Testing Techniques for Detection of Hardware Trojans in Integrated Circuits of Trusted Systems

Leonel Acunha Guimarães

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Nano Électronique & Nano Technologies**

Arrêté ministériel : 25 mai 2016

Présentée par

Leonel ACUNHA GUIMARÃES

Thèse dirigée par **Laurent FESQUET**
et co-encadreée par **Rodrigo POSSAMAI BASTOS**

réparée au sein du **Laboratoire TIMA**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique & Traitement du Signal (E.E.A.T.S)**

Testing Techniques for Detection of Hardware Trojans in Integrated Circuits of Trusted Systems

Thèse soutenue publiquement le **1 décembre 2017**,
devant le jury composé de :

M. Guy GOGNIAT
Professeur, Université de Bretagne du Sud, Président
M. Giorgio DI NATALE
Directeur de Recherche, LIRMM-CNRS, Rapporteur
M. Lilian BOSSUET
Professeur, Université Jean-Monnet de Saint-Etienne, Rapporteur
M. Laurent FESQUET
Maître de Conférences, Université Grenoble Alpes, Directeur de thèse
M. Rodrigo POSSAMAI BASTOS
Maître de Conférences, Université Grenoble Alpes, Co-Encadrant de thèse

To Nadir, Cléia, and Joana

Acknowledgement

I would like to dedicate a few words to acknowledge some people who were essential for the accomplishment of this work.

Firstly, I would like to express my sincere gratitude to my advisors Dr. Laurent Fesquet and Dr. Rodrigo Possamai Bastos who gave me the chance to work with them and guided me through this harsh road. Their guidance surely helped me to improve my research and mitigate all the insecurities a PhD student may have. I wish to express my special thanks to Prof. Guy Gogniat, Dr. Giorgio Di Natale, and Prof. Lilian Bossuet who accepted being part of my examining board. Their valuable and constructive feedback certainly counted for improving the quality of this manuscript and the dissertation defense.

I would like to thank all the staff of TIMA Laboratory, EEATS, and CIME Nanotech who always helped me somehow with uncountable issues, specially Alejandro Chagoya. I would like to name a few researchers, PhD and former PhD students who were great fellows whose collaboration impacted decisively in the result of this work: Thiago, Otto, Rodrigo Iga, Jean, Tugdual, Chadi, Amani, Raphael, Matheus, Ricardo, Assia, Karim, Sophie, Grégoire, Sylvain, Arthur, Alexandre, and Ali. Their feedbacks, friendship, and numerous rubber duck debbuging were essential for the final result of this work. I would like to also address special thanks to all my friends I had the pleasure to meet in Grenoble.

I wish to thank my loved ones who I had to stay apart since I decided to move to France: my aunts, uncles, cousins, grandmother, godfather, and all friends in Brazil.

I would like to extend great thanks to my dear Vó Cléia, Vó Joana, and Dinda Nadir who I dedicate this thesis to. I am grateful for your teaching, love, kindness, and affection. I would change everything to pass more time with you.

I would especially like to thank my parents Leonel and Ivana for everything they have done for me, they are the best parents I could have. This work would be nothing without them.

Finally, I wish to thank my beloved girlfriend Natália for her love, friendship, support, and inspiration.

Table of Contents

List of Figures

List of Tables

Abstract

The world globalization has led the semiconductor industry to outsource design and fabrication phases, making integrated circuits (ICs) potentially more vulnerable to malicious modifications at design or fabrication time: the hardware Trojans (HTs). New efficient testing techniques are thus required to disclose potential slight and stealth HTs, and to ensure trusted devices. This thesis studies possible threats and proposes two new post-silicon testing techniques able to detect HTs implanted after the generation of the IC netlist. The first proposed technique exploits bulk built-in current sensors (BBICS) – which are originally designed to identify transient faults in ICs – by using them as testing mechanisms that provide statistically-comparable digital signatures of the devices under test. With only 16 IC samples, the testing technique can detect dopant-level Trojans of zero-area overhead.The second proposition is a non-intrusive technique for detection of gate-level HTs in asynchronous circuits. With this technique, neither additional hardware nor alterations on the original test set-up are required to detect Trojans smaller than 1% of the original circuit. The studies and techniques devised in this thesis contribute to reduce the IC vulnerability to HT, reusing testing mechanisms and keeping security features of original devices.

Keywords: Hardware Trojans, design for test & security, side-channel analysis, transient faults

Résumé

La mondialisation et la déverticalisation des métiers du semi-conducteur a mené cette industrie à sous-traiter certaines étapes de conception et souvent la totalité de la fabrication. Au cours de ces étapes, les circuits intégrés (CIs) sont vulnérables à des altérations malignes : les chevaux de Troie matériels (HTs). Dans les applications sécuritaires, il est important de garantir que les circuits intégrés utilisés ne soient pas altérés par de tels dispositifs. Afin d'offrir un niveau de confiance élevé dans ces circuits, il est nécessaire de développer de nouvelles techniques de test pour détecter les HTs, aussi légers et furtifs soient-ils. Cette thèse étudie les menaces et propose deux approches originales de test post-fabrication pour détecter des HTs implantés après synthèse. La première technique exploite des capteurs de courant incorporés au substrat (BBICS), originalement conçus pour identifier les défauts transitoires dans les CIs. Dans notre cas, ils fournissent une signature numérique obtenue par analyse statistique permettant de détecter tout éventuel HT, même au niveau dopant. La deuxième proposition est une méthode non intrusive pour détecter les HTs dans les circuits asynchrones. Cette technique utilise la plateforme de test du circuit et ne requiert aucun matériel supplémentaire. Elle permet la détection de HTs dont la surface est inférieure à 1% de celle du circuit. Les méthodes et les techniques mises au point dans cette thèse contribuent donc à réduire la vulnérabilité des CIs aux HTs soit par adjonction d'un capteur (BBICS), soit en exploitant les mécanismes de test s'il s'agit de circuits asynchrones.

Mots-clés : Chevaux de Troie matériels, conception pour le test & la sécurité, analyse par canaux-cachés, défauts transitoires

Chapter 1

Introduction

In the increasing process of globalization, IC companies rely on outsourcing the different steps of their projects to minimize costs and time-to-market. This trend has been largely adopted since the mid-1980s when fabless models became viable thanks to the development of dedicated semiconductor foundries and commercial CAD tools that enabled the design and fabrication of more complex ICs.

This ecosystem have made ICs the result of a production chain carried by multiple companies often based in different continents. Moreover, the usage of third-party components, tools, and manufacturing process hamper the fully certification of the whole environment, rendering the devices vulnerable to malicious inclusions. Hardware Trojans (HT) can therefore be inserted in the systems to change their functionalities, leak device's secret data or make it able to run obscure functions. For instance, in the IC manufacturing process, a foundry and its personnel have access to physical layout design files (e.g. GDSII, OASIS) in which all informations about the IC original design can be extracted, making them able to recreate and modify it.

The HT concern was formally discussed for the first time in 2005 within an U.S. Defense Science Board study [106] in which the Department of Defense of the U.S. encouraged research efforts towards the exploration and improvement of the trustworthiness of ICs used in military systems. It has led the Defense Advanced Research Projects Agency (DARPA), the Pentagon's R&D wing, to initiate the TRUST in IC program [9] focusing on the development of Trojan detection methods. Thereafter, researchers and engineers have proposed attack models and detection techniques able to deal with ICs vulnerabilities [130].

Among the harmful HT-induced effects, IC-based systems can be led to shut down by simple induction of delay errors in internal blocks or machines of computer networks can become unavailable, interrupt or suspend services. Other resourceful Trojans can add backdoors in security-oriented circuits to make easier side-channel attacks, and, thus, leak secret keys of cryptographic systems [72]. Trojans may also be designed to change functionalities and degrade system performance [120]. Furthermore, they can be unobtrusive and dormant, acting only within certain rare circumstances, which, in some occasions, can only be reached by attackers,

rendering Trojans imperceptible over regular functional tests, and consequently very hostile after their activation. Prominent taxonomies [24, 118, 130, 132, 137] classify HT insertion at different IC-abstraction levels, from specification to assembly and package phases: system, development environment, register transfer (RT), gate, transistor, and physical (layout).

According to the degree of trust of each IC design flow phase, different countermeasures are applicable. The more trusted phases are ensured, the more information is available to eventually detect HTs in the design. Ultimately, if all phases from IC design flow are certified until the generation of a trustworthy layout, a possible Trojan inserted afterwards could be detected by reverse engineering and physically inspecting the manufactured device under Trojan test (DUTT) in order to recover its layout shape and comparing it with the previously designed one. In the case where designers have at least access to trusted system hardware description codes, primary outputs and side-channel signals such as power consumption are exploitable parameters to be used as references (golden data) in a further comparative analysis between this golden data and the signals produced by the manufactured DUTT. However, if a company cannot ensure not even code reliability, less trusted data are available, thus, making HT detection conditioned to designer assumptions about possible HT models that could be implemented in the system.

The implementation of the methods for Trojan detection inherently imposes a trade-off between the desired detection effectiveness and the price designers can afford. The techniques may require extra on-chip circuitry, considerable modifications on post-silicon test set-up or both to allow the obtainment of the internal signals necessary to perform the detection. Therefore, the Trojan detection cost is a combination of the required area, and test duration and set-up charge overhead. The search for alternatives to mitigate this cost is the main challenge in HT detection techniques.

To address these issues, this thesis presents two new testing techniques designed to detect Trojans by adapting and reusing circuit inherent structures for this purpose. The first one is reusing the bulk built-in current sensor (BBICS) – which was originally designed to detect transient faults in ICs [92, 125]– as a mechanism able to track physical modifications caused by a Trojan in sub-regions of the IC. In addition, the main BBICS architectures are reported and a new very low area design is proposed. The other detection technique presented in this thesis draw on clockless circuit –asynchronous circuit– intrinsic properties to perform the HT detection neither requiring extra circuitry nor dedicated test set-up. A test procedure based only on the global supply current measure is needed to detect Trojans in such architectures, which are already security solutions robust against different side-channel attacks [57, 95, 104, 127]. Thus, these two new techniques are able to explore intrinsic properties of the pre-designed structures that allows the HT detection combined with their security features, making both designs able operate in two different modes: post-silicon test-time to detect Trojans and run-time to prevent possible attacks or faults.

In addition, we also propose new attacks based on different types of transistor-level Trojans

implemented in ring oscillators (ROs). These HTs are presented to explore and expose the vulnerabilities of security systems based on ROs and the capability of a few transistors to reduce the circuit reliability.

This thesis is organized as follows. Chapter 2 introduces the state of the art of existing Trojan taxonomy, presenting threats and methods to detect Trojans recently proposed in the literature. Very simple transistor-level Trojans are presented in chapter 3 in order to show that even small transistor-level Trojans inserted in ROs are able to cause different effects in the systems. In chapter 4, the main BBICS architectures are analyzed including a new efficient dynamic BBICS proposition with reduced area overhead. In chapter 5, a testing technique is proposed to use the same BBICS to track transient faults and monitor the original circuit substrate against ultra-small Trojans down to physical-level. A testing method using properties of asynchronous circuits is proposed in chapter 6 to detect gate-level Trojans. Finally, chapter 7 concludes this thesis, summarizing the main contributions and presenting the conclusions and future perspectives of this work.

