0.1
	≥ 6	0	0	0	0	0
3 days	2	0.29	0.25	0	0	0
	≥ 4	0	0	0	0	0
15 days	≥ 2	0	0	0	0	0

N-CDIR. However, for medium and large designs, the area of the O-CDIR or N-CDIR would hardly impact the overall area of the design.

Table 4.4: Area overhead analysis.

Benchmark	Size (# Gates)	Area Overhead (%)					
				AN-CDIR/SN-CDIR			
		O-CDIR [1]	N-CDIR	n=2	n=4	n=6	n=10
i2c	1142	5.52	9.98	19.44	38.35	57.3	95.1
spi	3277	1.92	3.48	6.774	13.37	20	33.1
b14	8679	0.73	1.31	2.558	5.047	7.54	12.5
b15	12562	0.50	0.91	1.767	3.487	5.21	8.65
DMA	19118	0.33	0.60	1.161	2.291	3.42	5.68
DSP	32436	0.19	0.35	0.684	1.35	2.02	3.35
ethernet	46771	0.135	0.24	0.475	0.936	1.4	2.32
vga_lcd	124031	0.051	0.09	0.179	0.353	0.53	0.88
leon2	780456	0.008	0.01	0.028	0.056	0.08	0.14

The area overhead for the AN-CDIR and SN-CDIR is comparably high for higher numbers of RO-pairs. Both CDIRs with 2 RO-pairs can be implemented in designs larger than the benchmark *DMA* with minimum area overhead. On the other hand, these CDIRs with 10 RO-pairs can only be implemented in large designs. As the size of most current system-on-chips (e.g., microprocessors, digital signal processors, microcontrollers, etc.) are comparable or larger than *vga_lcd* benchmark, we can successfully implement these CDIRs without affecting the area overhead. In summary, the designers can select a CDIR depending on the area budget that can satisfy the requirements on minimum usage time for detection.

4.4.2 Attack Analysis

Due to the evolving nature of IC recycling activities, it is of utmost importance to analyze all of the possible attacks on these CDIRs and their vulnerabilities in order to examine their robustness. Recyclers are always in the process of improving their old technologies through experience and adopting new methodologies. In this section we will analyze all the possible

attack scenarios and their impact on our CDIRs.

• *Removal or tampering of the CDIR*: The first attack on the CDIRs could be removal or tampering attacks. In this scenario, the attacker tries to replace the stressed RO with a new counterpart or tries to tamper with the connections inside the multiplexer. However, it is fairly impossible to replace the stressed RO with a new one. Currently, recyclers have the capability to tamper with the connections by using FIB circuit edit [100]. If we assume that the tampering is possible, then the counterfeiter must remove the old package and again repackage and remark it according to its original specifications. This removal and repackaging may not be cost effective to the counterfeiters. Hence, it is unlikely to be used in practice.

• *Age Reference RO*: The attacker will try to intentionally age the Reference RO to mask the frequency differences between the reference and stressed ROs. In this scenario, the attacker forces the CDIR to work in authentication mode (MODE 10, in Table 3.1) under accelerated stress conditions. However, in this mode, the stressed RO will also be in oscillation resulting similar amount of aging. To mask the initial aging difference, the recycler must age the chip for a long period of time. Burn-in is very expensive as there are hundreds of different IC types, and the recycler must have an expensive setup for all different ICs. The primary incentive for counterfeiting is cheap recycling, not adding extra cost to the components. There might not be any motivation left for the counterfeiters when they are forced to add burn-in to their recycling process.

4.5 Summary

In this chapter, we have presented three different structures based on NBTI-aware ring oscillators to detect recycled ICs used only for very short period of time. The reference ROs in these CDIRs remain quiet during the normal operation of the IC while the stressed RO gets aged at an accelerated pace utilizing NBTI of PMOS transistors. This helps to get a reasonable frequency difference between the reference and stressed ROs even though an IC is used only a very short duration. We proposed two different versions of CDIRs with multiple RO-pairs where the designer can select the number of RO-pairs depending on their area budget. These CDIRs provide better prediction accuracy compared to N-CDIR. AN-CDIR with 10 RO-pair

can detect recycled ICs aged only for 2 hours with a very little misprediction rate. SN-CDIR provides even better accuracy than AN-CDIR. We can detect recycled ICs with certainty even though they have been used only for 2 hours in the field.

Chapter 5

Establishment of Forward Trust for Protecting IPs and ICs

In Chapters 3 and 4, we have discussed the detection and avoidance of recycled ICs. However, a large number of counterfeit ICs belongs to the overproduced, out-of-spec/defective and cloned types. In this chapter, we focus on developing design-for-anti-counterfeit (DfAC) measures to prevent IP piracy, IC cloning, IC overproduction, and sourcing of out-of-spec/defective ICs into the supply chain. We begin by discussing the vulnerabilities associated the first three phases (e.g., design, fabrication, and assembly) of the supply chain (see Figure 1.3), where these counterfeit types are originated.

The complexity of IC design has grown exponentially due to the persistent trend of device scaling, which enabled designers to fit more and more functionality on a system-on-chip (SoC) to reduce overall area and cost of a system. It is fairly impossible to design a complete system by an SoC designer alone. Therefore, the semiconductor industry has shifted gears to the concept of design reuse rather than designing the whole SoC from scratch. Nowadays, the SoC designers obtain licenses for various functional blocks (known as intellectual properties or IPs) for their SoCs to optimize the design process and decrease time-to-market.

In parallel, the increased complexity of the fabrication process has resulted in a majority of SoC designers no longer maintaining a fabrication unit (foundry or fab) of their own. Building and maintaining such foundries for modern SoCs are reported to cost more than several billions of dollars and increasing as technology further scales [101]. Given the increasing cost, the semiconductor business has largely shifted to a contract foundry business model (horizontal business model) over the past two decades. In this business model, the SoC designers first get licenses for 3PIPs to be used in their SoC designs, design the SoCs by integrating the various 3PIPs and then outsource the SoC design to the foundries and assemblies for fabrication and

packaging to reduce time-to-market and manufacturing costs.

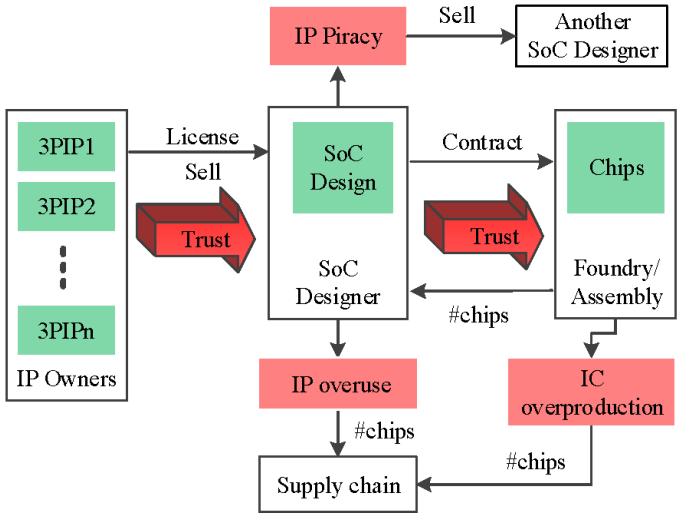


Fig. 5.1: Lack of trust between the IP owners, SoC designer, and foundries/assemblies in SoC design and fabrication process.

In the modern SoC design and fabrication flow, forward trust does not exist between the participating entities. The IP owners cannot have complete trust on the SoC designers, whereas the SoC designers may not trust the foundries or assemblies. The lack of transparency and the resulting lack of trust may lead to the following vulnerabilities, as shown in Figure 5.1.

- *IP overuse*: The SoC designer may produce more ICs and report a lesser amount to the IP owners to escape the licensing cost. At the same time, the SoC designer may illegally use an IP that was licensed to be used in a different design. In the current manufacturing practice, the IP owners have no or little means to verify how many chips have been manufactured with their IPs and where they have been used. When an untrusted party overuses the IPs and sells the extra chips in the open market, the IP owners lose any possible revenue that could have been gained from those chips.

- *IP piracy*: An SoC designer may legally purchase a third party IP core from an IP owner and then make clones, or illegitimate copies of the original IP [102] [103] [68] [104]. Similarly, untrusted foundries may sell illegal copies of the GDSII files that they receive from SoC designers for fabrication. Further, the integrity of the IP may be at risk as an

untrusted SoC designer can potentially add some extra features to those 3PIPs to make them look like a different one and then sell them to another SoC designer. An SoC designer may also modify a 3PIP in order to introduce a backdoor or hardware Trojan into the chip to leak secret information to the attacker or disable a system at some critical point in time.

- *IC overproduction*: Untrusted foundries and assemblies may produce more than the number of chips they are contracted to manufacture [24] [65] [104]. As no R&D cost are incurred for these chips and the masks are developed with SoC designer's expenses, they can receive illegitimately larger profits by selling these chips with the name of SoC designer. In addition, they can also overbuild chips practically at zero cost by reporting a lower yield (i.e., percentage of defect-free chips to the total number of chips) to the SoC designer [25] [69]. Along with the financial loss of the SoC designer, an even bigger concern with these ICs is that of reliability. Overproduced ICs may simply end up in the market with minimal or no testing for reliability and functionality. These ICs may also find their way into the supply chain for many critical applications, which raises concerns for safety and reliability. Since these ICs have the same name of the SoC designers, their failure would tarnish company reputation.

5.1 Prior Work

The existing work on preventing IP piracy and IC overproduction can be classified into three major categories.

5.1.1 Logic obfuscation

This is a technique where a design is transformed to a different one to obfuscate the inner details of the original design, thus preserving the original functionality [105]. In [68], the authors proposed a methodology which can be integrated into the SoC design and manufacturing flow to simultaneously obfuscate and authenticate a design. In this approach, the circuit operates in a normal mode when it receives a predefined sequence of patterns, known as a key, at its input. However, it is not clear how this key will be hidden from the foundries or assemblies as it is

necessary to prevent overproduction. In addition, this technique does not address IP overuse.

The authors in [65] first proposed to encrypt a netlist by using a lock (a set of XOR/XNOR gates) and it can only be unlocked by using a chip unlock key (CUK). The design is not resistant to reverse engineering as key gates are directly related to the key bits (XOR and XNOR gates indicate 0 and 1 at CUK location, respectively) and vulnerable to key sensitization attacks [106]. The authors in [106] addressed those problems by proposing different logic encryption techniques. The authors in [107] has shown that any logic encrypted circuit can be broken. However, they assume that *an attacker can use scan-chain to read/write the values of all flip-flops in the design.* This assumption does not conform to today's designs. Every design now use test compression architecture to significantly reduce the test cost by reducing test time and test data volume [108–110]. Test responses are compacted many folds before it becomes available for off-chip access. As the modern EDA tools provides diagnostic support (high defect coverage and accurate fault diagnostics) with compression in place [108–110], it is impractical not to incorporate test compression in the design. It is now impossible to access the individual flip-flop values (the output of the combinational circuit, Y for a solution to a QBF) for chips where the design uses test compression. It is impossible to find a key using the approach suggested by the authors by looking at the compacted scan output values. Thus, it is still safe to use the scheme proposed in [106] to encrypt netlist.