Chapter 2

Hardware Trojan Insertion and Detection

Thanks to IPs, netlists, libraries, standard cells, tools, and scripts supplied by third-party vendors, semiconductor companies nowadays rapidly design, simulate, and verify complex circuits in modern technologies. Besides that, outsourcing IC design phases is an opportunity to reduce investments and time-to-market. The best-know example is the fabless model [99] in which companies outsource the manufacturing process of their devices, making ICs without the set up of their own foundries that could demand up to 20 billion dollars for modern technologies [44]. Moreover, as technologies change approximately every 2 years, initiating a fully in-house company is a very risky business. Accordingly, companies can use third-parties know-how to design auxiliary application-oriented IPs to avoid spending time with its development.

Hence, assuming that all phases of the IC production chain can be outsourced, untrusted suppliers or their personnel can maliciously alter the original projects – during any phase – for suspicious reasons [9]. Such modifications are referred to as hardware Trojans (HT). In this chapter, HTs are fully described including their motivations and implementations in a detailed taxonomy as well as case studies and approaches proposed in literature to perform their detection. Section 2.1 introduces aspects related to Trojan insertion, while section 2.2 presents the different Trojan detection categories and their applications.

2.1 Trojan Insertion

Any IC design phase is vulnerable to HT insertions whether the third-party suppliers of each phase are not fully certified. Figure 2.1, based on [24, 48, 61, 132], depicts a modern IC design flow adopted by most of the semiconductor companies and the level of trust in each step that might be a backdoor to an adversary insert a Trojan.

The level of trust in each step depends on the adversary opportunities and the difficulty of

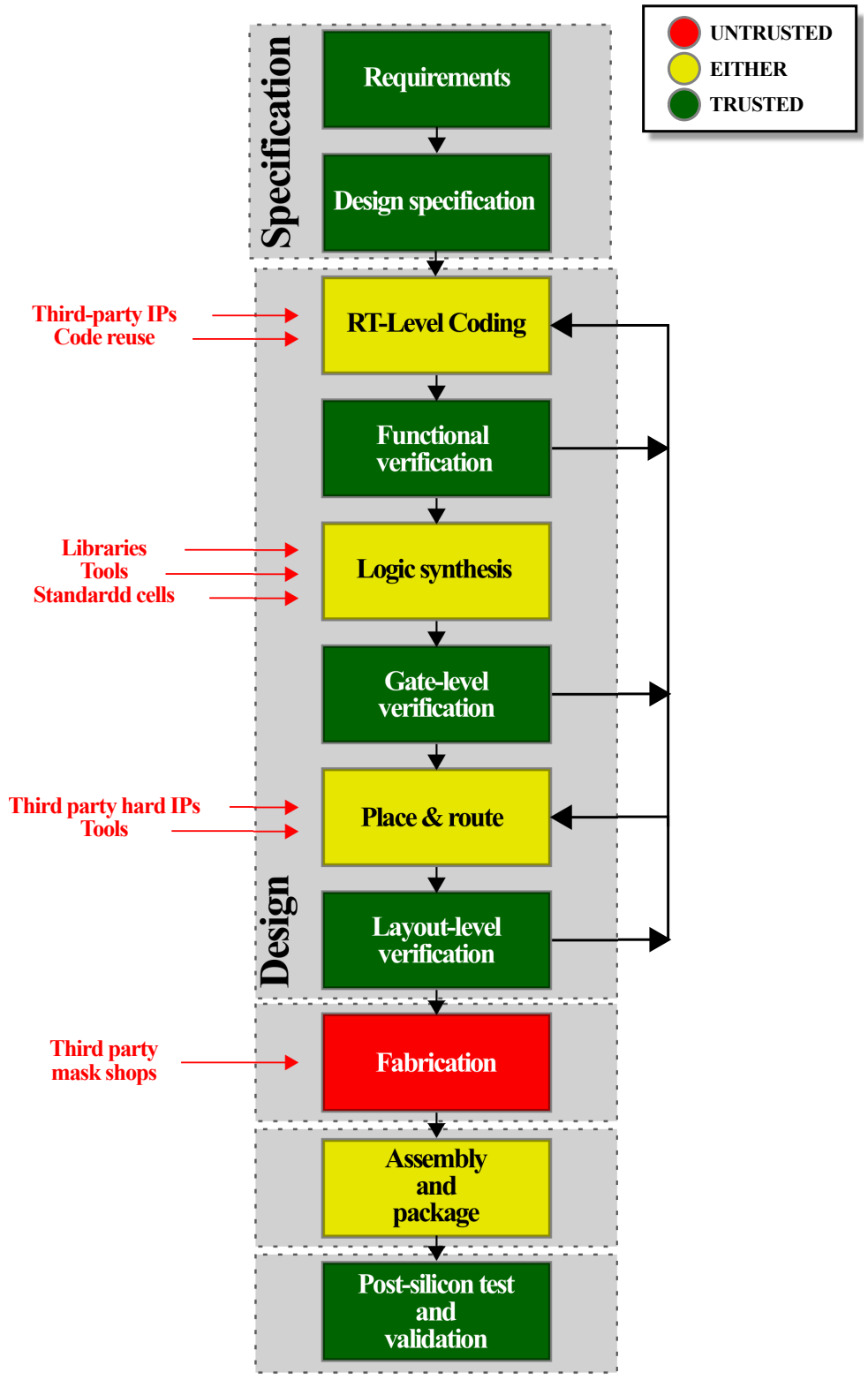


Fig. 2.1: Vulnerable steps of a modern IC production chain and their levels of trust taking into account: the adversary's information about the design, the difficulty of tampering with the system, and the facility of detecting modifications [24, 48, 61, 132].

identifying malicious adjustments or alterations. The whole specification phase is considered trusted since suspicious constraints are in the form of text and thus are easily debugged without any advanced test mechanism by the companies targeting to design a trusted IC. On the other hand, during the design phase, hardware description files (e.g. VHDL, VERILOG) may rely on the use of untrusted third-party IPs and codes, considerably compromising the design trustworthiness. The level of trust of logic synthesis, and place and route steps depends if the related libraries, tools, standard cells, and third-party hard IPs are properly certified. That is the reason that these phases are classified as either: trusted or untrusted. Even if adversaries have a limited access to the genuine design, only if all tools, libraries, and components were fully certified these steps would be considered trusted. In contrast, verification steps – typically performed in-house – are trusted if using testbenches and tools certified by qualified organizations. The same, however, is not applicable to the fabrication phase, normally executed within an outhouse foundry, and thus considered the most sensitive phase of IC production chains. In fact, foundries, third-party mask shops, or their personnel have access to all genuine stream files (e.g. GDSII, OASIS), which might be used to predict system applications and tamper with genuine designs. Furthermore, even the phase of assembly and package might be untrusted assuming that adversaries may modify authentic hardware components during the chip integration or replace them by malicious ones. However, unlike fabrication phase, obtaining the system design information during the assembly and package phase is a very challenging task, which considerably limits the possibility of attacks at this phase in relation to the fabrication one. Finally, the post-silicon test and validation phase could be trusted only if they are performed in-house or by a fully certified company. In the case where they are outsourced to an untrusted company, the test reports generated could be altered to mask possible effects caused by Trojans. Nevertheless, as this phase represents the last chance to detect Trojans before delivering the IC for deployment or costumers, it must be trusted.

2.1.1 Trojan Model

A Trojan model was defined in literature [24, 132] in which the HT is decomposed into two parts: the payload and the trigger. The payload is the part of the circuit that indeed causes harmful effects in the target circuit. The trigger is a monitoring circuit that serves as a mechanism to keep the payload inactive and thus hide the their effects until a rare condition is accomplished to activate it. Adversaries may use this clever strategy to make their HTs stealthy during the verification and validation steps. If the trigger is an always-on circuit, it is supposed to be less perceptible than the payload in order to reduce HT impacts in the original circuit and thus prevent its detection. On the other hand, as the payload remains most of the time inactivated, it can be a more sophisticated circuit. Thereby the HT model is basically a dormant circuit that, once triggered, modifies the targeted system original behavior. Fig. 2.2 depicts this HT model.

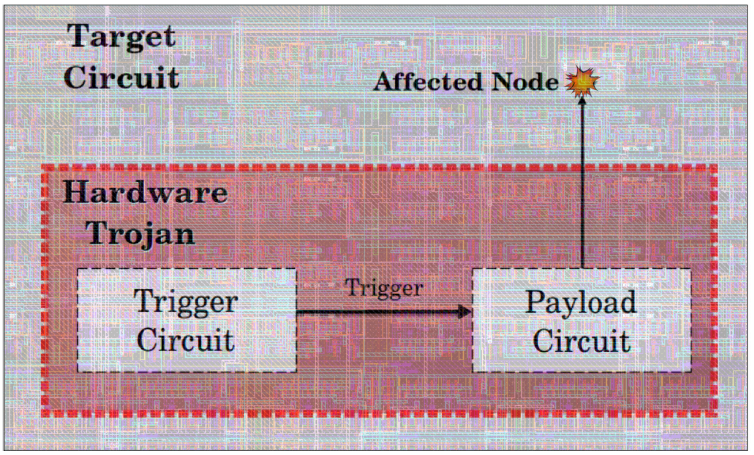


Fig. 2.2: Architecture of a Trojan inserted on a target circuit.

2.1.2 Trojan Taxonomy

In order to evaluate the risks of HTs, several studies have been reported taxonomies [24, 63, 79, 118, 120, 130, 131, 132, 132, 137] abstracting different categories related to the architecture, effects, and insertion of Trojans in ICs. Fig. 2.3 summarizes the existing Trojan insertion taxonomies. A discussion about the different categories provided in the taxonomy of Fig. 2.3 is presented throughout this section.

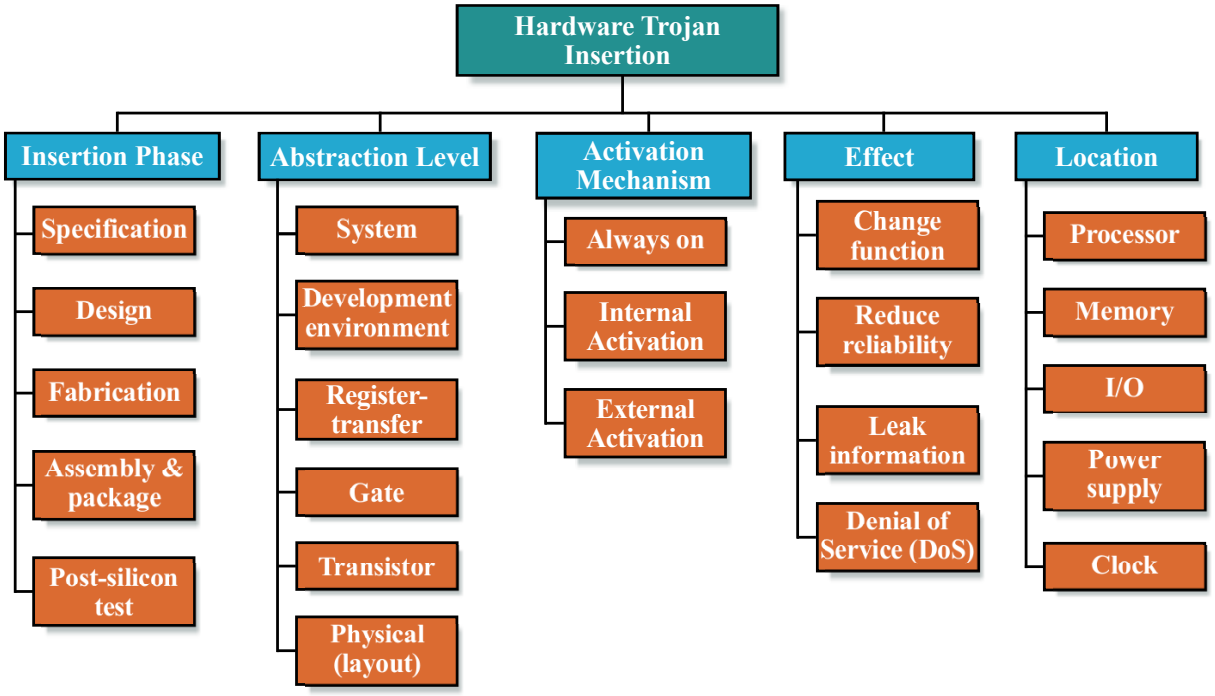


Fig. 2.3: Trojan insertion taxonomy [24, 63, 79, 118, 120, 130, 131, 132, 137].

2.1.2.1 Insertion Phase

This category represents different phases of the IC design flow in which a possible adversary can be located. The following analysis presents the vulnerabilities to Trojan insertion in each phase.

- **Specification:** An adversary could intentionally define weak requirements for the system. As a possible result, design reliability may become compromised making the device vulnerable to leak sensitive information.
- **Design:** Even if the whole design is done in-house, the simple usage of untrusted tools, libraries, third-party IPs and standard cells may affect the it in a harmful way. For instance, untrusted tools may add extra circuitry in the system to introduce backdoors in the genuine design. If any step of the design phase is outsourced, a Trojan could be directly added to the hardware description files of the genuine circuit.
- **Fabrication:** An untrusted foundry, mask shop or their personal are able to retrieve the genuine circuit components and thus predict its behavior and probable applications. Therefore, the design becomes susceptible to addition or removal of components. Modifying physical circuits characteristics (sizes and channel doping concentration level) [72] can also fairly increase the circuit vulnerability to fault based attack.
- **Assembly and package:** The IC is encapsulated in a supporting case and the packaged chip is assembled in a PCB with other hardware. An adversary may add malicious hardware components surrounding the genuine design to provoke malfunctions or increase leakages.
- **Post-silicon test:** At testing phase, an adversary is no longer able to modify the genuine circuit structure, however the test set-up, programs or reports may be changed in order to mask possible Trojan effects. Besides that, as it is the last step of the IC design flow, it is the last opportunity to original designers to detect Trojans before the deployment phase.

2.1.2.2 Abstraction Level

The abstraction level refers to possible tampers with the design if an adversary has access to sensitive files at different abstraction levels. The following analysis presents HT insertion opportunities in each abstraction level.

- **System:** A HT can simply be alterations in function specifications, protocols, interfaces and constraints of the genuine design. An adversary involved at the system level may add some obscure specifications to give him the control of secret data flowing through the manufactured device. For instance, an adversary at the specification phase could

change specifications of true random number generators (TRNG) to make it work in a predictable way due to some condition that only the owner of the HT is aware of. This is able to considerably reduce the reliability of secure systems based on these architectures and provide secret information to attackers.

- **Development environment:** Untrusted tools and scripts may present hidden functions, leading designers to generate circuits infected by Trojans. In addition, untrusted simulation tools and testbenches could mask HT effects. Any unreliable third-party vendor is able to insert Trojans at this level.

- **Register-transfer:** A HT can also be a simple modifications in genuine RT-level codes or constraint files. An adversary can modify circuit functions in order to provoke significant consequences such as failures in cryptographic blocks. Attackers at design phase or an untrusted code supplier are possible sources for the HT insertion at this level.

- **Gate:** The addition or removal of one or more gates in the original netlist is considered a gate-level HT. Standard delay format (sdf) files, that contains system timing data can also be modified, changing timing check, constraints, and delays to hide HT effects. Adversaries at the gate design phase and third-party vendors have access to implement Trojans at this level.

- **Transistor:** Transistor addition can significantly increase leakages, opening backdoors for attackers to get knowledge about security-oriented circuit internal states. Moreover, transistors may be added to increase critical path delays, leading the circuit to malfunction. Adversaries at design phase or untrusted tools, libraries and models are possible sources of Trojans at this level.

- **Physical (layout):** Original parameters of circuit components are vulnerable even after the layout generation. For instance, an attacker could alter original masks, changing transistor lengths, widths or channel doping concentrations. Moreover, wires can be resized, generating malfunctions and extra leakages. Adversaries at design and fabrication level, or third-party mask shops have access to modify the original layout and insert such Trojans.

2.1.2.3 Activation Mechanism

If a Trojan is always activated, its effects in the circuit may upset some device property, making it exposed to verification and validation routines. However, if a HT remains dormant until the deployment phase, its disturbances in the circuit behavior become less noticeable, considerably hampering its detection. For this purpose, Trojans are likely to feature activation mechanisms used to activate them by a certain condition accomplished only after verification and validation

phases. Therefore, HTs are considered dormant during test routines and hostile after being activated.

- **Always on:** The target circuit behavior is always affected by the HT. The Trojan is therefore composed only by the payload.

- **Internal activation:** A Trojan is activated when a specific internal condition occurs in the circuit. For instance, an internal counter may trigger the HT if the clock exceeds a certain value. Besides that, internal signal patterns or rare conditions may trigger this type of Trojan.

- **External activation:** Trojans are activated by an attacker aware of the HT presence in the circuit. For instance, a Trojan may be designed to be activated whenever a certain value is set in the circuit inputs. Thus, attackers with the knowledge of such an activation mechanism are able to activate the HT. Sophisticated trigger mechanisms rely on very rare sequences, conditions or even side-channel attacks to be activated, making its detection almost impossible by users which are not aware of its activation mechanism.

2.1.2.4 Effect

A HT may lead the device to different effects by depending on the adversary possibilities and intentions. In the following analysis, a classification of the HT effects is presented in the sequel.

- **Change function:** Trojans change, add or remove original circuit functions. For instance, Trojans could lead to improper calculations under specific conditions, compromising the main system operations. For example, missile guidance systems may intentionally calculate a wrong target, leading the missile to strike a completely unintended destination.

- **Reduce reliability:** An adversary can implement Trojans in a system for downgrading its performance or rendering it more vulnerable to side-channel attacks. In other system applications, Trojans may increase the power consumption, causing a faster battery discharge to interrupt the circuit operation.

- **Leak information:** Trojans are designed to leak keys and plaintexts of cryptographic circuits by primary outputs or side-channel signals. An adversary could add a comparator Trojan that enables the key leakage whenever a certain input or sequence of outputs is set.

- **Denial of service (DoS):** Trojans can make the circuit no longer able to work properly. For instance, a Trojan inserted in a military radar may lead it to not detect some specific threats.

2.1.2.5 Location

Trojans are also classified regarding to their location in the design. According to that, different kinds of attacks are possible.

- **Processor:** Trojan may add and remove instructions of processors, leading it to operate suspicious functions and cause malfunctions.
- **Memory:** Attackers with the control of memory elements may be able to get access to secret informations and clear sensitive data stored in the device.
- **I/O:** Pins be controlled by the HT may lead the circuit to avoid some specific conditions, output wrong signals and monitor communications.
- **Power supply:** Trojans in the power grid may control the device voltages and current, increasing leakages or causing failures.
- **Clock:** Trojans may alter circuit frequency or increase clock noise causing glitches and jitters. This threat can make secure blocks to leak information and create backdoors to side-channel attacks.

2.1.3 Case Studies: Trojans at Different Abstraction Levels

In this section, a few attacks using Trojans are presented in order to illustrate possible threats. In this regard, each Trojan presented in this section is accordingly classified to the taxonomy.

2.1.3.1 Trojan in an AES-128 [118, 120]

Authors of [118, 120] have developed a set of benchmarks, the Trust-Hub (http://www.trust-hub.org/), sponsored by the USA National Science Foundation (NSF). It is a source of benchmarks infected by different types of Trojans at diverse abstraction levels in order to support the security community to compare the behavior of Trojan-free and Trojan-infected circuits to evaluate HT detection methods. A set of benchmarks are available, considering different categories in the Trojan taxonomy. In [120], a report about the available benchmarks is presented to encourage its application.

The *AES-T2000* benchmark consists of a Trojan surrounding an AES-128 encrypting block with the purpose of leaking its secret key. This HT is triggered after detecting a sequence of 4 input plaintexts. The payload is responsible for generating a significant current leakage indicating the actual state of each bit of the secret key. An attacker with access to the manufactured device can therefore exploit its current leakage to have access to the key. The code in the Fig. 2.4 illustrates this Trojan implementation. The library aes_128 is the original AES circuit, Trojan_Trigger and Trojan_Payload are respectively the trigger and the payload circuit.

top.v – AES-128 with Trojan Trigger and Payload

```
1  module top(clk, rst, state, key, out);
2  input          clk, rst;
3  input  [127:0] state, key;
4  output [127:0] out;
5
6  aes_128 AES (clk, state, key, out);
7  Trojan_Trigger Trigger(clk, rst, state, Tj_Trig);
8  Trojan_Payload Payload (clk, rst, key, Tj_Trig);
9
10 endmodule
```

Fig. 2.4: Architecture of the Trojan-infected AES-T2000 benchmark [120].

The Trojan classification according to the Trojan taxonomy is:

- Insertion phase: design
- Abstraction level: RT
- Activation mechanism: activated by a sequence of primary inputs
- Effects: leak information
- Location: processor

2.1.3.2 Trojan in the Configuration Bitsream of FPGAs [26]

The HT in [26] illustrates an attack caused by an untrusted FPGA synthesis tool. The HT alters directly the FPGA configuration bitstream. To this goal, the cyclic redundancy check (CRC) used to verify the bitstream correct implementation is disable to mask the HT effects. Therefore, ROs are added in the circuit to perform redundant switching activity causing power dissipation. Results in [26] show that the operating temperature is increased by over 160°C without affecting the functionality of the original design. The consequences of such an attack drastically speed up the aging effects in the FPGAs.

The Trojan classification according to the Trojan taxonomy is:

- Insertion phase: design
- Abstraction level: development environment
- Activation mechanism: always-on
- Effects: reduce reliability, DoS
- Location: any

2.1.3.3 Trojan in a Wishbone Interconnect Matrix [118, 120]

A gate-level implementation is illustrated by the benchmark *wb_conmax-T100* from Trust-hub [118, 120]. It is composed by a Wishbone bus infected by a gate-level Trojan able to stuck at '1' the four most significant bits of the address bus of the first master, which determines its slave. In this particular example, the trigger is larger than the payload in order to impose a very low activation probability about 9.78×10^{-68}. The gate-level scheme of this Trojan design is presented in the Fig. 2.5. Note that the original 4 most significant bits of the address bus are replaced by the Trojan affected Tj_address signals.