Recently, Design Automation Standards Committee of the IEEE developed the standard P1735 [111] to provide the guidance for encryption and management of IPs, which has been adopted by most IP and EDA vendors. In the encryption approach, the IP is encrypted with a random symmetric session key. This session key is then encrypted with the public keys of different EDA vendors and attached to the IP such that these vendors can later reconstruct the original IP. Figure 5.2(a) shows a very simple IP which performs *AND* operation in every clock cycle. To protect from any unwanted modification, the IP is encrypted by using Synopsys *encryptP1735.pl* script [112]. In this encryption process, the code inside the *'pragma protect* block (encircled in red in Figure 5.2(a)) will be encrypted. The encrypted IP is shown in Figure 5.2(b), where the code inside the *'pragma protect* block (encircled in red) is not recognizable to anyone. During decryption, the session keys are decrypted by using the private key of the EDA vendor and then encrypted portions of the IP is decrypted by using this session key. One can

find this process in detail in [112] [113]. Unfortunately, this encryption approach cannot prevent placing additional features to an existing IP as it does not provide any integrity verification. Figure 5.2(c) shows this modified encrypted IP where the attacker adds an extra feature (*OR* operation) to the existing one (*AND* operation). We will provide a solution by adding an IP digest resulted from a cryptographic hash function [114] in the IP header (see Section 5.3.5) to prevent any unauthorized modifications. In addition, the encrypted IP does not provide any protection against copying of the whole IP to make an exact clone. As our solution uses an encrypted netlist, copying the entire IP will not help an attacker unless he possesses a valid *CUK* (see Section 5.3.2).

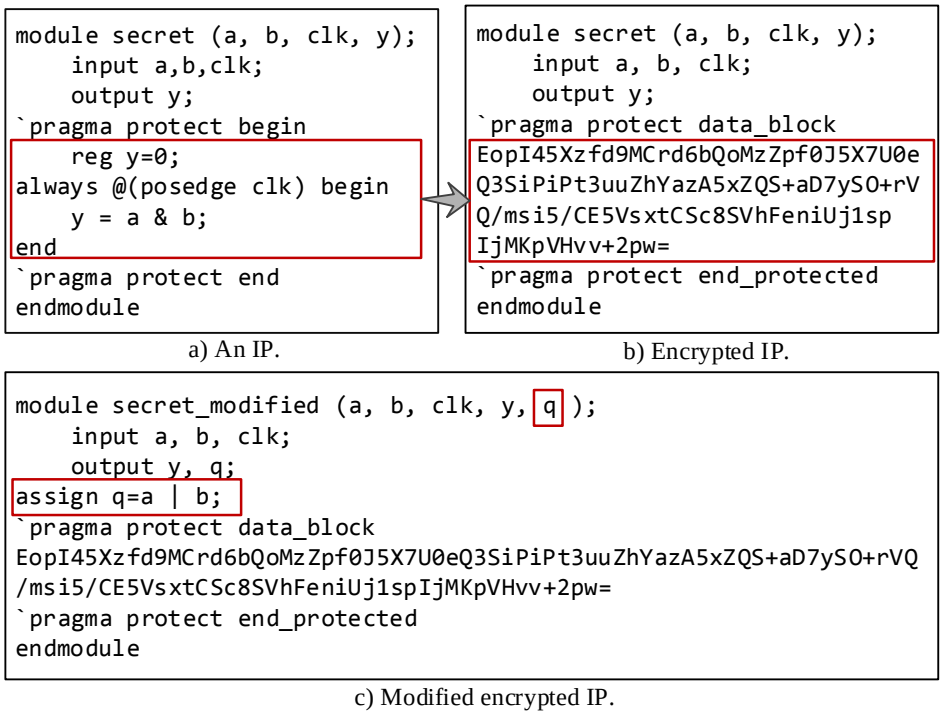


Fig. 5.2: Vulnerabilities associated with an encrypted IP.

5.1.2 Hardware watermarking

This approach has received much attention in the recent years for validating the authorship of an IP. Watermarking techniques uniquely identify an IP by creating a unique fingerprint in it [115] [116] [117] [102] [118] [119]. As the watermarking technique is passive, one cannot use

it to prevent IP overuse, IP piracy, and IC overproduction. Rather, it can only be used to verify proof of IP use.

5.1.3 IC metering

The existing metering approaches prevent IC overproduction by attempting to give an SoC designer control over the number of ICs manufactured. These approaches can be either passive or active. Passive approaches uniquely identify each IC and register the ICs using challenge-response pairs. Later, suspect ICs taken from the market are checked for proper registration [60] [61] [50] [62] [48] [59].

For passive metering techniques, one major limitation is that they cannot "actively prevent" overproduction. PUF-based detection techniques relies on the matching unclonable IDs/signatures generated by the PUF. The challenge-response pairs are stored in a secure database and verified later whether the responses are listed in the database or not. In the context of our problem, the SoC designers have to count on the foundries/assemblies to send them all defect free chips and trust them blindly on yield information. An untrusted foundry/assembly can hide actual yield information and practically build huge amount of defect free chips. They can literally send these unregistered chips to many different places (subsidiary companies, rogue system integrators, and many others who look for low cost parts). These untrusted entities might not care about the authenticity of ICs.

Active metering approaches lock each IC until it is unlocked by the SoC designer [65] [53] [63] [64] [66] [67]. The PUF-based active metering technique presented in [120] has an applicability limitation. In this proposed scheme, foundry needs to capture the initial power-up state through the scan-chains and send that state to the design house to receive the passkeys. The authors did not provide solutions with test compression architecture in place. Test compression is being adopted by the community to significantly reduce test data not out of preference, but of necessity. Every designs now use test compression architecture [2]. Test responses (the flip-flop values) are compacted many folds before it becomes available for off-chip access. It is impossible to access the flip-flop values unless there is a bypass of the compression module. This may create additional overhead. Similar analysis is also applicable to the scheme presented in [53,64].

In [25] [69] the authors proposed secure split-test (SST) to prevent overproduced, out-of-spec/defective, and cloned ICs. SST enables the design house to participate in the manufacturing test process by placing a set of security measures in the design and controlling the test flow. However, the major disadvantage is the back and forth communication between the SoC designer and the foundry/assembly, which increases delay in the test process.

In [65], the authors used an on-chip TRNG to generate a public-private key-pair for RSA encryption. This approach suffers from three major issues. First, there is large design overhead due to the on-chip RSA key generation. The keys (e.g., 1024 bit to achieve 80-bit security) are derived by a complex algorithm from two large prime numbers, p and q, which are generally 512 bits long each [121]. A prime checker is also required to verify these numbers are indeed prime. Second, the scheme assumes a secure transfer of public key from the chip to the SoC designer *which creates a vulnerability to man-in-the-middle attacks*. The foundry can always intercept the public key from the chip (more interestingly, foundry initiates the communication) and replace it with a new key, which nullifies the objective of creating on-chip key-pairs. Third, the scheme suffers from the key sensitization attacks [106].

5.2 Contributions

In this chapter, we will present ***FORTIS***, a comprehensive solution for establishing forward trust for protecting IPs and ICs against the attacks discussed above. We address each issue as follows:

5.2.1 IC overproduction

We develop a novel communication protocol for activating chips after fabrication. The protocol is similar to Pretty Good Privacy (PGP) by Phil Zimmermann, which is commonly used today in email delivery systems and has demonstrated excellent secrecy over the years [122]. In our approach, the design is locked by using a set of key gates and can only be unlocked upon receiving a CUK. To encrypt a design by using a CUK was first introduced in [65]. An improved version which is resistant to reverse engineering and various attacks, was presented in [106]. FORTIS uses one of this attack resistant encrypted netlist to prevent IC overproduction.

The major challenge here is to transfer this CUK to the chip from the SoC designer without being intercepted by any untrusted party (including untrusted foundry). Our proposed approach addresses this problem of key transfer from SoC designer to the foundry/assembly. Every chip has two static RSA keys (same for all chips) and a dynamic session key (different for everyone). Our approach does not require on-chip key generation which significantly reduces the area overhead compared to previous techniques.

As discussed above, prior approaches also have major limitations when testing is performed. Either the chip has to be unlocked [65] or test responses to be sent to the SoC designer [25] [69] [12] create additional vulnerabilities in the design flow. In our proposed approach, it is not required to provide CUK during test pattern generation (see Section 5.3.2). This helps us to perform manufacturing tests without unlocking the chips. Our proposed approach does not impact manufacturing tests and prohibit unwanted activation of ICs during test.

5.2.2 IP overuse

We address IP overuse by introducing a trusted authentication platform (TAP) in the SoC. This TAP is trusted by all parties involved in the SoC design, and can be imported as a trusted third party IP. In our proposed approach, each IP is locked with key gates. The synthesis and test pattern generation flow is very similar as before. To the best of our knowledge, our proposed IP metering approach addresses the third party IP (3PIP) metering problem for the first time in a forward trust manner.

5.2.3 IP piracy

We use IP encryption [111] in our design flow to encrypt the netlist. We propose IP integrity verification (see Figure 5.9) to make it resistant to modification, whereby the malicious SoC designer/foundry cannot modify a 3PIP by adding/disabling features. Along with this the netlist is locked by using a set of key gates to prevent the cloning of IPs. One question that could arise is that if a 3PIP is locked by using a secret CUK, then how an SoC designer will simulate an SoC which uses that 3PIP. We address this issue by attaching CUK to the IP header and then encrypt it by using EDA tool's public key such that the tool can retrieve the CUK during simulation.

Table 5.1: Comparison of different approaches to ensure forward trust.

Scheme	IP Overuse	IP Piracy		IC Overproduction		Resistant to Attacks
		Detection	Prevention	Detection	Prevention	
Logic Obfuscation	✗	✓	✗	✗	✗	Low
Hardware Watermarking	✗	✓	✗	✗	✗	Low
IC Metering	✗	✗	✗	✓	✓	Low
FORTIS	✓	✓	✓	✓	✓	High

Table 5.1 shows the summary of our contributions compared to existing research. IP piracy and IC overproduction are both categorized into detection and prevention categories. An IP/IC can only be detected as pirated/overproduced by the detection approaches, whereas prevention approaches prevent pirated IPs or overproduced ICs from entering into the supply chain. Our proposed approach, FORTIS, addresses the challenges for establishing forward trust, whereas the other approaches try to address the problem partially. The 3PIP owners protect their IPs from untrusted SoC designers and foundries when they use FORTIS in their design flow. Similarly, the SoC designers protect their SoCs from untrusted foundries. Further, FORTIS is the only approach that prevents modifications to any 3PIPs. Our proposed design flow inherently assures forward trust in the SoC design and fabrication process.

5.3 FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs

With increasing SoC design complexity, design reusability has become an integral part of the SoC design process. Unfortunately, this creates the risk of overuse of 3PIPs by untrusted SoC designers and foundries. In addition, the SoC designers lose profits once an untrusted foundry/assembly overproduces chips and sells them under their name. Thus, forward trust is extremely important to the entities involved in the SoC design. The IP owners need to trust the SoC designer, whereas the SoC designers must trust the foundries and assemblies. In this section, we present a comprehensive solution for establishing forward trust between the entities involved in SoC design and fabrication for protecting IPs and ICs. FORTIS automatically

ensures the forward trust among these entities.

5.3.1 Proposed Design Flow of FORTIS

Figure 5.3 shows our proposed design flow for establishing forward trust between various entities involved with the SoC design process. The design flow is very similar to the existing IC design flow except for the lock insertion and functional activation steps. Our design process starts with the insertion of locks by using a set of key (XOR/XNOR) gates using an existing secure logic encryption technique [106], where the authors already has investigated how/where to insert these gates. The circuit produces functionally correct output when it receives a chip unlock key (CUK). The number of XOR or XNOR gates depends on the level of security one wants to achieve. We now modify the gate level netlist to enable manufacturing tests before the activation of chips.

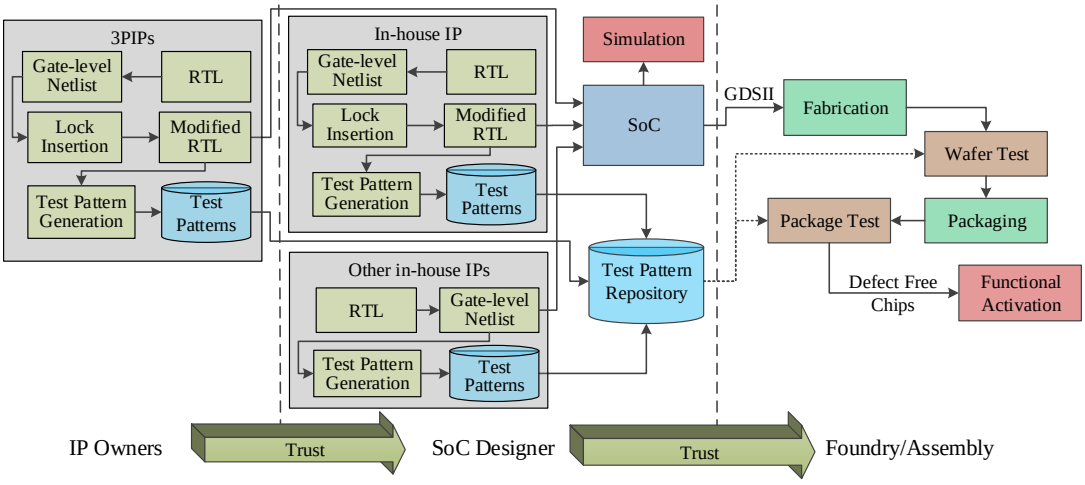


Fig. 5.3: FORTIS for enabling IC/3PIP metering to ensure forward trust in the SoC design and fabrication.

Each 3PIP owner inserts key gates to lock their design and then generates test patterns. The SoC designer receives all these locked IPs and integrates them in the design. The SoC designer also inserts a lock in one of the in-house IP to protect against IC overproduction. The SoC designer collects all the test patterns from different IP owners and stores them in a pattern repository for future wafer and package tests. As all the 3PIPs are locked, the simulation may

be a challenge for an SoC designer. We address this in Section 5.3.5.

The GDSII file corresponding to the SoC design is now sent to the foundry. The foundry first processes wafers, which generally contains hundreds of dies in a single wafer. Foundry then performs wafer test to inspect dies to find gross defects. If there are too many dies on a wafer that are defective, the foundry sometimes rejects the whole wafer. After wafer tests, the defect-free dies are sent to assembly for packaging. The good chips are then sorted out by using package tests and the chips that have been damaged during the packaging process are discarded. Our proposed design flow does not modify the existing fabrication, packaging and test processes. Finally, each chip is unlocked using a valid CUK by the entity who perform the final manufacturing test (foundry, assembly, or SoC designer) before supplied to the market.

5.3.2 Enabling Manufacturing Test before Chip Activation

It is absolutely necessary to activate the chips after the tests have been performed, which will prevent an untrusted foundry/assembly to pile up defect free ICs by hiding actual yield to the SoC designer. In this section, we will present an architecture that enables structural tests before the activation of chips.

In the previously proposed architectures, the structural test patterns are generated considering a predetermined CUK value. This is due to the existent of forward implication of the key gates. A forward implication exists when the inputs of a logic gate are assigned in a way that the output is uniquely identified [123]. For a two input XOR gate, the other input must be specified to either 1 or 0. If we do not assign a value at $CUK[i]$, the ATPG tool will consider this input as X, and all the faults before the gate k_i (logic cone shown in shaded grey color) will be untestable due to the non-existence of the forward implication.

Let us illustrate this point with an example by considering a fault D, shown in Figure 5.4(b). This fault will be testable if it is being propagated to the output Y_{1m}. If $CUK[i]$ is 1 then the output of the gate k_i becomes $\bar{D}$, otherwise it becomes D. The corresponding Y_{1m} will be $\bar{D}$ or D depending on the $CUK[i]$.

$$Y_{1m} = \begin{cases} D & if\ CUK[i] = 0 \\ \bar{D} & if\ CUK[i] = 1 \end{cases}$$

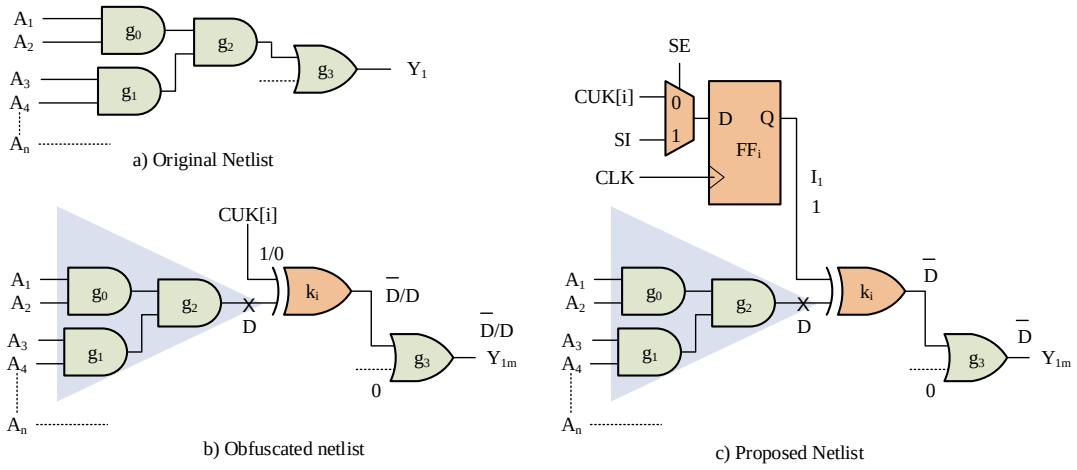


Fig. 5.4: Modification of an obfuscated netlist to enable manufacturing test before the functional activation of chips.

To maintain a forward implication, we need to provide a CUK value during test pattern generation. Thus, the previously proposed designs need a CUK (for example, $CUK[i] = 0$ or $CUK[i] = 1$) before the structural test pattern generation to test all the faults before the key gate. It is now necessary to load the same key into the chips before the manufacturing test begins. If we activate the chips before manufacturing tests then the objective of preventing overproduction will be unsatisfied. An untrusted foundry/assembly can overbuild chips by asking for more keys and reporting a lower yield to the SoC designer.

In our proposed netlist, the ATPG tool assigns a unique value at the I_1 input to maintain the forward implication (assign 1 for example) for the key gate to transfer the fault $\bar{D}$ to the output Y_{1m}. Thus, the ATPG tool can generate test patterns without knowing the key. In this thesis, we refer structural or scan test patterns as patterns. These patterns will be used later during wafer and package tests to find defect free chips from a manufacturing unit.

Figure 5.4(c) shows our proposed netlist, where the key bit $CUK[i]$ is connected to a scan flip-flop (FF_i). The output of FF_i drives the key gate k_1. In the test mode, when the scan enable (SE) signal is asserted, this flip-flop becomes a part of the scan chain. The ATPG tool generates test pattern for this modified netlist with $n+1$ inputs $(A_1, A_2, ..., A_n, I_1)$ rather than the original netlist (Figure 5.4(a)) with n inputs $(A_1, A_2, ..., A_n)$ or obfuscated netlist (Figure 5.4(b)) with n

inputs $(A_1, A_2, ..., A_n)$ and $CUK[i] = 0/1$.

Let us now consider key sensitization attack presented in [106]. In key sensitization attack, the key bits are treated as Xs and propagated to the output. As the unlocked chips contain 0 or 1 at a key bit location, these key values are visible at the output and the attacker can recover the key. For traditional DFT, where there is no compaction of test responses, the key sensitization attack works. However, this attack may not be feasible in any design, which uses an on-chip test response compaction module. On-chip test response compaction is very common in today's designs [108–110]. Almost every chip uses response compaction to significantly reduce test data not out of preference, but of necessity.

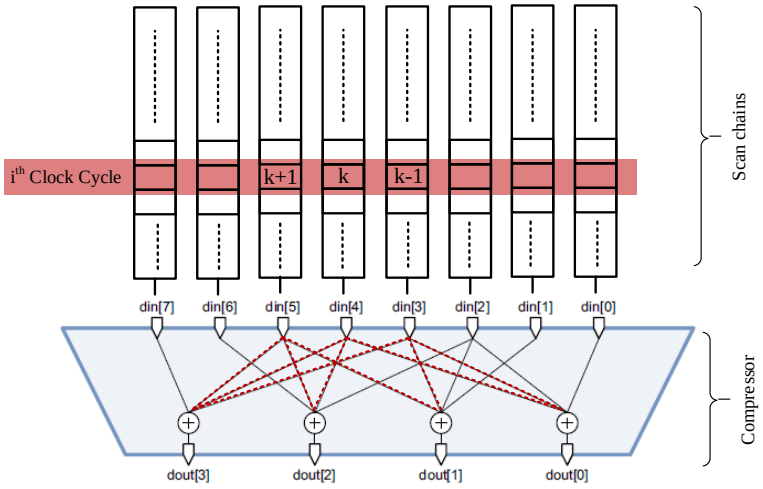


Fig. 5.5: An example of the compressor logic structure for 8-to-4 compressor [2].

Figure 5.5 shows an example of a compressor logic structure with a compression ratio 2. The effect of Xs (FFs captured the key bits) will be suppressed at the output $dout$ if at least two of the inputs of the XOR gates in the compressor are Xs. In this example, we can select scan chain 3,4, and 5. At i^{th} clock cycle three key bits $(k-1, k, k+1)$ will be at $dout$ simultaneously and their individual effect cannot be separated.

$$dout[0] = din[4] \oplus din[3] \oplus din[2] \oplus din[0] = k \oplus (k-1) \oplus \ldots = X \oplus X \oplus \ldots$$
$$dout[1] = din[5] \oplus din[3] \oplus din[2] \oplus din[1] = (k+1) \oplus (k-1) \oplus \ldots = X \oplus X \oplus \ldots$$
$$dout[2] = din[6] \oplus din[5] \oplus din[4] \oplus din[2] = (k+1) \oplus k \oplus \ldots = X \oplus X \oplus \ldots$$
$$dout[3] = din[7] \oplus din[5] \oplus din[4] \oplus din[3] = \ldots \oplus (k+1) \oplus k \oplus (k-1) = \ldots \oplus X \oplus X \oplus X$$

The key propagation will be failed as there is no forward implication for these XOR gates. Thus, by selecting the scan chains carefully and place key gates at the same location on these scan chains, we can circumvent key sensitization attack.

One could argue that the diagnostics done for failure analysis may be impacted due to the compressed test responses. However, modern EDA tools provide diagnostic support (high defect coverage and accurate fault diagnostics) with compression in place [108–110]. The compacted responses collected during the test can be used for diagnostics without going back to the traditional DFT (without compressions). So with this added feature, we do not see any reason why the SoC designers will not use test compression in their SoCs.

It is worthwhile to mention here that our proposed key insertion flow does not impact the test process using JTAG [124] in the field as the test patterns are generated after the insertion of the key gates and has no impact on CUK. No modifications to the design are made after test pattern generation.

5.3.3 Communication Flow of FORTIS for Preventing IC Overproduction

The success of the proposed design flow lies in the secure transfer of CUKs to the chips without interception by any untrusted entity in the supply chain. In the following, we will describe the transfer of CUK from SoC designer to the chips to prevent IC overproduction by the untrusted foundry. Then we will extend this communication protocol from 3PIP owners to prevent IP overuse by the untrusted SoC designer.