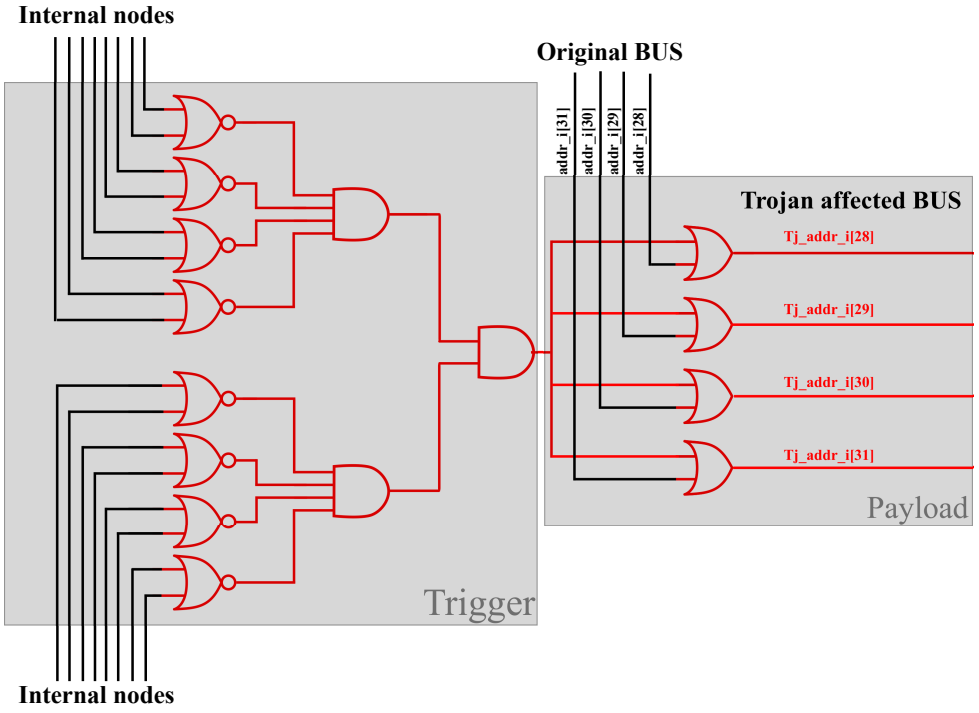


Fig. 2.5: Architecture of the gate-level Trojan implemented in the Wishbone bus in the benchmark *wb_conmax-T100* [118, 120].

The Trojan classification according to the Trojan taxonomy is:

- Insertion phase: design
- Abstraction level: gate
- Activation mechanism: internally conditionally triggered
- Effects: change Functionality, denial of Service
- Location: I/O

2.1.3.4 Parametric Trojans in Security Systems [17, 72]

Physical-level Trojans have been presented in literature as a dangerous backdoor for side-channel attacks. In [17], authors proposed an extremely stealthy approach for Trojan implementation, only by modifying the doping area, polarity or concentration of a few transistors in the original design of Intel's Random Number Generators (RNG). Using the same principle, authors in [72] perform key leakages in a PRINCE block cipher. Such Trojans alters the regular threshold voltage (V_{th}) of a few inverters from the original design, only making them vulnerable to V_{DD} glitches. As only the owner of the Trojan is aware of this modification, the HT activation is unlikely during regular tests. A possible type of layout modification is presented in Fig. 2.6 in order to illustrate the implementation of the dopant-level HT in which an adversary alters the channel doping concentration of a few gates of the genuine design.

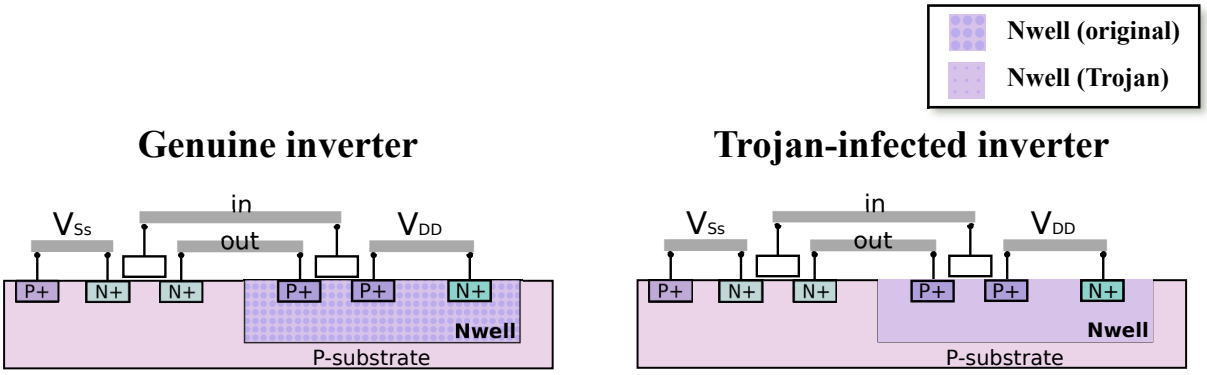


Fig. 2.6: Layout of an inverter with the dopant Trojan presented in [72].

The Trojan classification according to the Trojan taxonomy is:

- Insertion phase: fabrication
- Abstraction level: physical (layout)
- Activation mechanism: externally triggered (side-channel attack)
- Effects: leak cryptographic keys, reduction of reliability
- Location: processor

2.2 Trojan Detection

For ensuring the IC trustworthiness, different techniques can be implemented to detect or prevent Trojans according to the level of trust in each phase of the IC design. Several studies [48, 64, 69, 94, 130, 132] have reported comprehensible surveys about most detection methods presented in literature, classifying them in accordance with their approaches.

The techniques basically evaluate the deviations caused by HTs on the system behavior or look for possible HT profiles. To this aim, the designers must be aware of at least a single specific parameter from the genuine device or define a target HT model to be detected. If the deviation produced in the evaluated parameter of a design under Trojan test (DUTT) is greater than an acceptable margin, the DUTT is classified as Trojan infected. In Fig. 2.7, a scheme based on prior surveys and works presents the main categories of testing techniques for detection of Trojans.

Among the categories of Trojan detection listed in Fig. 2.7, the destructive techniques require reverse engineering of the genuine design in order to physically inspect the HT insertion. Otherwise, the non-destructive techniques are divided into: (1) post-silicon testing techniques that rely on detecting Trojans before the deployment of the device; and (2) run-time monitoring techniques that consist in on-line mechanisms able to detect and indicate – during the normal IC operation – malicious activities or malfunctions caused by Trojans. The content of Fig. 2.7 and the existing detection techniques are the theme of this section.

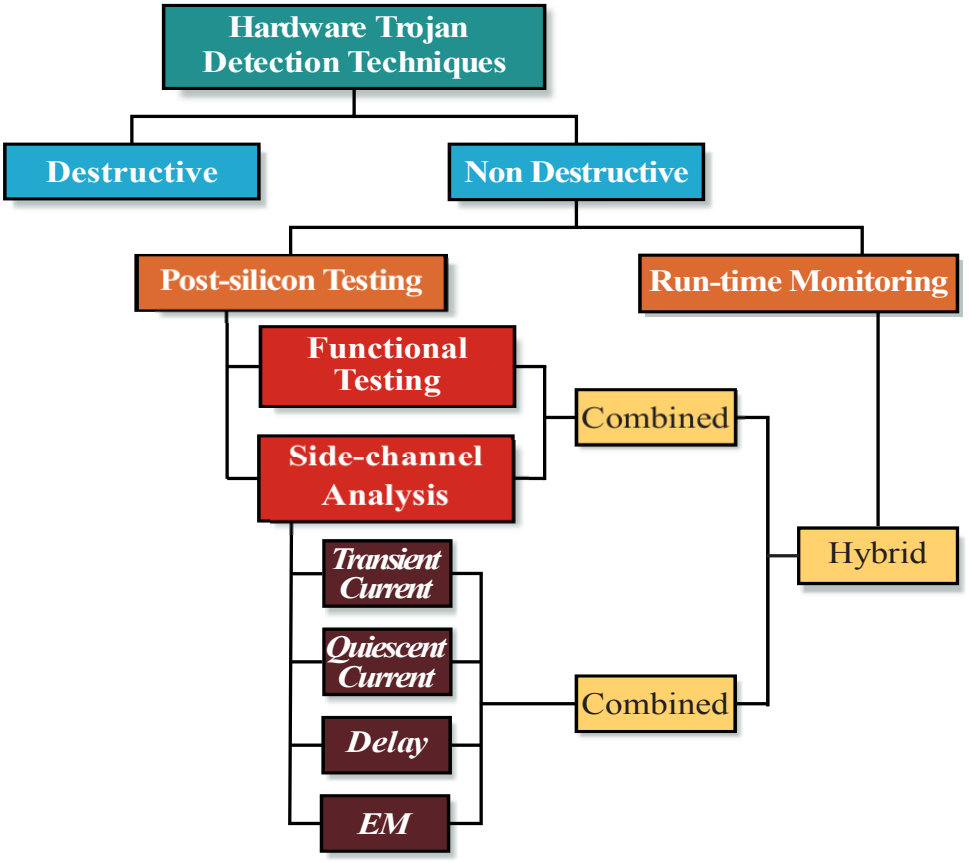


Fig. 2.7: Classification of hardware Trojan detection techniques [48, 64, 69, 94, 130, 132].

2.2.1 Destructive: Physical Inspection

A brute-force strategy for Trojan detection after fabrication is reverse-engineering the manufactured DUTT in order to recover its layout and look for discrepancies in relation to the original trusted one. This approach is possible thanks to high precision optical and scanning electron microscopes (SEM) after chemical mechanical polishing (CMP). In [18], it is shown the possibility of detecting HTs with only a top-layer image in the CMOS 130 nm technology if the placement or the routing have been redone by the foundry. The image cross-correlation calculations are applied to extract discrepancies between the original and the DUTT layouts. However, problems such as image noise, size and layer-level of Trojan modification make the detection considerably challenging. A SEM-based method was performed in [34] to detect Trojans in the same technology. In [152], filler cells are designed to be highly reflective at near-infrared wavelengths in order to produce easily measured watermarks of the genuine layout.

Despite presenting reliable results, these techniques feature some drawbacks such as being expensive, time-consuming, destructive, and difficult to be integrated into the regular testing phases. Moreover, to validate the manufactured lot of a particular IC, it would be necessary to sample at a least a few DUTTs to test it. Hence, even though a DUTT is stated as Trojan-free, it cannot be deployed after the physical inspection. Due to such inconveniences, the electric tests are more suitable, even if optical inspection offers accurate results. Besides that, as far as the technology nodes become smaller, the gates and connections in an IC will be more dense and compact, making this approach more and more difficult in a near future [48]. On the other hand, the approach is certainly applicable to certify a sample DUTTs as Trojan-free ICs (i.e. golden ICs), providing a set of fingerprints that are indeed referential data collected before the physical inspection alters the chips. The data from genuine ICs can be used afterwards to be statistically compared with results from the DUTT, allowing to classify it as infected or not.

2.2.2 Functional (Logic) Testing

Functional or logic testing techniques are originated from regular verification and validation phases. Its operation consists in evaluating the behavior of primary outputs and internal nodes of circuit given a set of input vectors. If the DUTT presents suspicious deviations or properties, the design is assumed to be Trojan infected. Functional tests can therefore be performed to detect Trojans at any step of the IC design.