To ensure the safe transfer of CUK from the SoC designer to the chips, the following are required:

- *Message integrity*: The SoC designer must ensure the integrity of the request received from the chips. If the SoC designer detects an altered request, either modified by an

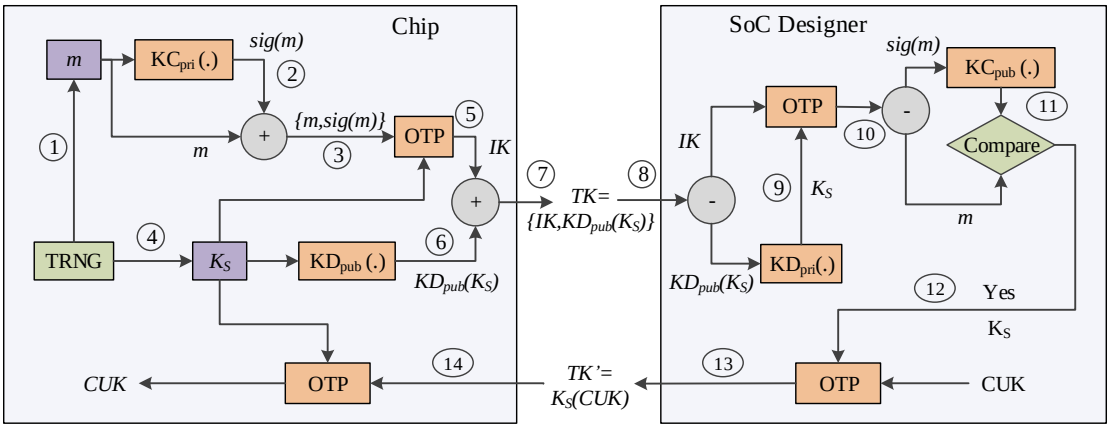


Fig. 5.6: Architecture and communication flow of FORTIS to prevent IC overproduction.

attacker or errors in the transmission, it is necessary to stop the transmission of the encrypted CUKs.

- *End-point authentication*: The SoC designer must verify that the request was initiated by the chips and not by an untrusted foundry or any other entity in the supply chain. As the chip cannot communicate by its own, the foundry only gets the information from the chip and forwards it to the SoC designer.

- *Confidentiality*: Only the SoC designer and the chip should understand the contents of the transmitted messages.

All these can be achieved by using a combination of asymmetric and symmetric key encryption. The widely used Rivest-Shamir-Adleman (RSA) algorithm [125] is used as the asymmetric key encryption algorithm to provide message integrity and end-point authentication. Note that Discrete logarithm or elliptic curve algorithms [121] can also be used instead of RSA. Depending on the area budget, one can select one of the algorithms from the above. One-time-pad (OTP) [121] is used for symmetric key encryption to provide the confidentiality. OTP has low area overhead as it only requires a simple XOR network for the encryption and decryption.

Figure 5.6 shows our proposed protocol to securely transfer CUK from SoC designer to the fabricated chips. To achieve this we need the keys (public key of the SoC designer (KD_{pub}) and private key (KC_{pri}) of the design) to be embedded in the design. Thus all the fabricated

chips have the same CUK, KD_{pub}, and KC_{pri}. The SoC designer has the other two keys, KD_{pri}, and KC_{pub}. The steps for transferring the CUK from the SoC designer to the chip are listed below:

1) The on-chip TRNG generates a message (m) which is unique for each and every chip.

2) The message m is encrypted with the private key KC_{pri} stored in the chip to form a signature, i.e., $sig(m) = KC_{pri}(m)$. This signature will be used to validate message integrity and verify end-point authentication.

3) The message m and its signature $sig(m)$ are concatenated.

4) The TRNG generates a random session key (K_S), which is unique for every communication. This session key can be stored in a non-volatile memory for future decryption to receive CUK. If the entire activation is performed while the chips are powered on, we can even store K_S in a volatile memory. This unique session key helps us to prevent replay attacks.

5) A one-time-pad (OTP) encrypts the concatenated message (m) and its signature ($sig(m)$) with K_S.

$$IK = K_S(\{m, sig(m)\}) = K_S \oplus \{m, sig(m)\}$$

6) The session key, K_S, is encrypted with the public key, KD_{pub}, of the SoC designer.

7) The transmission key is formed by concatenating encrypted K_S and IK. $TK = \{KD_{pub}(K_S), IK\}$. The foundry receives TK from the chip and forwards it to the SoC designer.

8) Upon receiving the TK from the foundry, the SoC designer separates encrypted K_S and IK.

9) Session key K_S is retrieved by decrypting $KD_{pub}(K_S)$ with KD_{pri}.

$$K_S = KD_{pri}(KD_{pub}(K_S))$$

10) A one-time-pad is used to decrypt IK to retrieve the concatenated m, and its signature $sig(m)$.

$$IK \oplus K_S = K_S \oplus \{m, sig(m)\} \oplus K_S = \{m, sig(m)\}$$

11) The SoC designer retrieves the message from the signature by using chip's public key, KC_{pub}.

$$KC_{pub}(sig(m)) = KC_{pub}(KC_{pri}(m)) = m$$

12) A comparison is performed to match m and decrypted signature $sig(m)$. This step verifies the integrity of m and end-point authenticity. The SoC designer now knows that the TK is originally coming from the chip if m equals to the $KC_{pub}(sig(m))$, not from an attacker.

13) After verifying the authenticity of the sender, the SoC designer encrypts CUK by using an OTP with the session key K_S and sends another transmission key (TK') to the foundry.

$$TK' = K_S(CUK) = K_S \oplus CUK$$

14) The foundry applies this TK' to the chip. The chip now reconstructs the correct CUK after decrypting TK' by using the OTP with its stored session key, K_S.

$$K_S(TK') = K_S \oplus CUK \oplus K_S = CUK$$

This correct CUK is then stored in a non-volatile memory (NVM) [126] to provide inputs to the key gates. The size of the NVM depends on the size of the CUK. One needs to make sure that the CUK values are not accessible by the JTAG [124] in the field.

5.3.4 Architecture of FORTIS for Preventing IP Overuse

The overuse of IP occurs when an SoC designer makes a foundry manufacture extra chips (including IC overproduction) without the knowledge of the 3PIP owners, which results in a loss of revenue. In this section, we will present an approach to prevent 3PIP overuse.

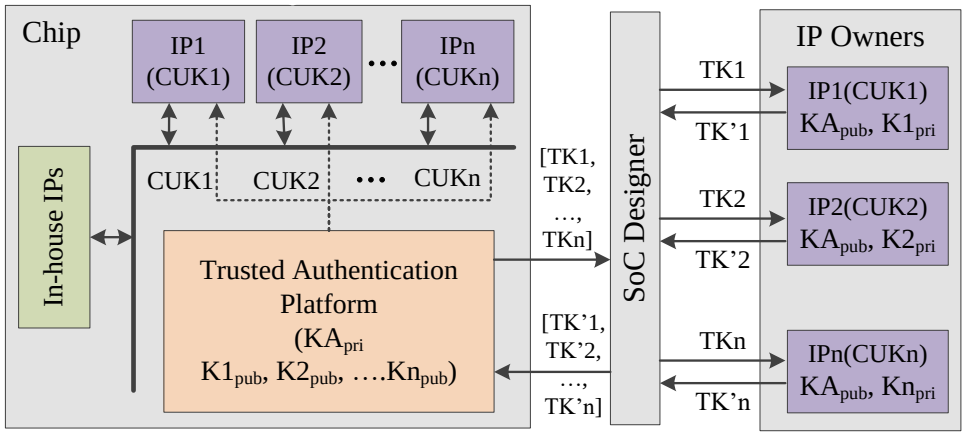


Fig. 5.7: Architecture of FORTIS to prevent IP overuse.

Figure 5.7 shows our proposed FORTIS architecture to prevent IP overuse. The IC contains a trusted authentication platform (TAP), which is introduced in the SoC design in order to reduce the area of each 3PIP by eliminating individual encryption/decryption blocks for each IP block, and is trusted by all the 3PIPs in that SoC. In addition, TAP can be encrypted by our propose approach (see Section 5.3.5) such that inner details are hidden to the SoC designer and it is modification resistant. The connection details between the TAP and 3PIPs are also obfuscated by the EDA tool such that SoC designers cannot add additional circuitry to observe $CUKs$ and provide them to the 3PIPs directly. Note that, we assume trusted EDA tools throughout this chapter and it cannot be modified to get an undue advantage by the SoC designers.

Each IP contains a lock (i.e., the key gates) which can only be unlocked by using the correct chip unlock key CUK_i of IP i. This CUK_i is only known by the i^{th} IP owner. The IPs only receive CUK_is from the TAP for the activation. TAP holds its own private key (KA_{pri}) and public keys ($\{Ki_{pub}\}$) for all the IPs in the design. TAP generates the transmission keys ($TK1$, $TK2$, ..., TKn) and sends them to the SoC designer. The SoC designer forwards each transmission key (TKi) to the corresponding IP owner. In return, the IP owners send the encrypted chip unlock key ($TK'i$) to the SoC designer. Upon receiving all the $TK'i$s from the IP owners, the SoC designer sends them to the foundry to unlock each IP in the fabricated chips.

Figure 5.8 shows the generation of transmission keys by the trusted authentication platform. TAP has a built-in TRNG, which generates a message (m) and separate session keys (K_S) for all different IP owners. First, the signature of m is generated and then concatenated with its signature. This ensures the message integrity and end-point-authentication for all the IP owners and also that the request is indeed coming from the TAP and not from a tampered TAP used by an attacker. TAP then generates one transmission key in each step. At step 1, a session key (K_{S1}) for IP owner 1 is obtained from the TRNG. This session key helps to encrypt $\{m, sig(m)\}$ and the encrypted output is concatenated with the encrypted K_{S1} to form $TK1$. At step 2, a different session key (K_{S2}) for IP owner 2 is received from the TRNG. This session key is then used to encrypt $\{m, sig(m)\}$ and the encrypted output is concatenated with the encrypted K_{S2} to form $TK2$. In a similar fashion, all the transmission keys (TKi) are generated. Then the foundry receives all the TKi, sends them to the SoC designer, and waits for the encrypted $CUKs$.

After receiving the transmission keys ($TK'i$s), the foundry applies them to the TAP. TAP

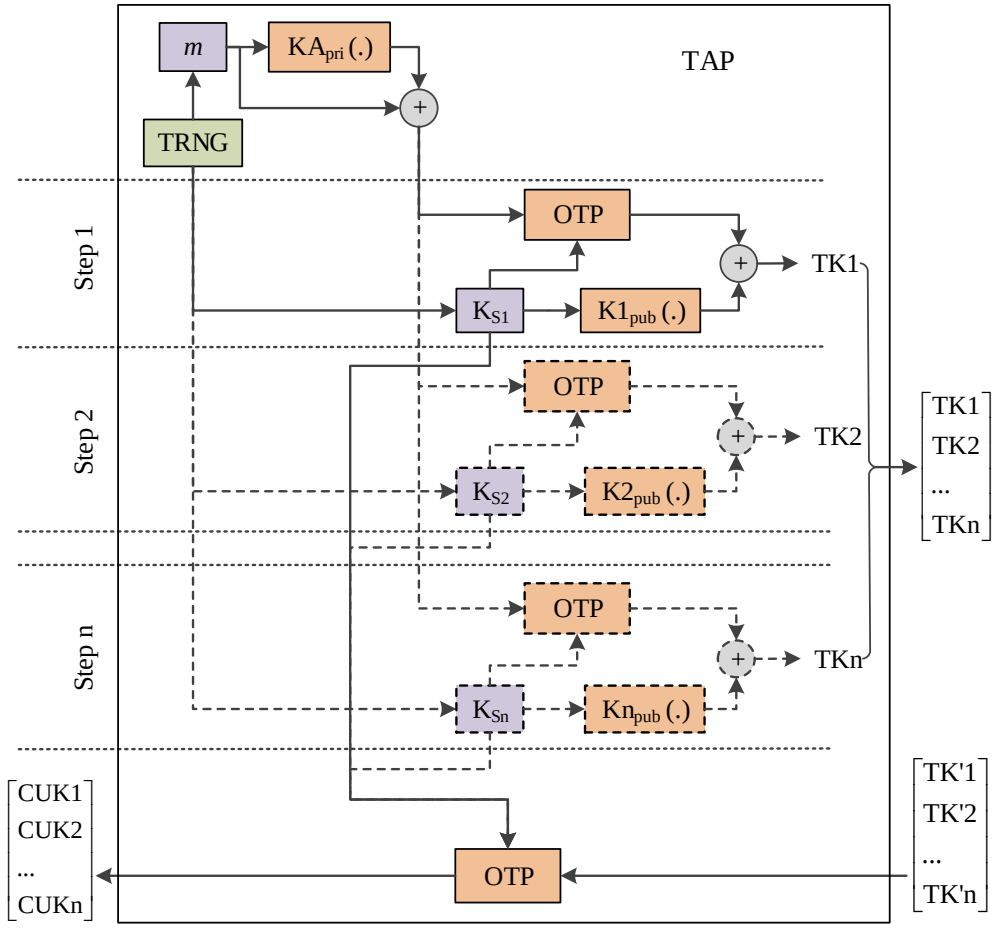


Fig. 5.8: Architecture of TAP and communication flow to reconstruct $CUKs$ for all 3PIPs in an SoC.

decrypts these $TK'is$ by using its session keys, K_Ss, to generate the chip unlock keys, CUK_is, for all different IPs.