There are different approaches addressing functional testing. A common approach is defining and identifying possible suspicious nets in the netlist. The authors in [19] proposed a method for finding weakly correlated signals or isolated sections in the netlist to find possible HT triggers. In [107], the authors compare Trojans in the literature in order to define architectural patterns frequently used in HT designs. Thereafter, they implement a score-based classification method to detect Trojans in untrusted netlists. Both methods are able to detect gate-level

Trojans in non certified netlist based on their assumptions about the Trojan model.

The studies [15, 30, 117], propose detecting suspicious activities caused by Trojans in third party IPs. In addition, if a complete trusted specification is available, a high-level golden model can be generated to perform a formal verification method such as sequential equivalence checking (SEC) to identify a possible HT. However, without having any trusted specification of the design, it is consider as a black box, rendering the HT detection quite challenging.

Techniques like in [25, 65] are able to detect Trojans implemented in different levels by applying data vectors at DUTT primary inputs with the intention of stimulating the HT activation and checking possible modifications at DUTT primary outputs caused by active Trojans. In [40, 78, 119], the authors presented test generation strategies to optimize the number of test vectors needed to activate a Trojan.

Furthermore, other approaches such as the one presented in [27] are implemented to maximize the probability of triggering Trojans by inputing test pattern based on multiple excitation of rare logic conditions. This method allows the reduction of the number of required test vectors compared to a weighted random pattern. With the activation of the HT, the effectiveness of functional tests for its detection is fairly enhanced.

2.2.3 Side-Channel Analysis

Another approach used to detect hardware Trojans is side-channel analysis [10]. These techniques are based on the fact that Trojans, even inactivated, cause leakages in terms of power, delays and EM emissions [132]. If a golden model is available, the Trojan detection is performed by comparing the side-channel traces from certified Trojan-free devices (golden ICs) and DUTTs.

In side-channel analysis, security designers have to deal with two main challenges: (1) the process (PV) and environment variations – which possibly masks Trojan effects in the side-channel signals–, and (2) the need for a golden model. The effects of PV basically results in an alteration of circuit parameters such as threshold voltages (V_{th}), channel lengths (L), and oxide thickness (T_{ox}). For instance, V_{th} can approximately fluctuate 20% among its original value in modern technologies [33]. Thus, ultra-small Trojans – sized on the order of 100 to 10000 times smaller than the original circuit dimensions – would naturally be masked by PV. Therefore, design and test efforts must be considered in order to reduce or compensate the PV effects. Each method proposes different strategies with this purpose. The need for a golden model is overcome by collecting signatures from golden ICs obtained from devices certified by physical inspection or certificated fabrication process. Furthermore, the methodology in [83] shows that is possible to generate fingerprints only based on trusted simulation models and measurements from process control monitor, without requiring certified ICs.

An illustrative example of the detection procedure is presented in Fig. 2.8. A certain input vector is applied in the primary inputs of a set of golden ICs and thus, the side-channel signals

are collected to produce a golden signature in a space of parameters. The same test procedure is performed in each DUTT, producing data to be compared with this golden signature. In Fig. 2.8, the golden data are used to generate error ellipses surrounding it considering a different significance levels (α_1 and α_2). Thereby, if the data obtained from a certain DUTT is outside this ellipse, this device is classified as Trojan-infected with the significance level previously considered to generate the error ellipse.

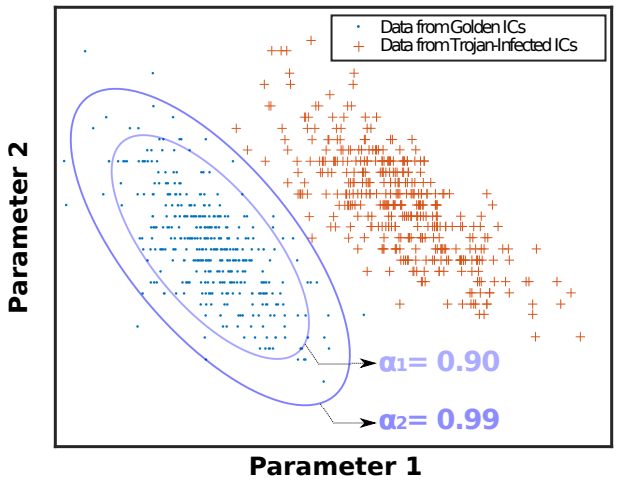


Fig. 2.8: Space of parameters generated by the data obtained from golden devices – error ellipses with difference significance levels $\alpha_1 = 0.90$ and $\alpha_2 = 0.99$ surround it –, and from Trojan-infected DUTTs.

The HT detection problem is defined in terms of a statistical hypothesis testing:

$$\begin{cases} H_0 : \text{The DUTT is Trojan-free} \\ H_1 : \text{The DUTT is Trojan-infected,} \end{cases}$$

in which the detection is performed whenever the null hypothesis H_0 is rejected. For the illustrative test of Fig. 2.8, the region of acceptance of the null hypothesis H_0 is given by the ellipses, generated according to a significance level α_i. A specific data outside this ellipse has the probability $1 - \alpha$ to reject H_0 and therefore accept H_1. Moreover, if the data is obtained from a set of known Trojan-infected DUTTs, the total number of points outside the ellipse normalized by the total of tests denotes the true negative rate (or the detection rate) which is the number of well-succeed detections. In addition, the total amount of data generated from Trojan-infected devices located inside the ellipse normalized by the total number of events denotes the false negative rate (F_N), also refereed to in literature as false alarms. The relation between the false negative rate F_N and the detection rate D_R is given by Eq. 2.1.

$$D_R = 1 - F_N \tag{2.1}$$

2.2.3.1 Parameters

Different side-channel signals such as transient current (I_{DDT}) [10], quiescent current (I_{DDQ}) [8], path delay [66] and EM [60] traces are used to generate signatures from golden devices and DUTTs. Besides that, combination of intrinsically related parameters were also proposed as a solution to compensate PV effects [96]. In the following analysis, different methods using these parameters are presented and discussed.

- **Transient Current (I_{DDT}):** As HT circuitries share the same power supply with the target system, traces obtained from the power supply pins can track possible alterations caused by a Trojan by evaluating the generated current trace. For this purpose, the switching activity in the circuit is used to gain information about the amount and the type of gates consuming dynamic power. The first Trojan detection method using side-channel analysis [10] used indeed the power trace generated by the transient current to gather a set of fingerprints of Trojan-free and Trojan-infected DUTTs. In this study, a Karhunen–Loève (KL) expansion is used to eliminate the measurement noise and therefore perform the detection. Alternatively, further studies have also addressed detecting Trojans even in the presence of PV. The approaches presented in [114, 115, 136] rely on measuring multiple power ports or pads individually in order to isolate the Trojan effects to a specific chip location and thus increase its relative impact. In [136], the strategy was integrate the total current from a specific pad, while in [114, 115], similar methodologies used the I_{DDT} provided by each power port (see Fig. 2.9) as parameters to generate scatter plots PP_{xx} vs. PP_{yy}.

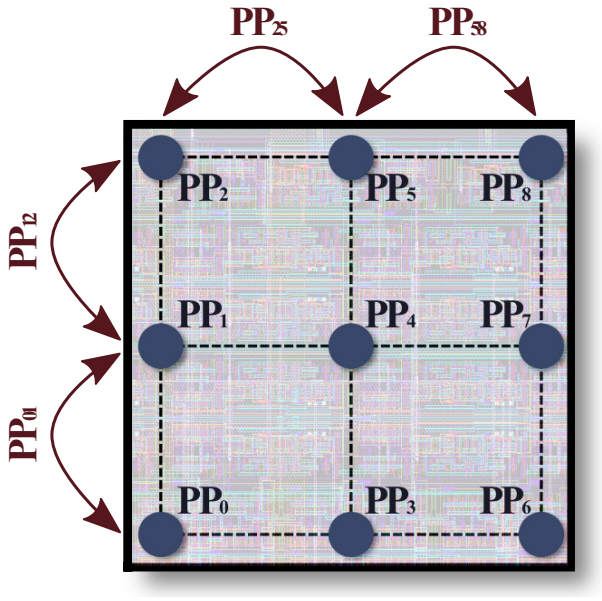


Fig. 2.9: Chip partition approach based on the measurement of multiple power supply pins for Trojan isolation [114, 115].

In [20], an on-chip current sensing structure combined with a power gating architecture is used to isolate Trojans and produce digital results carried out by a scan chain. Silicon demonstrations of Trojans design and detection in wireless cryptographic ICs are shown in [84]. In this work, authors present an always-on Trojan able to leak keys of a 128-bits AES core and detect it by measuring the transmission power obtained in different inputs.

- **Quiescent Current (I_{DDQ}):** Quiescent current leakage adds by a HT is another traceable parameter to identify it, even in scenarios where there is no switching activity in any Trojan nets. In [8, 140], the authors demonstrated the effectiveness of analyzing I_{DDQs} measured simultaneously from multiple locations of the chip. A test structure is used in order to emulate the Trojans in different positions in the circuit and perform its detection by measuring multiple power ports. Despite requiring distinct input vectors, test procedures for obtaining I_{DDQ} are very similar to the ones used in I_{DDT}. Most of the proposed methods using I_{DDQ} consider it as an auxiliary signal in multiple parameter analysis.

- **Path Delay:** Another consequent effect of a HT infection is the delay addition encountered in specific nodes of the original circuit. A Trojan inserted between two blocks modifies the authentic datapath and thus increases a delay in such nets. In another possible implementation, it is directly connected to an original node of the circuit –without necessarily cutting lines of device's nets– enhancing the fan-out and capacitive loads of the previous gate and therefore, increasing the path delay.

  Measuring path delays in sequential circuits after fabrication, however is not a simple task. If no extra on-chip circuitry is used for this purpose, it is only possible to measure path delays which originates from primary inputs and terminate at primary outputs. Besides that, as in synchronous circuits the clock controls the data flow from stage-to-stage, it is not possible to measure the delay of each stage. For this reason, extra on-chip circuits such as full-scans must be used to enable the measurement of these delays during post-silicon testing phase. Indeed, Trojan detection techniques based on path delay rely on using mechanisms able to output variables indicating the path delay. In [80], a delay characterization is done by a secondary clock signal controlling shadow registers. In Fig. 2.10, a diagram shows the architecture used in this work. The original clock (clk1) drives the regular registers, a secondary clock (clk2) controls a shadow register –used specifically for detection of Trojans– and the outputs of both are connected to a comparator. The test procedure is basically retarding gradually the clk2 in relation to clk1 until the comparator indicates that outputs from registers are different. At this time, the skew between clk1 and clk2 denote the delay of this combinational path. The same test procedure must be repeated for each pipeline stage of the circuit in order to characterize path delays of the whole circuit.