5.3.5 FORTIS for Preventing IP Piracy

To establish a forward trust between the IP owners and SoC designers, foundries, and assemblies, it is necessary to add security measures in the IP to prevent IP piracy, such as, cloning, and modification of IPs by untrusted SoC designers and foundries. FORTIS inherently prevents the cloning of IPs. As each IP is locked by using a set of key gates, even if the attackers copy the netlist completely, they cannot unlock it without the proper CUK. However, simulation of

an SoC having these locked 3PIPs needs to be addressed, as these IPs will work properly only upon receiving a proper *CUK*. At the same time, it is necessary to protect these *CUKs* from the SoC designer. Otherwise, there is no point of adding them into the IPs in the first place. Our objective of simulating a 3PIP will be successful if we provide a *CUK* securely to the simulation tool without interception by the SoC designer.

We also propose IP integrity verification to prevent IP modification by the SoC designer. We use a cryptographic hash function [114] to create an IP digest (see message digest [121]) to make it resistant against modification. Any modification, including addition or deletion of extra features, to a 3PIP will result in a different IP digest than the original one, which can easily be detected by comparison in an EDA tool.

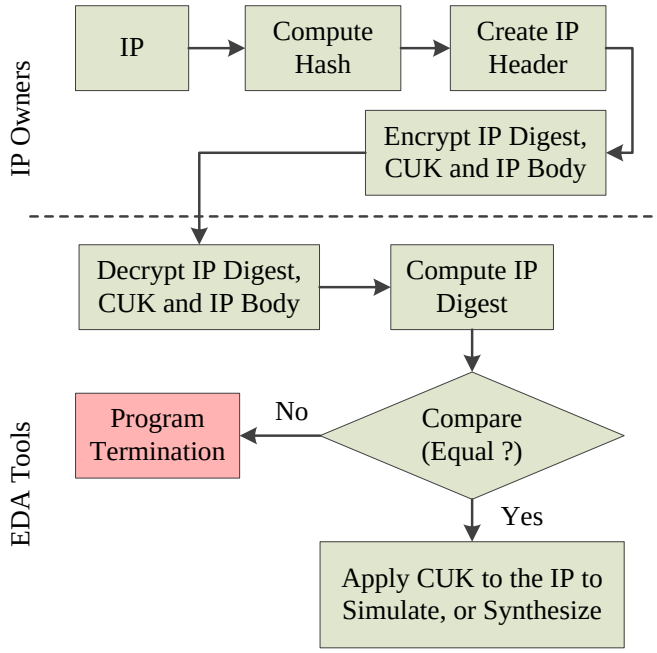


Fig. 5.9: Proposed flow to prevent IP piracy integrated into FORTIS.

Figure 5.9 shows our proposed flow to prevent cloning and modification of 3PIPs. The IP owners first compute IP digest which is the hash of the entire locked netlist. An IP header is the created which contains the *CUK* for the simulation of an SoC and the IP digest. The IP is then encrypted (the code inside the *'pragma protect* blocks) by using a symmetric encryption

method (e.g., Advanced Encryption Standard - Cipher Block Chaining (AES-CBC) [127]) recommended in encryptP1735.pl script [112]. This symmetric key is now encrypted by the public keys of different EDA vendors such that these vendors can later on decrypt them to get the IP.

We propose a new IP digest comparison flow during synthesis and simulation of SoCs. The EDA tool first needs to decrypt the encrypted portion of the IP header and the IP body. An IP digest has to be calculated from the decrypted IP by using the same hash function used before to form the IP digest. A comparison needs to be performed with the IP digest retrieved from the IP header and newly computed IP digest. If both of them are equal, then it is ensured that no modifications to the program has been made, otherwise, the program has to be terminated.

a) Locked IP.

b) Locked and encrypted IP.

Fig. 5.10: IP header insertion for the simulation of a locked IP.

Figure 5.10 shows an example of our proposed encrypted IP. We use SHA-512 [114] to form an IP digest, which is attached to the IP header along with the *CUK*. We use Synopsys encryptP1735.pl script [112] to encrypt the IP header and IP body. Figure 5.10(a) shows a locked IP. The encryption is carried out in two parts - (i) The IP vendor encrypts the IP data (data block) using its own symmetric key which is called the data key. We use aes256-cbc as symmetric encryption algorithm to encrypt the data block. (ii) The IP vendor then encrypts

the data key with its public key by using asymmetric encryption to create a key block. The encryption version, encode type, key owner, key name and key method need to be mentioned. We use RSA as asymmetric encryption to generate the key block which is attached to the IP header (see Figure 5.10(b)).

5.4 Results and Analysis

5.4.1 Test Metrics Analysis

The objective of analyzing the test metrics is to provide an evidence of the impact of our proposed modification of the locks to enable manufacturing test before the activation of chips (see Section 5.3.2). The proposed architecture is evaluated with implementations on large benchmark circuits from ITC99 [128], opencores.org [129], and OpenSPARC T1 processor core [130]. In this evaluation, we have used SAED 32/28nm library [131] provided by Synopsys to synthesize all benchmark circuits. Each synthesized benchmark circuit is then locked with a 128 bit CUK. A total of 128 XOR/XNOR gates along with 128 D-flip flops are inserted for each synthesized benchmark circuit. Scan-chain insertion is performed on both unsecured and secured versions of the same circuits to evaluate and compare test metrics, such as test pattern count and test coverage. A comparison between the unlocked and locked versions of these benchmarks are presented in Table 5.2.

Table 5.2: Test Metrics Comparison.

		Pattern Count			**Test Coverage**		
Benchmark	**Gate Count**	**Before**	**After**	**Change (%)**	**Before**	**After**	**Change (%)**
des	16,341	60	59	-1.67	100.00	99.78	-0.22
b18	25,397	404	388	-3.96	99.53	99.45	-0.08
ethernet	30,534	627	629	0.32	99.94	99.84	-0.10
b19	40,789	424	418	-1.42	99.34	99.40	0.06
vga_lcd	43,346	1,710	1,721	0.64	100.00	99.94	-0.06
leon3mp	253,050	3,821	3,848	0.71	100.00	99.99	-0.01
sparc	836,865	3,220	3,185	-1.09	99.16	99.09	-0.07

As shown in Table 5.2, the gate count of investigated benchmark circuits range between 16,341 to 836,865. We chose these larger benchmark circuits to better represent typical large industrial designs. We also present the pattern count and test coverage figures of the implemented designs before and after applying our proposed architecture, in order to present the impact of proposed architecture on circuit testability. We do not expect any major change in test pattern count and test coverage, but there may be a minor improvement to test coverage as each key gate with a D flip-flop adds a test point in the design. On the other hand, the XOR/XNOR gates and D flip-flops create additional faults in the netlist, which may lead to the reduction of test coverage. The shown pattern count change range between -3.96% and 0.71%, which means at worst the proposed architecture would result in less than 1% overhead in pattern length. Similarly, the change in test coverage ranges between -0.22% and 0.06%. Both impacts are minor and would not significantly impact testability of the secured design.

5.4.2 Area Overhead Analysis

The area overhead of our proposed FORTIS consists of:

(i) RSA module: The RSA module used in our proposed design to encrypt the session key and generate the signature makes up a major part of the area overhead. This area can be reduced significantly depending on the speed of operation. As speed is not our major concern, we can select a slower, but more area efficient RSA module. It is reported that a minimum size RSA datapath can be implemented by using only 861 gates [132].

(ii) OTP module: The size of the one-time pad depends on the size of the CUK. For a 128 bit CUK, we need 128 XOR gates. The same OTP can be used for the encryption of $\{m, sig(m)\}$ and decryption of TK'.

(iii) Keys gates: The size due to keys also depends on the CUK. To implement one key bit, we need one XOR/XNOR gate and a scan flip-flop.

(iv) RSA Keys: Extra storage or logic is needed to keep or generate at least 1024 bit KC_{pri} for chips or KA_{pri} for TAP. We can simply neglect the size of the public keys (KD_{pub} or $Ki_{pub}s$) as they can be as small as number 3 or 17 [121].

(v) TRNG: A single TRNG is used for generating the message, m and session keys, K_Ss. We propose the use of an area efficient cryptographically secure pseudorandom number

generator [133] or [134] depending on the implementation choice.

(vi) Non-volatile memory: The size of the non-volatile memory depends of the session keys, K_Ss. We need a 128 bit non-volatile memory to store a 128 bit K_S.

There is no area overhead of any 3PIPs for preventing IP overuse except for the key gates. The trusted authentication platform (*TAP*) provides the $CUKs$ to all different 3PIPs. The primary motivation for implementing TAP in any design for an SoC designer is that they need to prevent IC overproduction.

Considering all these modules, the total gate count is approximately 10K. Table 5.3 shows the overhead analysis. For benchmark circuits, it ranges from 24.52% to 1.19%. However, for industrial designs it becomes less than 1%. For Xilinx Artix-7 and Kintex-7 [135] the overhead becomes 0.77% and 0.15% respectively. It becomes negligible for Virtex-7 [136]. The area overhead may further be reduced if the original design already contains a TRNG and a RSA module, as is the case for most of the industrial designs.

Table 5.3: Area overhead analysis.

	Design	**Gate Count**	**Overhead (%)**
Benchmarks	b19	40,789	24.52
	vga_lcd	43,346	23.07
	leon3mp	253,050	3.95
	sparc	836,865	1.19
Industrial Designs	Artix-7	1.3M	0.77
	Kintex-7	6.6M	0.15
	Virtex 7	20M	0.05

5.4.3 Security Analysis

The security of our proposed protocol is of prime importance to prevent the overproduction of ICs and overuse of 3PIPs. In the following, we will perform the security analysis of our proposed approach.

Exhaustive key search: The length of a chip unlock key, CUK, should be long enough such

that it can withstand exhaustive key search or brute-force attacks. We need to achieve at least 80 bits of security as this is the lower minimum requirement for exhaustive key search [121]. To achieve this, we require 80 key gates (XOR/XNOR). However, the key size may be increased up to 256 bits for higher security, which will hardly impact the overall area of a modern design.

Encryption: In our approach, we use RSA to encrypt the session key and generate signature. One can achieve 80 bit of security while the key length is 1024 bits. However, 128 bit security can be achieved with the key length of 3072 bits [121]. Depending on the area budget one can select a desired security level of n bits. We have used one-time-pad to encrypt $\{m, sig(m)\}$. As the session keys, K_Ss, are generated from a TRNG, a perfect secrecy can be achieved. Thus, we can achieve an overall RSA equivalent secrecy in our proposed protocol.

Man-in-the-middle attack: As the key-pairs for the RSA are generated by the IP owners and reside in the circuit, no key transfer is required. This prevents an attacker (e.g., untrusted foundry) from becoming a man-in-the-middle.

Replay attack: In this attack scenario, the attacker copies a message between two parties and replays that message to one or more of them. Our proposed protocol is inherently resistant to replay attacks as a new session key, K_S, is generated every time during encryption. Every time, the encrypted message will be different from the previous one. In addition, the message (m) is unique for every chip, which also helps to make a unique transmission key for every chip.

Reverse engineering: As we use a secure logic encryption technique [106], it is extremely hard for an attacker to find CUK by reverse engineering. Even if we assume that reverse engineering is possible to find the key, an attacker cannot feed the CUK to a chip, as they do not know the private key of the SoC designer (KD_{pri}) to retrieve K_S. As the session key, K_S, is unique for every chip, it is not economical for the attackers to retrieve K_S for each chip by reverse engineering. We also assumed that the attacker cannot model the TRNG to predict its output after observing certain K_Ss. Finally, we believe that it is extremely expensive to perform reverse engineering for modern designs manufactured with $22nm$ or lower technology nodes.

Tampering RSA Keys: In this attack scenario, an untrusted foundry reconstructs new masks to replace the keys, KC_{pri} and KD_{pub}, with its own. This enables the foundry to unlock unlimited number of chips when it receives the CUKs from the IP owners. Fortunately, this attack can

easily be prevented by the IP owners. The SoC designer can request only one locked chip and then verify the correct keys. If the foundry replaces KC_{pri} and KD_{pub} by its own, the SoC designer will not be able to unlock the chip and consequently, it can detect mask modification.

Tampering TRNG: An untrusted foundry can modify the masks to bypass the TRNG and write a permanent value for K_Ss and m. Once it knows the CUK, it can unlock any number of chips. Fortunately, this attack can also be detected by the IP owners and can be prevented. Like before, the SoC designer can request few locked chips to monitor the message, m and the session key, K_S. If either m or K_Ss from these chips are the same or biased, it will definitely be the indication of the tampering of TRNG. As it is extremely expensive to design a new set of masks, there is little economic incentive for an untrusted foundry to manufacture a product with two different set of masks.

Tampering IP Digest: In this attack scenario, the attacker tries to tamper the IP digest by replacing the original IP digest with the tampered IP digest. Fortunately, this tampering can be detected. As the attacker does not have the private key of the EDA tool (we assume a trusted EDA tool for synthesis and simulation), he cannot reconstruct the original IP from its encrypted version. If the attacker try to modify the IP and then compute the digest, it will be different than the original one.

5.5 Summary

In this chapter, we have presented FORTIS, a comprehensive solution for establishing forward trust for different entities involved in the SoC design and manufacturing process. FORTIS uses a novel communication protocol between the fabricated chips and the SoC designers/IP owners to activate the chips for preventing IP overuse and IC overproduction. FORTIS uses an existing logic encryption technique to obfuscate the netlist of an SoC or a 3PIP and can only be unlocked upon receiving a correct chip unlock key (CUK). A modification is proposed to the existing obfuscated netlist to enable manufacturing tests before the activation of chips which is one of the key requirement to prevent IC overproduction and IP overuse. Our proposed modification does not have any impact on manufacturing test process.

To address IP overuse, we have introduced a trusted authentication platform in the SoC.

This TAP is trusted by the all parties involved in the SoC design and can be synthesized and placed in an SoC by a trusted EDA tool automatically. To the best of our knowledge, the metering approach we have presented to prevent IP overuse is the first in the literature. The encrypted IP with additional IP digest check prevents the SoC designer from IP piracy. As an IP is locked by using a set of key gates, even if the attackers copy the entire netlist, they cannot make it work properly without a correct CUK, which prevents IP cloning. Finally, our proposed design flow is resistant to all known attacks.

Chapter 6

Conclusion

Integrated circuits are becoming increasingly vulnerable to counterfeiting and piracy due to the globalization of the electronic component supply chain. To address this issue, we have presented the complete treat space and evaluated the detection capability of the existing test regimen for counterfeit detection. We have presented several design-for-anti-counterfeit measures for to improve the trustworthiness, security, and reliability of different ICs. In this chapter, we present a detailed summary of our major contributions, and then present the future research direction for ensuring the security of the component supply chain.

6.1 Summary of Contributions

The detection and prevention of counterfeit components in the component supply chain becomes a major challenge that needs to be addressed in the near future. In this thesis, we have systematically addressed the problems of counterfeit ICs by - (i) presenting different taxonomies of counterfeit types, defects, and test methods; (ii) proposing test metrics, and developing a comprehensive framework for the assessment of test methods; and (iii) developing different design-for-anti-counterfeit measures for all different counterfeit types.

We have classified the counterfeit ICs into seven distinct categories: recycled, remarked, overproduced, out-of-spec/defective, cloned, forged documentation, and tampered. Our counterfeit method taxonomy describes all the test methods currently available for the detection of counterfeit ICs. The test methods are broadly classified into two distinct types: physical tests and electrical tests. Counterfeit defects are those anomalies and changes that are not typically found in authentic parts. Anomalies vary based on size, shape, type, number, etc., depending on the capabilities possessed by the counterfeiters. The detection of one or more anomalies may

be an indication of a component being counterfeit. Counterfeit defects are divided into four categories: procedural, mechanical, environmental, and electrical.

We have proposed different test metrics for evaluating the capability of the test methods for counterfeit IC detection. These metrics are counterfeit defect coverage (*CDC*), counterfeit type coverage (*CTC*), under-covered defects (*UCDs*) and not-covered defects (*NCDs*). We have developed a comprehensive framework to assess the test methods. We have developed a web-based tool, *Assessment Framework*, for this purpose. The tool is currently deployed in the server of University of Connecticut's CHASE Center. One can access the tool at $http://ece-chaseweb.engr.uconn.edu/cdc-site/index.php$ with proper user credentials. The tool provides two options - (i) *static assessment* where a user can evaluate a preexisting test plan based on the aforementioned metrics; and (ii) *dynamic assessment* where the user finds an optimum set of test methods to maximize *CDC*.

We propose different lightweight structures for combating die and IC recycling (CDIR) as a part of design-for-anti-counterfeit measures. These structures are of ring-oscillator (RO)-based NBTI-aware, which exploits aging much more efficiently. The reference ROs in these CDIRs remain quiet during the normal operation of the IC while the stressed RO gets aged at an accelerated pace utilizing NBTI of PMOS transistors. This helps to get a reasonable frequency difference between the reference and stressed ROs even though an IC is used only a very short duration. We proposed two different versions of CDIRs with multiple RO-pairs where the designer can select the number of RO-pairs depending on their area budget. In addition, we also propose fuse-based CDIRs (F-CDIRs) can be implemented in any components (small, or large, and analog, or digital) and any technology node. These CDIRs can authenticate ICs very effectively and require a very low cost multimeter.

We present FORTIS, a comprehensive solution for protecting IPs and ICs by assuring forward trust between all entities involved in the SoC design and fabrication process. We propose a novel design flow to prevent IC overproduction and IP overuse. We use an existing logic encryption technique to obfuscate the netlist of an SoC or a 3PIP and propose a modification to enable manufacturing tests before the activation of chips which is absolutely necessary to prevent overproduction. We also propose to attach an IP digest (a cryptographic hash of the entire IP) to the header of an IP to prevent modification of the IP by the SoC designers.

6.2 Future Work

6.2.1 Assessment of Test Methods

Chapter 1 briefly describes the defects and anomalies present in different counterfeit components. These four categories - procedural, mechanical, environmental, and electrical - of defects are originated either from the counterfeiting process or the deliberate test escapes after manufacturing. The detection of one or more defects necessarily increases the confidence of a component being counterfeit (Chapter 2). The question now becomes how efficiently we can detect them. We need to monitor these defects very carefully, as counterfeiting is an evolving problem. Counterfeiters are adopting new technologies to improve their process, and the shape and size of these defects are also evolving. The easy to detect defects may not be present in the counterfeit components in near future. The progress of the detection capability by our test methods are needed to monitored to cope with this evolving problem.

The detection of counterfeit parts is still in infancy and the data for most of these key elements used in *Assessment Framework (Chapter 2)* does not exist today or change continuously. First, the number of test methods may increase over the period and their detection capability will change. Currently, the test cost and time data varies significantly in between test labs, thus, we take an average of them for each test methods during the simulation. Second, the confidence level (CL) matrix contains the values, generated by the subject matter experts (SMEs) participating in the G-19A group. Third, the defect frequency (DF) data does not exist today. However, we are hoping that, in the near future, this data will be available as the two reporting agencies ERAI [137] and GIDEP [138] are capturing the incidence of counterfeit parts worldwide. Forth, the decision index (DI) in Table 2.2 was again generated by the SMEs as we do not have enough information available regarding the visibility of different counterfeit types. Finally, it is important that the *Assessment Framework* should be flexible enough to handle changes in the input data regardless of its source. In future, we will continuously work on *Assessment Framework*.

6.2.2 Design-for-Anti-Counterfeit Measures

Different available technologies (see Figure 1.12) target different counterfeit types. Different types of CDIRs can effectively detect recycled components irrespective of their sizes. However,

there are no technologies available today to detect overproduced and out-of-spec/defective ICs, which belongs to the analog and mixed-signal types. As the analog and mixed-signal ICs are small in size, FORTIS cannot be applied to detect those ICs due its area overhead and different process technologies. DNA and NR can be used to detect remarked and cloned ICs. However, they are not free from various challenges and limitations. We further investigate new DfAC measures to detect the complete spectrum of counterfeit ICs.

6.2.3 Counterfeit Electronic Systems

In this thesis, we have not considered counterfeit electronic systems, which is growing in scope and magnitude in recent times. It is becoming increasingly difficult to assure the security and integrity of an electronic system due to the globalization of semiconductor supply chain. This is because the electronic systems today are assembled all across the globe and might consist of components sourced from the different parts of the world. This makes it virtually impossible to gauge the origin of these systems and their components, and track their route in the supply chain. Today, the design companies do not manufacture their own products. The companies responsible for manufacturing the products are located all across the globe. These offshore companies can easily copy the drawings and specifications of the products they are contracted to manufacture. This trend is very disturbing as these contract manufacturer can make clone products on a separate production line. In addition, they can also ship defective systems which did not pass system tests to different independent distributors. Numerous incidents have pointed out the far-reaching penetration of such systems into the electronics supply chain, which are presented in our book *Counterfeit Integrated Circuits: Detection and Avoidance [12]*. Thus, it becomes absolutely necessary to verify the authenticity of an electronic system. In future, our prime focus will be on developing solutions to detect these counterfeit systems and prevent them entering into the supply chain.

Bibliography

[1] X. Zhang, N. Tuzzio, and M. Tehranipoor, "Identification of recovered ics using fingerprints from a light-weight on-chip sensor," in *Proc. IEEE-ACM Design Automation Conference*, pp. 703 –708, June 2012.

[2] Synopsys, "DFT Compiler, DFTMAX^{TM}, and DFTMAX^{TM} Ultra User Guide," Septermber 2015.