  Other detection methods propose improving this technique effectiveness by using the

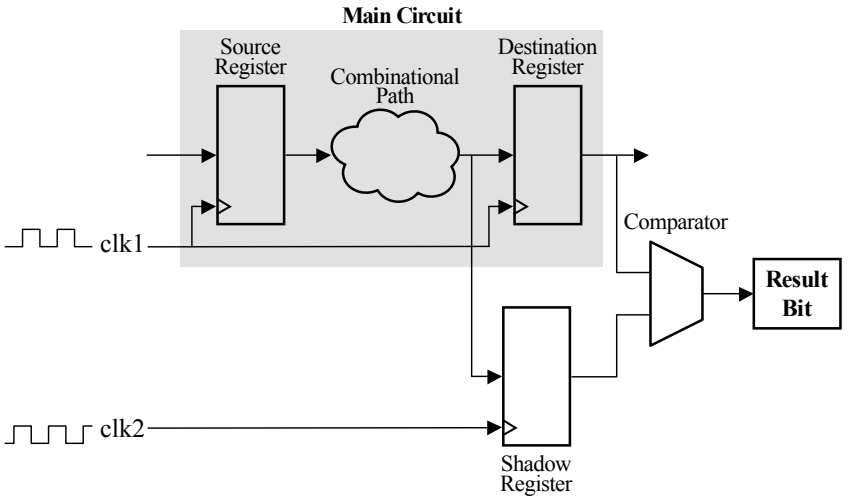


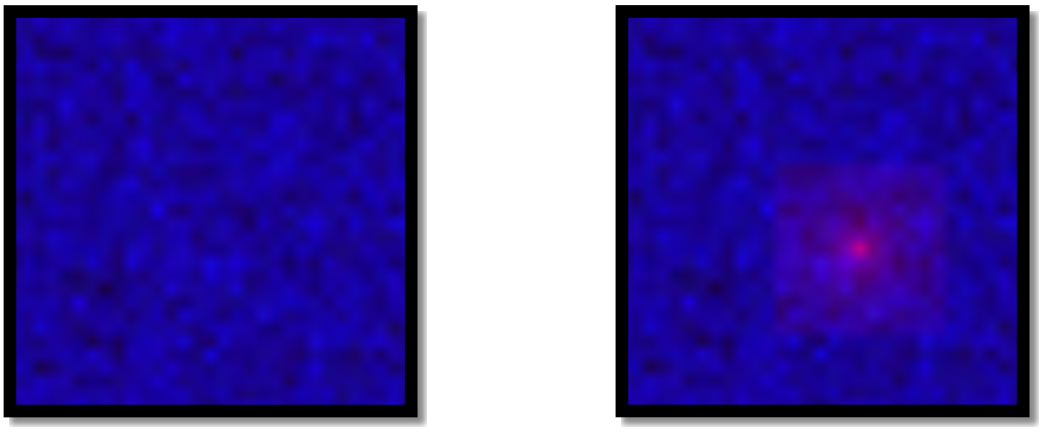
Fig. 2.10: Delay characterization with negative-skewed shadow registers [80].

comparator outputs as chip authentication [81]. In [21], authors proposed selecting more vulnerable paths to and input vectors to activate them. Moreover, other techniques used embedded test structures for on-chip measurements of path delays [62, 74, 75] while in [36] a framework based on self-authentication is proposed. In [66], an effective manner to gather the fingerprint of all path delays is proposed while a more recent approach [146] uses latch-structures to compare relative delays of different paths in the circuit to identify discrepancies.

- **EM Emissions:** Switching activity in Trojans nets is a source of unsuspected EM emissions. Non-invasive techniques are therefore used to track DUTT emissions and compare them with a golden model. Prior studies [14, 129] use similar approaches to detect Trojans inserted in different locations in FPGAs. The analysis consists in scanning the whole circuit with an EM probe able to collect data from different spots of the circuit. Once collected, the golden and DUTT data are compared in order to generate a map depicting the obtained differences between them. In addition, the technique in [60] uses a thermal map with the same purpose. In Fig. 2.11, an illustrative map presents how results are generated considering different Trojan locations. Each pixel of the EM map represents a specific spot in the circuit and its color the difference between DUTT and golden device.

  Recently, [56] reported another methodology able to detect Trojans using EM emissions without the need of neither a golden IC nor the netlist. In their study, RT-level simulations generate patterns to be compared with the ones obtained from FPGA to perform the detection of activated Trojans.

- **Combined Parameters:** Another efficient approach consists in combining signatures obtained from different side-channels and thus, increasing the amount of obtained data to enhance the Trojan detection effectiveness. Moreover, the intrinsic relation between



(a) (b)

Fig. 2.11: EM emission maps generated considering a Trojan free DUTT (a); and a Trojan-infected DUTT (b). For each pixel, blue shades represents low difference between EM emission from golden device and DUTT, and red shades denotes considerable discrepancies [14, 129].

different side-channel signals is a clever strategy to compensate PV effects [147]. For instance, if PV acts increasing the power consumption of a specific logic-gate, its consequent effect is reducing its path delay. Thus, the value of the power consumption of a given gate allows predicting its path delay in this PV environment. Authors in [96] take advantage of it to propose a detection technique considering transient current and path delay (obtained indirectly by the maximum operation frequency). In [59], the relation between transient and quiescent current are used while in [103], the technique uses delay and electromagnetic measurements to detect Trojans. An unified framework is proposed in [70] providing detection results for all side-channel signals and thereafter combining them.

2.2.3.2 Implementation Cost

As presented in the previous sections, side-channel analysis detection techniques require strategies with considerable efforts in terms of design and test to compensate PV effects. The most used design-level approach consists of splitting the original circuit in several measurement domains in order to isolate the Trojan impacts to a specific domain [8, 21, 136]. These strategies necessarily require extra on-chip circuitry or multiple pads to separate the subcircuits signals and additional post-manufacture tests and dedicated test set-ups to generate all needed signatures, thus increasing the project cost.

Most of path delay based methods uses a secondary clock signal to control a set of shadow registers. Similarly, most of current based methods require measuring the signals from multiple power supply pins in order to isolate the Trojan and applying power gating to enhance the detection rate. Moreover, even if EM-based methodologies does not require necessarily extra hardware in the original design, high resolution devices to analyze EM or thermal maps are

needed, increasing time-to-market and set-up cost of testing phase, besides the challenging procedures for nanoscale technology nodes.

2.2.4 Run-time Monitoring

Logic and side-channel signals are evaluated by run-time monitoring structures embedded in the original design. In this case, if a Trojan is activated after the deployment phase, the monitoring system is able to generate a flag indicating a Trojan alert. In [35, 39], techniques treat interferences in circuit functionalities caused by activated Trojans as fault-models and thus detect it. Furthermore, the technique presented in [47] relies on monitoring the temperature of the circuit. Extra power consumption caused by a Trojan causes a discrepancy between the expected and the measured temperature, making the method able to detect Trojan-infected circuits.

2.2.5 Combined and Hybrid Methods

In order to combine benefits from side-channel analysis, functional testing, or run-time monitoring, authors have proposed mixed techniques able to detect Trojans by several means. In this subsection two main combined methods – Ring Oscillator and Gate-Level Characterization (GLC) – are presented.

2.2.5.1 Ring Oscillator

Detection techniques presented in [58, 67, 76, 149] propose the implementation of a set of RO distributed over the original layout. If a HT is inserted close to one of the RO that composes the network, the V_{DD} drop caused by the Trojan slightly reduces the RO supply current, therefore modifying its frequency. By analyzing this deviation, designers are able to track Trojans. This method combines the leakage current and the functional behavior of ROs, making it able to detect even inactive Trojans. Fig. 2.12 illustrate the basic implementation of RO based methods. Besides that, while ROs keep running after IC deployment, this method can also be used as a run-time monitoring, increasing its efficacy if the Trojan is activated. In a context in which the adversary knows this HT detection method, he is able to simply change the size of the gates that composes the RO, leading it to run in the same expected frequency. Nevertheless, this cat-and-mouse game action increases the supply current, making the HT detection by other side-channel analysis method more likely, forcing the adversary to improve even more his attack.

2.2.5.2 Gate-Level Characterization (GLC)

In [113], GLC were firstly proposed for detecting Trojans with the support of a trusted netlist. The method is based on characterizing the side-channel signals of all gates from a specific

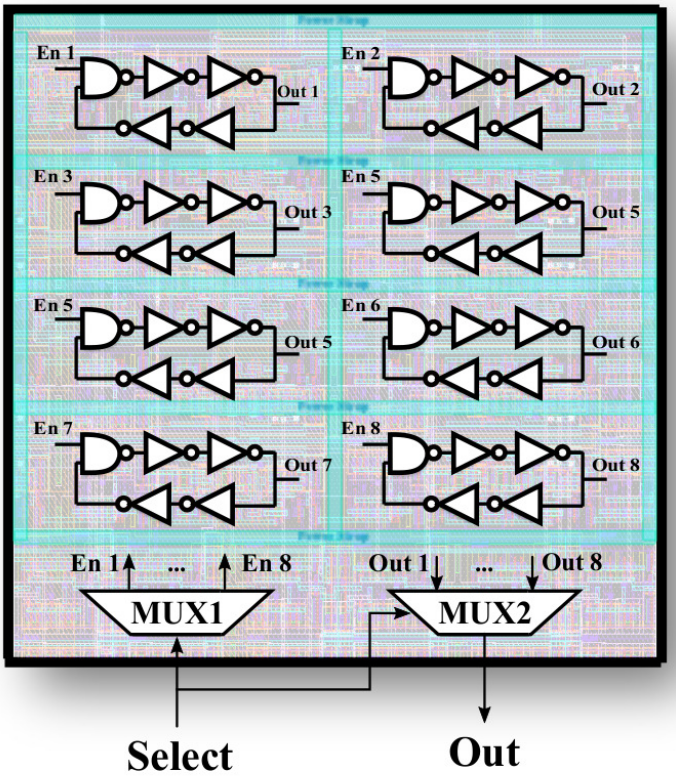


Fig. 2.12: A basic RON structure [149] with 8 ROs distributed in the circuit layout.

segment of the manufactured device. The technique is basically solving the linear equation 2.2 to find the PV scaling factor of each gate.

$$\sum_{j=1}^{n} \alpha_j \, I_{k,j}^{nom} = \hat{I}_k^{total} \tag{2.2}$$

where α_j is the PV scaling factor and $I_{k,j}^{nom}$ is the nominal leakage power for the gate j given an input vector k, and $\hat{I}_k^{total}$ is the total measured leakage power for the k-th input vector.

By having a number of input vectors larger than the count of all gates in the circuit segment under test, it is possible to solve the system of equations stated in 2.2. Their solutions are combined in order to statically estimate bounds for the calculated scaling factors. Basically, if the results of α_j diverge to different values as far as the number of tests increases, the nominal model does not represent the manufactured DUTT and thus, it is classified as Trojan infected.

Other methods were proposed using the same GLC principle. In [138] authors propose using thermal control process to characterize the PV scaling factors for all gates while in [139], the same authors propose overlapping circuit segments in order to find inconsistencies between the obtained results, indicating the presence of a Trojan. A similar approach for inconsistency-based characterization was proposed in [11] considering the quiescent current as side-channel signal

in analysis. Moreover, optimizations in terms of the number of required tests are presented in [148] and, in [31] statistical learning algorithms are used in order to enhance the Trojan detection efficiency with low cost tests under high measure error rates.

2.2.6 Trojan Prevention

In addition to Trojan detection methods, other design-level strategies were proposed in literature to prevent its insertion. In [145, 153], homomorphic encryptions are proposed to hide the authentic inputs and outputs of untrusted IPs or systems, preventing adversaries to be aware of real input states of the circuit.