[3] IHS iSuppli, "Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market," 2011, http://press.ihs.com/press-release/design-supply-chain/top-5-most-counterfeited-parts-represent-169-billion-potential-cha.

[4] OECD, "The Economic Impact of Counterfeiting and Piracy," 2007, http://www.oecd.org/dataoecd/13/12/38707619.pdf.

[5] D. Chardonnal, "Impacts of counterfeiting and piracy to reach US$1.7 trillion by 2015," February 2011.

[6] J. Cassell, "Reports of Counterfeit Parts Quadruple Since 2009, Challenging US Defence Industry and National Security," April 2012.

[7] U.S. Environmental Protection Agency, "Electronic waste management in the united states through 2009," May 2011.

[8] U. Guin, M. Tehranipoor, D. DiMase, and M. Megrdichian, "Counterfeit IC Detection and Challenges Ahead," *ACM/SIGDA E-NEWSLETTER*, vol. 43, no. 3, March 2013.

[9] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[10] U. Guin, D. Forte, and M. Tehranipoor, "Anti-Counterfeit Techniques: From Design to Resign," in *Microprocessor Test and Verification (MTV)*, 2013.

[11] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[12] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer, 2015.

[13] U. Guin, D. DiMase, and M. Tehranipoor, "A comprehensive framework for counterfeit defect coverage analysis and detection assessment," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 25–40, 2014.

[14] U. Guin and M. Tehranipoor, "On Selection of Counterfeit IC Detection Methods," in *IEEE North Atlantic Test Workshop (NATW)*, May 2013.

[15] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-cost On-Chip Structures for Combating Die and IC Recycling," in *Proc. of ACM/IEEE on Design Automation Conference*, 2014.

[16] U. Guin, D. Forte, and M. Tehranipoor, "Design of accurate low-cost on-chip structures for protecting integrated circuits against recycling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1233–1246, April 2016.

[17] B. Shakya, U. Guin, M. Tehranipoor, and D. Forte, "Performance optimization for on-chip sensors to detect recycled ics," in *IEEE International Conference on Computer Design (ICCD)*, pp. 289–295, Oct 2015.

[18] U. Guin, Q. Shi, D. Forte, and M. Tehranipoor, "FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2016.

[19] U. Guin, T. Chakraborty, and M. Tehranipoor, "Functional fmax test-time reduction using novel dfts for circuit initialization," in *IEEE International Conference on Computer Design (ICCD)*, pp. 1–6, Oct 2013.

[20] U. Guin, T. Chakraborty, and M. Tehranipoor, "Novel DFTs for Circuit Initialization to Reduce Functional Fmax Test Time," in *IEEE North Atlantic Test Workshop (NATW)*, May 2013.

[21] U.S. Department Of Commerce, "Defense Industrial Base Assessment: Counterfeit Electronics," January 2010.

[22] R. Torrance and D. James, "The state-of-the-art in ic reverse engineering," in *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES '09, pp. 363–381. Berlin, Heidelberg: Springer-Verlag, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04138-9_26

[23] I. McLoughlin, "Secure embedded systems: The threat of reverse engineering," in *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, pp. 729–736, Dec 2008.

[24] F. Koushanfar and G. Qu, "Hardware metering," in *Proc. IEEE-ACM Design Automation Conference*, pp. 490–493, 2001.

[25] G. Contreras, T. Rahman, and M. Tehranipoor, "Secure Split-Test for Preventing IC Piracy by Untrusted Foundry and Assembly," in *Proc. International Symposium on Fault and Defect Tolerance in VLSI Systems*, 2013.

[26] M. Tehranipoor, H. Salmani, and X. Zhang, *Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection*. Springer, 2014.

[27] SAE, "Counterfeit electronic parts; avoidance, detection, mitigation, and disposition," 2009, http://standards.sae.org/as5553/.

[28] SAE, "Test methods standard; counterfeit electronic parts," Work In Progress, http://standards.sae.org/wip/as6171/.

[29] CTI, "Certification for coutnerfeit components avoidance program," September 2011.

[30] IDEA, "Acceptability of electronic components distributed in the open market," http://www.idofea.org/products/118-idea-std-1010b.

[31] C. Mouli and W. Carriker, "Future Fab: How software is helping Intel go nano–and beyond," *IEEE Spectrum*, vol. 44, no. 3, pp. 38–43, 2007.

[32] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, January 2010.

[33] "Defense Science Board (DSB) study on high performance microchip supply (2005)," http://www.acq.osd.mil/dsb/reports/ADA435563.pdf.

[34] H. Levin, "Electronic Waste (E-Waste) Recycling and Disposal Facts, Statistics & Solutions," 2011, online. [Online]. Available: http://www.moneycrashers.com/electronic-e-waste-recycling-disposal-facts/

[35] L. W. Kessler and T. Sharpe, "Faked Parts Detection," *Printed Circuit Design & Fab*, vol. 27, no. 6, p. 64, 2010.

[36] J. Villasenor and M. Tehranipoor, "Chop shop electronics," *Spectrum, IEEE*, vol. 50, no. 10, pp. 41–45, 2013.

[37] M. Alam and S. Mahapatra, "A comprehensive model of pmos nbti degradation," *Microelectronics Reliability*, vol. 45, no. 1, pp. 71 – 81, 2005.

[38] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbti effect for reliable design," in *Proc. of IEEE on Custom Integrated Circuits Conference*, pp. 189 –192, September 2006.

[39] V. Reddy, A. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan, "Impact of negative bias temperature instability on digital circuit reliability," in *Proc. on Reliability Physics*, pp. 248 – 254, 2002.

[40] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Applied Physics*, vol. 94, no. 1, pp. 1 –18, July 2003.

[41] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "Compact modeling and simulation of circuit reliability for 65-nm cmos technology," *Device and Materials Reliability, IEEE Transactions on*, vol. 7, no. 4, pp. 509 –517, December 2007.

[42] K.-L. Chen, S. Saller, I. Groves, and D. Scott, "Reliability effects on mos transistors due to hot-carrier injection," *Electron Devices, IEEE Transactions on*, vol. 32, no. 2, pp. 386 – 393, February 1985.

[43] S. Mahapatra, D. Saha, D. Varghese, and P. Kumar, "On the generation and recovery of interface traps in mosfets subjected to nbti, fn, and hci stress," *Electron Devices, IEEE Transactions on*, vol. 53, no. 7, pp. 1583 –1592, July 2006.

[44] J. McPherson, "Reliability challenges for 45nm and beyond," in *Proc. of ACM/IEEE on Design Automation Conference*, pp. 176 –181, 2006.

[45] F. Jensen and N. E. Petersen, *Burn-In: An Engineering Approach to the Design and Analysis of Burn-In Procedures*. Wiley, December 1982.

[46] CHASE, "ARO/CHASE Special Workshop on Counterfeit Electronics," January 2013, http://www.chase.uconn.edu/arochase-special-workshop-on-counterfeit-electronics.php.

[47] R. Pappu, "Physical one-way functions," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[48] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. of ACM/IEEE on Design Automation Conference*, pp. 9 –14, June 2007.

[49] K. Kursawe, A.-R. Sadeghi, D. Schellekens, B. Skoric, and P. Tuyls, "Reconfigurable physical unclonable functions - enabling technology for tamper-resistant storage," in *Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 22 –29, July 2009.

[50] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly puf protecting ip on every fpga," in *Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 67 –70, June 2008.

[51] L. Bolotnyy and G. Robins, "Physically unclonable function-based security and privacy in rfid systems," in *Proc. of IEEE International Conference on Pervasive Computing and Communications*, pp. 211 –220, March 2007.

[52] X. Wang and M. Tehranipoor, "Novel physical unclonable function with process and environmental variations," in *Proc. on Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1065 –1070, March 2010.

[53] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proc. of 16th USENIX Security Symposium on USENIX Security Symposium*, pp. 20:1–20:16, 2007.

[54] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. of the 9th ACM conference on Computer and communications security*, ser. CCS '02, pp. 148–160. New York, NY, USA: ACM, 2002.

[55] G. E. Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas, "Aegis: architecture for tamper-evident and tamper-resistant processing," in *Proc. of the 17th annual international conference on Supercomputing*, ser. ICS '03, pp. 160–171. New York, NY, USA: ACM, 2003.

[56] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical unclonable functions and public-key crypto for fpga ip protection," in *International Conference onField Programmable Logic and Applications*, pp. 189–195, 2007.

[57] A. Hosey, M. T. Rahman, K. Xiao, D. Forte, and M. Tehranipoor, "Advanced analysis of cell stability for reliable sram pufs," in *Asian Test Symposium (ATS)*, pp. 348–353, 2014.

[58] A. Mazady, M. T. Rahman, D. Forte, and M. Anwar, "Memristor puf – a security primitive: Theory and experiment," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 5, no. 2, pp. 222–229, 2015.

[59] F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual property metering," in *Inform. Hiding*, pp. 81–95. Springer-Verlag, 2001.

[60] K. Lofstrom, W. Daasch, and D. Taylor, "Ic identification circuit using device mismatch," in *Proc. of IEEE International Solid-State Circuits Conference*, pp. 372 –373, 2000.

[61] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Proc. of Digest of Technical Papers on VLSI Circuits*, pp. 176 – 179, June 2004.

[62] Y. Su, J. Holleman, and B. Otis, "A 1.6pj/bit 96circuit using process variations," in *Proc. of IEEE International on Solid-State Circuits Conference*, pp. 406 –611, February 2007.

[63] R. Chakraborty and S. Bhunia, "Hardware protection and authentication through netlist level obfuscation," in *Proc. of IEEE/ACM International Conference on Computer-Aided Design*, pp. 674 –677, November 2008.

[64] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *Proc. of IEEE/ACM international conference on Computer-aided design*, pp. 674–677, 2007.

[65] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," in *Proc. on Design, Automation and Test in Europe*, pp. 1069 –1074, March 2008.

[66] J. Huang and J. Lach, "IC activation and user authentication for security-sensitive systems," in *Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 76 –80, June 2008.

[67] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 66 –75, Jan.-Feb. 2010.

[68] R. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493 –1502, October 2009.

[69] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. Tehranipoor, "Csst: Preventing distribution of unlicensed and rejected ics by untrusted foundry and assembly," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*, pp. 46–51. IEEE, 2014.

[70] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. Tehranipoor, "Csst: An efficient secure split-test for preventing ic piracy," in *Test Workshop (NATW), 2014 IEEE 23rd North Atlantic*, pp. 43–47, May 2014.

[71] J. Villasenor and M. Tehranipoor, "Are you sure its new? the hidden dangers of recycled electronics components," in *IEEE Spectrum*, 2012.

[72] X. Zhang and M. Tehranipoor, "Design of on-chip lightweight sensors for effective detection of recycled ics," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 5, pp. 1016–1029, May 2014.

[73] K. Arndt, C. Narayan, A. Brintzinger, W. Guthrie, D. Lachtrupp, J. Mauger, D. Glimmer, S. Lawn, B. Dinkel, and A. Mitwalsky, "Reliability of laser activated metal fuses in drams," in *Proc. of IEEE on Electronics Manufacturing Technology Symposium*, pp. 389–394, 1999.

[74] N. Robson, J. Safran, C. Kothandaraman, A. Cestero, X. Chen, R. Rajeevakumar, A. Leslie, D. Moy, T. Kirihata, and S. Iyer, "Electrically programmable fuse (efuse): From memory redundancy to autonomic chips," in *CICC*, pp. 799–804, 2007.

[75] M. Miller, J. Meraglia, and J. Hayward, "Traceability in the age of globalization: A proposal for a marking protocol to assure authenticity of electronic parts," in *SAE Aerospace Electronics and Avionics Systems Conference*, October 2012.