Furthermore, other approaches [13, 144] consists in filling unused spaces in the circuit layout with functional filler cells instead of nonfunctional ones. Hence, possible free spaces in the circuit for Trojan insertion are removed, hampering its placement in the original layout. In [23], an obfuscation methodology is proposed to prevent attackers at fabrication level to get knowledge about the rareness of a given node of the design. As the adversary is not aware of a rare event in the circuit, a possible Trojan implemented would be easier detectable.

2.2.7 Overall Analysis

The discussion held in this section is summarized in Table 2.1, which compares the main Trojan detection categories: functional (logic) and side-channel analysis according to different aspects related to their properties. Briefly, functional analysis is only effective if the HT is active or the designer have clues about the HT design to identify suspicious nodes in the circuit. On the other hand, side-channel analysis methods are able to detect even inactive HTs with the cost of requiring a golden model, which is a set of golden ICs for most of the methods. These techniques also have the inconvenient of dealing with PV masking effects, making the detection of small Trojans quite challenging. Therefore, the combination of both have been the most promising solution for HT detection, since positive aspects from either approaches can be matched, increasing the technique effectiveness. For this purpose, the Table 2.2 is presented synthesizing most of the side-channel analysis and combined methods described in this section according to its strategy and implementation cost.

Table 2.1: Comparison of the main Trojan detection categories.

Method	Requirement for Trojan Activation	Requirement for Golden IC	Difficulty to Detect Small Trojans	Implementation Cost	Trojan Detection Effectiveness
Functional Analysis	Yes for most of the methods	No	Ok for most of the methods	Increment of verification and validation testing time	Low
Side-Channel Analysis	No for most of the methods	Yes for most of the methods	The smaller the Trojan, the more difficult the detection	Design or test high cost depending on the method	High

Table 2.2: Comparison of the main side-channel analysis based detection techniques.

Technique	Transient Current	Quiescent Current	Path Delay	EM	Trojan Level	Chip Validation	Implementation Cost
Rad et al. [115]	✓	✓			Gate		Design: extra on-chip circuitry
Aarestad et al. [8]		✓			Transistor (active)	✓	Design: multiple power pins and extra circuitry
Wilcox et al. [140]		✓			Transistor (active)	✓	Design: multiple power pins
Potkonjak et al. [113]	✓	✓	✓		Gate		Testing: extensive testing
Alkabani and Koushanfar [11]		✓			Gate		Testing: extensive testing
Wei and Potkonjak [139]	✓	✓			Gate		Testing: extensive testing
Hu et al. [60]				✓	Gate		Testing: dedicated set-up
He et al. [56]				✓	RT, Gate	✓	Testing: dedicated set-up
Ngo et al. [103]			✓	✓	Gate	✓	Testing: dedicated set-up
Söll et al. [129]				✓	RT, Gate	✓	Testing:dedicated set-up
Narasimhan et al. [96]	✓	✓ (possibly integrated)	✓ (Max. frequency)		Gate	✓	Design: multiple power pins and extra circuitry
Forte et al. [47]				✓	RT, Gate		Design: extra on-chip circuitry
Jin and Makris [66]			✓		Gate		
Lamech and Plusquellic [75]			✓		Transistor (active)	✓	Design:extra on-chip circuitry
Ismari et al. [62]			✓		Gate	✓	Design:extra on-chip circuitry
Rad et al. [114, 115]	✓				Gate		Design: multiple power pins

2.3 Conclusion

In this chapter, the different vulnerable design phases of IC have been presented. Thereafter, the full Trojan taxonomy depicting possible scenarios in which Trojans can be inserted are presented showing that regular steps of IC production are susceptible to Trojan insertion. Moreover, diverse Trojans, implemented at different abstraction levels, reported in recent bibliography illustrates possible attacks and concerns that security designers must consider during the design of their ICs.

In the second section, the main HT detection techniques are classified according to their strategies to detect Trojans. It is shown that many works in the literature have address different vulnerable steps of the IC production chain and propose attractive solutions. The techniques are also compared in terms of properties and implementation costs.

The Trojan concern is, however, far from being overcome. Adversaries aware of the main detection methods may develop more sophisticated Trojans able to be undetectable by the known techniques. Nevertheless, while more and more methods are proposed, more difficult is to an attacker to design a Trojan undetectable by all these methods. Consequently, the development of new innovative techniques is the key factor to make difficult possible attacks and thus increases the ICs trustworthiness against Trojans.

Chapter 3

Analysis of Transistor-Level Trojans in Ring Oscillators

Today's security systems require more and more reliable components for generating unpredictable random data, therefore ensuring secure authentication process is mandatory. A commonly used component for generating aleatory data is ROs. For example, most of true random number generators (TRNG) use a set of ROs as source of entropy [32, 52, 128, 143]. Similar concepts are employed in architectures of physical unclonable functions (PUF) in secure IC, which are used to provide unpredictable and unique identifications for a circuit [88]. Both TRNG and PUF are possible targets for HT attacks envisaging the range reduction of randomness by mitigating the sources of entropy.

The RO entropy is proportional to the jitter-frequency product [128], therefore if it is reduced by the inclusion of a HT, TRNG and PUF would become less random, and thus more predictable. Moreover, RO are also useful to produce periodic signals of high frequency for clock-based integrated systems. Hence, any modification of the RO frequency could generate delay errors that would lead systems to failure.

This chapter in sections 3.2 and 3.3 introduces three types of simple and low-area HT that efficiently modify the properties of RO, the Trojans are easily implementable at transistor level. This work was the topic of our publication [5] and its main contribution are: (1) demonstrating for the first time transistor-level HTs in ROs; and (2) warn and show that the tri-state logic property, if applied on HT, consistently prevents typical post-manufacturing functional testing and timing-based side-channel analysis from detecting the Trojans. As the tri-state feature allows a high-impedance isolation between HT and RO, HT can be set off, keeping Trojan-induced delay variations of RO elements in corner specifications. If triggered by some rare external events, like the fault injections described in [72], the presented HT would set RO with harmful frequencies and jitter that are out of corner ranges. Furthermore, thanks to their small sizes,

the resulting power overheads are lower than the power variations in process corners, and thus current-based side-channel analysis [10] are not capable to detect them as well. Finally, assuming the area of the affected target system is much larger than the presented HT, the expensive and time-consuming method of physical inspection [72, 130] would become difficult too, besides depending on the die samples elected for analysis having or not HT.

3.1 Motivation: Attack in RO-based TRNG

RO is a circuit composed of an odd number of gates and a feedback signal able to switch periodically the logic state of the first gate and the others, generating a steady periodic output. Fig. 3.1 illustrates the RO architecture taken into account in this work, the first RO stage is a NAND gate that allows the oscillation starting.

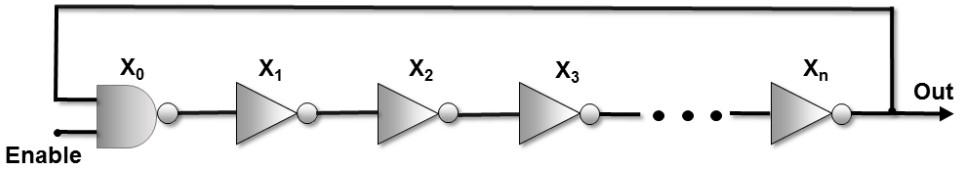


Fig. 3.1: A Trojan-free (n+1)-stage ring oscillator (RO).

RO circuits are widely employed as high-speed frequency generators and as fundamental sources of jitter in security systems demanding random outputs, such as TRNG and PUF [88]. RO-based TRNG, for example, are the most common architectures [16] and requires a high number of RO. Its basic architecture is depicted in Fig. 3.2. TRNG randomness depends on its entropy, which is proportional to the number of RO and the jitter-frequency product of each RO [128]. Thus, the minimum number of RO for designing a TRNG is given in function of the jitter-frequency product. Since the number of RO in a TNRG remains unchanged after its design, if the jitter-frequency product of a RO is decreased by the addition of a HT, the entropy and, therefore, the TRNG randomness would be reduced, rendering the system more vulnerable to prediction-based attacks that seek leaking information.

3.2 Tri-State Trojans on Ring Oscillators

The Trojans are implemented with their activation mechanisms designed with the purpose of preventing Trojan activation during functional testing routines. If attackers have the possibility to access the manufactured device after the post-silicon testing phase, the trigger circuit can be designed to be activated by side-channel attacks. For instance, Trojans could be designed to be stimulated by fault-injection events produced by external sources (e.g. lasers, temperature

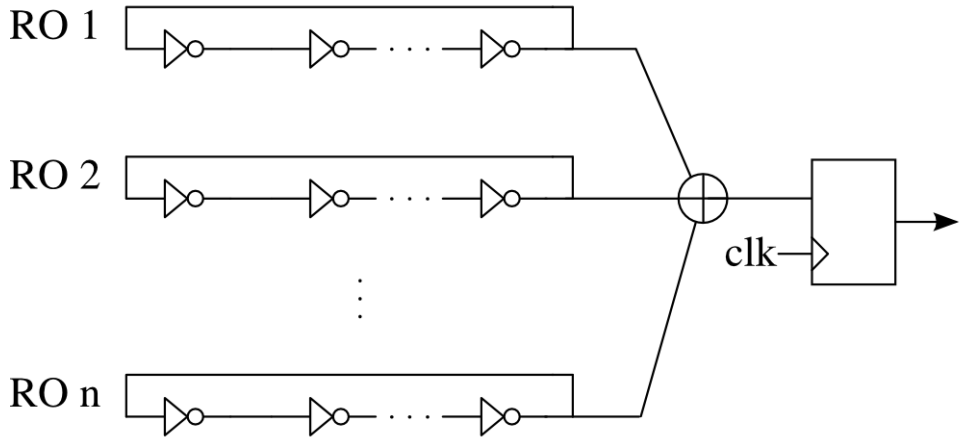


Fig. 3.2: The RO-based TRNG design.

alterations, or V_{DD} variations [72]), thus the trigger circuit could be a single memory bit composed of only cross-coupled inverters (4 transistors) and a reset transistor to guarantee an initial logic state in the trigger output. During a regular system start-up, the trigger output could be initialized, and thereafter, attackers would be able to simply focus on generating a single bit flip in the trigger circuit to produce the activation of the payload circuit.

The Trojan payloads proposed in this chapter are based on the tri-state logic property in which a high-impedance state allows isolating HT payload output from the affected node of the target RO circuit. During normal system operation, HT is inactivated and isolated by a high impedance, causing no anomaly in the circuit behavior. If activated, HT output assumes low-impedance state, which means that Trojan interacts with the affected node, and thus, the circuit behavior is considerably modified.

3.2.1 Capacitor Trojan

One of the most simple attacks to RO is increasing its delay by inserting a tiny capacitor (about 10fF for CMOS 65 nm technology) in an arbitrary node of the target RO. Fig. 3.3 shows the proposed modification using only two simple elements: an activation NMOS and a capacitor. The activation NMOS has a trigger signal connected to its gate terminal, which allows the attacker to activate the Trojan with a previously designed condition. This malicious alteration allows the attacker to decrease signal frequency when the Trojan is active.