[76] U.S. Defense Logistics Agency, "Dna authentication marking on items in fsc 5962," August 2012. [Online]. Available: https://www.dibbs.bsm.dla.mil/notices/msgdspl.aspx?msgid=685

[77] Semiconductor Industry Association (SIA), "Public Comments - DNA Authentication Marking on Items in FSC5962," November 2012.

[78] C. Kuemin, L. Nowack, L. Bozano, N. D. Spencer, and H. Wolf, "Oriented assembly of gold nanorods on the single-particle level," *Advanced Functional Materials*, vol. 22, no. 4, pp. 702–708, 2012.

[79] IBM Research, "Nanorods take down counterfeiters: IBM scientists create nano-sized patterns to thwart forgeries," http://www.research.ibm.com/articles/nano-counterfeit.shtml.

[80] T. Rahman, D. Forte, J. Fahrny, and M. Tehranipoor, "Aro-puf: An aging-resistant ring oscillator puf design," in *Proceedings of the Conference on Design, Automation & Test in Europe (DATE)*, pp. 69:1–69:6, 2014.

[81] M. T. Rahman, D. Forte, F. Rahman, and M. Tehranipoor, "A pair selection algorithm for robust ro-puf against environmental variations and aging," in *IEEE International Conference on Computer Design (ICCD)*, pp. 415–418, 2015.

[82] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant ro-puf for reliable key generation," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[83] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pp. 298–303, 2010.

[84] CHASE, "CHASE Workshop on Secure/Trustworthy Systems and Supply Chain Assurance," April 2014, https://www.chase.uconn.edu/chase-workshop-2014.php.

[85] X. Zhang, K. Xiao, and M. Tehranipoor, "Path-delay fingerprinting for identification of recovered ics," in *Proc. International Symposium on Fault and Defect Tolerance in VLSI Systems*, October 2012.

[86] K. Huang, J. Carulli, and Y. Makris, "Parametric counterfeit IC detection via Support Vector Machines," in *Proc. International Symposium on Fault and Defect Tolerance in VLSI Systems*, pp. 7–12, 2012.

[87] Y. Zheng, A. Basak, and S. Bhunia, "Caci: Dynamic current analysis towards robust recycled chip identification," in *Design Automation Conference (DAC)*, pp. 1–6, June 2014.

[88] C. Hu, S. C.Tam, F.-C. Hsu, P.-K. Ko, T.-Y. Chan, and K. Terrill, "Hot-Electron-Induced MOSFET Degradation - Model, Monitor, and Improvement," *JSSC*, vol. 20, no. 1, pp. 295–305, 1985.

[89] B. Tudor, J. Wang, Z. Chen, R. Tan, W. Liu, and F. Lee, "An accurate MOSFET aging model for 28 nm integrated circuit simulation," *Microelectronics Reliability*, vol. 52, no. 8, pp. 1565 – 1570, 2012.

[90] J. Chen, S. Wang, and M. Tehranipoor, "Efficient Selection and Analysis of Critical-reliability Paths and Gates," in *GLSVLSI*, pp. 45–50, 2012.

[91] Synopsys, "90nm Generic Library."

[92] A. T. Appel, "Rectangular contact used as a low voltage fuse element," Patent US6 774 457 B2, Aug, 2004.

[93] H. C. Nicolay, "Integrated circuit fuse," Patent US 4 272 753, Jun 9, 1981.

[94] B. Kaminska, K. Arabi, I. Bell, P. Goteti, J. Huertas, B. Kim, A. Rueda, and M. Soma, "Analog and mixed-signal benchmark circuits-first release," in *ITC*, pp. 183–190, 1997.

[95] M. K. Steven, "Fundamentals of statistical signal processing," *PTR Prentice-Hall, Englewood Cliffs, NJ*, 1993.

[96] "Predictive Technology Model (PTM)," http://ptm.asu.edu/.

[97] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," in *VLSI Test Symposium, 2007. 25th IEEE*, pp. 277–286, May 2007.

[98] T. Sato, T. Kozaki, T. Uezono, H. Tsutsui, and H. Ochi, "A device array for efficient bias-temperature instability measurements," in *Solid-State Device Research Conference (ESSDERC), 2011 Proceedings of the European*, pp. 143–146, Sept 2011.

[99] "normfit," http://www.mathworks.com/help/stats/normfit.html.

[100] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert, "Breaking and entering through the silicon," in *Proceedings of SIGSAC Conference on Computer and Communications Security*, pp. 733–744, 2013.

[101] A. Yeh, "Trends in the global IC design service market," DIGITIMES Research, March 2012.

[102] E. Castillo, U. Meyer-Baese, A. García, L. Parrilla, and A. Lloris, "IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 5, pp. 578–591, May 2007. [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2007.896914

[103] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Constraint-based watermarking techniques for design ip protection," *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 20, no. 10, pp. 1236–1252, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/43.952740

[104] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. Springer, 2012.

[105] X. Zhuang, T. Zhang, H.-H. S. Lee, and S. Pande, "Hardware assisted control flow obfuscation for embedded processors," in *Proceedings of the 2004 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, ser. CASES '04, pp. 292–302. New York, NY, USA: ACM, 2004. [Online]. Available: http://doi.acm.org/10.1145/1023833.1023873

[106] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. of ACM/IEEE on Design Automation Conference*, pp. 83–89, June 2012.

[107] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*, pp. 137–143, May 2015.

[108] Synopsys, "Compression for Highest Test Quality and Lowest Test Cost," 2015. [Online]. Available: https://www.synopsys.com/Tools/Implementation/RTLSynthesis/Test/Pages/dftmax-ultra-ds.aspx

[109] Synopsys, "High Quality, Low Cost Test," 2015. [Online]. Available: https://www.synopsys.com/Tools/Implementation/RTLSynthesis/Test/Pages/DFTMAX.aspx

[110] P. Nagaraj, "Choosing the Right Scan Compression Architecture for Your Design," Tech. Rep., 2015.

[111] DASC, "1735-2014 - IEEE Approved Draft Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)," 2014.

[112] Synopsys, "Synopsys FPGA Synthesis Synplify Pro for Lattice: User Guide," November 2014.

[113] Microsemi, "Libero SoC Secure IP Flow User Guide for IP Vendors and Libero SoC Users," 2014.

[114] NIST, "FIPS PUB 180-4: Secure Hash Standard," March 2012.

[115] E. Charbon, "Hierarchical watermarking in IC design," in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, pp. 295–298, May 1998.

[116] A. Kahng, J. Lach, W. Mangione-Smith, S. Mantik, I. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Constraint-based watermarking techniques for design IP protection," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 10, pp. 1236–1252, Oct 2001.

[117] G. Qu and M. Potkonjak, *Intellectual property protection in VLSI designs: theory and practice*. Springer Science & Business Media, 2003.

[118] J. Lach, W. Mangione-Smith, and M. Potkonjak, "Fingerprinting techniques for field-programmable gate array intellectual property protection," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 10, pp. 1253–1261, Oct 2001.

[119] D. Kirovski, Y.-Y. Hwang, M. Potkonjak, and J. Cong, "Protecting Combinational Logic Synthesis Solutions," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 12, pp. 2687–2696, Dec 2006.

[120] F. Koushanfar, "Provably secure active ic metering techniques for piracy avoidance and digital rights management," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 1, pp. 51–63, Feb 2012.

[121] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[122] J. Kurose and K. Ross, "Computer networks: A top-down approach," 2001.

[123] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer, November 2000.

[124] IEEE Standards Association and others, *1149.1-2001 - IEEE Standard Test Access Port and Boundary Scan Architecture*. IEEE, 2001.

[125] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[126] D. S. Jeong, R. Thomas, R. Katiyar, J. Scott, H. Kohlstedt, A. Petraru, and C. S. Hwang, "Emerging memories: resistive switching mechanisms and current status," *Reports on Progress in Physics*, vol. 75, no. 7, p. 076502, 2012.

[127] Morris Dworkin, "NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation ," 2001.

[128] ITC99 Benchmark Home Page, https://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[129] OpenCores, https://www.opencores.org.

[130] OpenSPARC T1, http://www.oracle.com/technetwork/systems/opensparc/ opensparc-t1-page-1444609.html.

[131] Synopsys 32/28nm Generic Library for Teaching IC Design, https://www.synopsys.com/COMMUNITY/UNIVERSITYPROGRAM/ Pages/32-28nm-generic-library.aspx.

[132] A. Miyamoto, N. Homma, T. Aoki, and A. Satoh, "Systematic Design of RSA Processors Based on High-Radix Montgomery Multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 7, pp. 1136–1146, July 2011.

[133] D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial sram state as a fingerprint and source of true random numbers for rfid tags," in *In Proceedings of the Conference on RFID Security*, 2007.

[134] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *Computers, IEEE Transactions on*, vol. 56, no. 1, pp. 109–119, Jan 2007.

[135] Xilinx, 2105, http://www.xilinx.com/publications/prod_mktg/zynq7000/Zynq-7000-combined-product-table.pdf.

[136] Xilinx, 2105, http://www.xilinx.com/publications/prod_mktg/Virtex7-Product-Table.pdf.

[137] ERAI, "Report to ERAI," http://www.erai.com/information_sharing_high_risk_parts.

[138] GIDEP, "How To Submit Data," http://www.gidep.org/data/submit.htm.

Appendix A

Publications Related to this Thesis

Book

1. M. M. Tehranipoor, U. Guin, and D. Forte, Counterfeit Integrated Circuits: Detection and Avoidance, Springer, 2015.

Journal Papers

1. U. Guin, Q. Shi, D. Forte, and M. Tehranipoor, "FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs", ACM Transactions on Design Automation of Electronic Systems (TODAES), 2016.

2. U. Guin, D. Forte, and M. Tehranipoor, "Design of Accurate Low-Cost On-Chip Structures for protecting Integrated Circuits against Recycling", IEEE Transactions on VLSI Systems (TVLSI), 2015.

3. U. Guin, K. Huang, D. DiMase, J. M. Carulli Jr., M. Tehranipoor, and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," Proceedings of the IEEE, 2014.

4. U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," Journal of Electronic Testing: Theory and Applications (JETTA), 2014.

5. U. Guin, D. DiMase, and M. Tehranipoor, "A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment," Journal of Electronic Testing: Theory and Applications (JETTA), 2014.

Conference Papers

1. B. Shakya, U. Guin, M. Tehranipoor and D. Forte, "Performance Optimization for On-Chip Sensors to Detect Recycled ICs" to appear in IEEE International Conference on Computer Design (ICCD), 2015.

2. U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-Cost On-Chip Structures for Combating Die and IC Recycling," Design Automation Conference (DAC), 2014.

3. U. Guin, D. Forte, and M. Tehranipoor, "Anti-Counterfeit Techniques: From Design to Resign," IEEE Microprocessor Test and Verification (MTV), 2013.

4. U. Guin and M. Tehranipoor, "On Selection of Counterfeit IC Detection Methods," IEEE North Atlantic Test Workshop (NATW), 2013.

5. U. Guin, T. Chakraborty, and M. Tehranipoor, "Functional F_{max} Test-Time Reduction using Novel DFTs for Circuit Initialization," IEEE Int. Conference on Computer Design (ICCD), 2013.

6. U. Guin, T. Chakraborty, and M. Tehranipoor, "Novel DFTs for Circuit Initialization to Reduce Functional Fmax Test Time," IEEE North Atlantic Test Workshop (NATW), 2013.

Technical Reports

1. U. Guin, M. Tehranipoor, D. DiMase, and M. Megrdician, "Counterfeit IC Detection and Challenges Ahead," ACM SIGDA, March 2013.