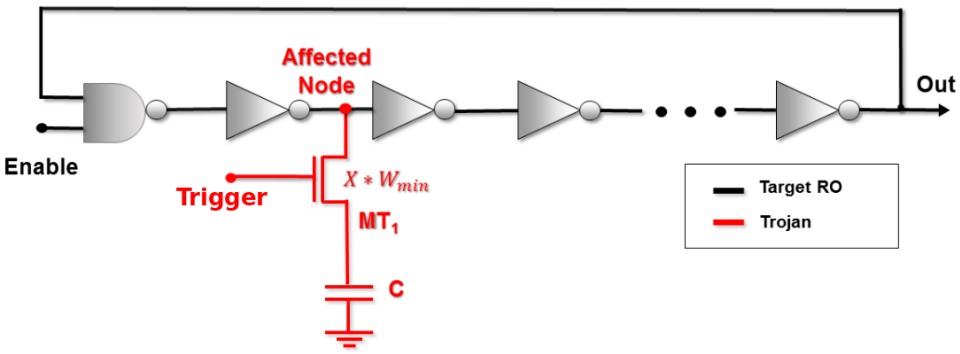


Fig. 3.3: Capacitor Trojan isolable by high impedance from the RO.

3.2.2 Double-Switch Trojan

This payload is a tri-state gate with the $\frac{W}{L}$ ratio for its transistors multiplied by a design factor X (see Fig. 3.4). Decreasing the design factor X increases the impedance of the tri-state, then it becomes a source of noise for the target RO. In regular RO designs, the rate between the active time (T_{ON}) and inactive time (T_{OFF}) is set to be 1 (i.e. duty cycle $D_{\%}$ is expected to be 50%). However, double-switch Trojans can unbalance this output due to the current driven to its target node. According to the $X \times \frac{W}{L}$ of its transistors, the attacker is able to modify the duty cycle $D_{\%}$ of RO output.

$$D_{\%} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\% \tag{3.1}$$

Adjusting X to be large enough, the HT would make the duty cycle approach 0% if the control node is assumed to be to be at the logic state '1'. If its state is set to '0', the duty cycle is 100%.

3.2.3 Transmission-Gate Trojan

The transmission-gate Trojans have a similar architecture as the double-switch HT. Nevertheless, its control node is connected to the affected node 1 of RO through a transmission gate. Thus, the target node of the tri-state buffer will follow the inverted logic of node1, exactly as it is expected to behave if it was a Trojan free circuit. Therefore, whenever activated, the payload keeps RO working. However, it changes the node delay and the frequency of the RO output, giving the attacker control of RO frequency and jitter variations with respect to the chosen $X \times \frac{W}{L}$ of its transistors. If the control node of HT was connected directly to node1, the Trojan would be more intrusive than others during inactivated conditions. To overcome this problem and isolate the node1 from the Trojan, transistors M_{T5} and M_{T6} constitute a transmission gate (see Fig. 3.5).

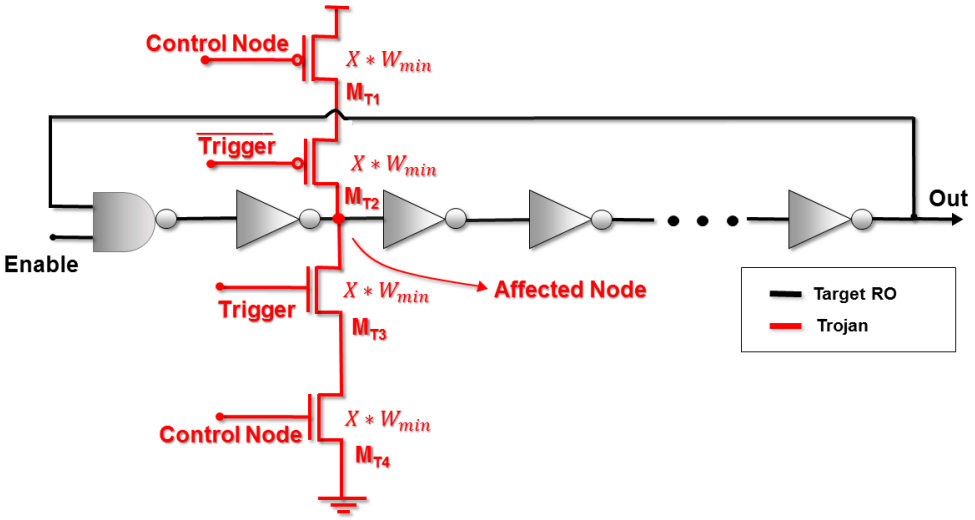


Fig. 3.4: Double-switch Trojan isolable by high impedance from the RO.

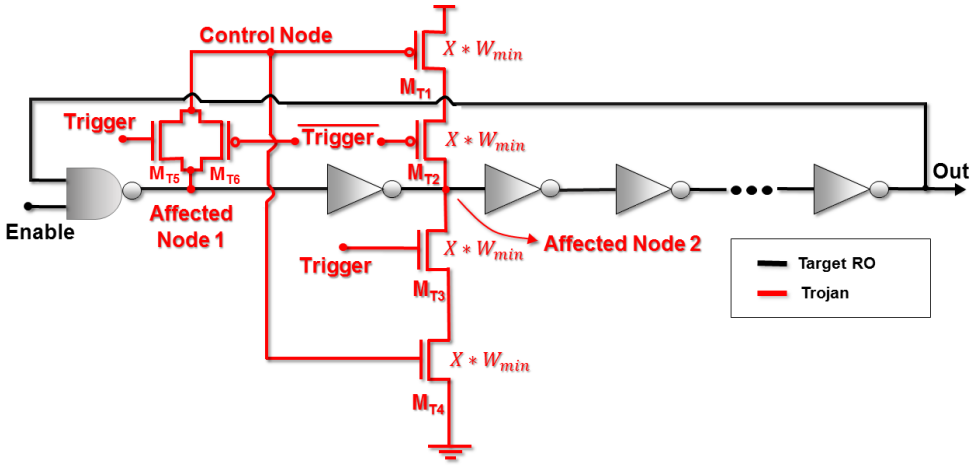


Fig. 3.5: Transmission-gate Trojan isolable by high impedance from the RO.

3.3 Trojan results in Ring Oscillators

Electrical-level simulation results of 7-stage RO circuits in commercial CMOS 65 nm technology are presented in this section. RO were designed with standard cells with the technology minimal dimensions, V_{DDD} 1.2V, nominal conditions, standard threshold voltage (SVT), considering a white noise in order to generate jitter. In addition, the same ROs are simulated with the Trojans detailed in previous section. Trojan transistors were sized with the minimum channel length $Lmin$, and the minimum diffusion width $Wmin$ multiplied by a design factor X. Power consumptions, oscillation frequencies, and jitter of RO are measured with three scenarios:

- Trojan-Free: target RO circuit with no HT. Trojan-Free simulations were performed at typical, FF, and SS corners in order to identify process variations induced on the power consumption, oscillation frequency, and jitter-frequency product of RO. Measures at FF

and SS corners define the ranges for evaluating the efficiency of the proposed Trojans;

- Trojan-OFF: target RO circuit with a HT that is not activated. This scenario allows verifying if a dormant HT penalizes the power consumption, frequency, and jitter-frequency product within corner ranges, otherwise functional testing and side-channel analysis are likely able to detect the Trojans;
- Trojan-ON: target RO circuit with a HT that is activated. This case indicates the HT potentials to harmfully modify the RO behavior.

Fig. 3.6 shows an example of Trojan operation in these three scenarios. In considering steady state of Trojan-Free scenario, RO frequency has the expected value for typical conditions. In extreme process corners (FF and SS), it is possible to predict maximum and minimum expected frequency for RO design. The behavior of the circuit infected by Trojan is also shown before (Trojan-OFF) and after (Trojan-ON) Trojan being activated (in $0.75\mu s$).

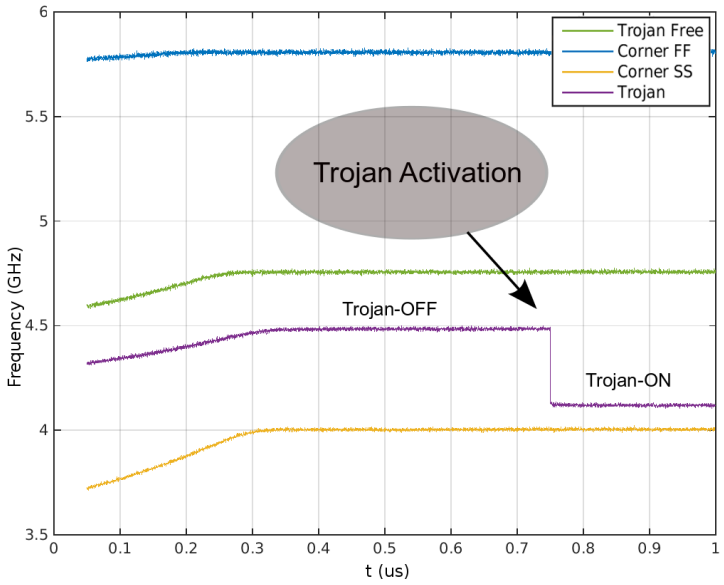


Fig. 3.6: Possible scenarios of Trojan insertion: Trojan-Free, Trojan-OFF, Trojan-ON and FF and SS corners of Trojan-Free circuit

3.3.1 RO Frequency Analysis

Simulation results in Trojan-ON and Trojan-OFF cases are discussed in the following subsections.

3.3.1.1 Trojan-ON

The frequency variations caused by HT can be verified in Fig. 3.7. It presents the rate between 7-stage RO frequency in Trojan-ON and Trojan-Free scenarios. Black lines represent capacitor

Trojan, green lines are the double-switch Trojan, and red lines highlight the transmission-gate Trojan results. Blue dashed horizontal lines represents of FF (above) and SS (below) corner results for Trojan-Free scenario. Thus, it is possible to observe Trojan influence in circuits, in comparison to corners FF and SS. The X axis of the graph is the design factor X of Trojan transistors.

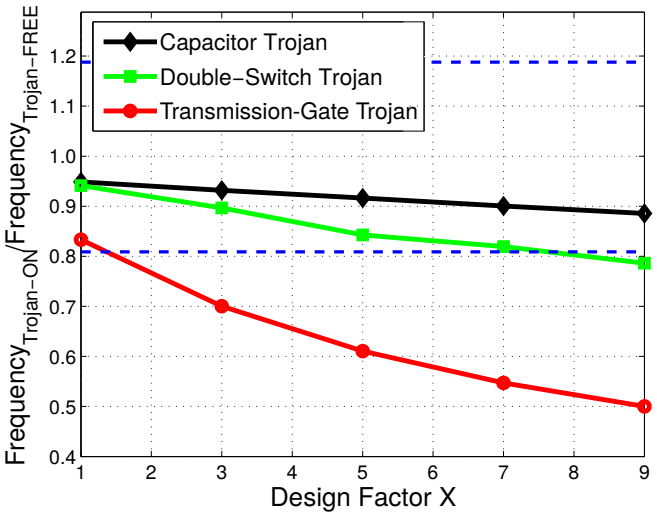


Fig. 3.7: Ratio of Trojan-ON frequency to Trojan-Free frequency vs. factor X of Trojan versions and corners FF and SS of the circuit Trojan-Free in a 7-stage RO.

In Fig. 3.7, it is possible to see that the larger the size of Trojan transistors, larger the frequency attenuation. In case of $W = 9W_{min}$, transmission-gate Trojan can halve the original RO frequency. Other Trojans also attenuate RO frequency, but it remains within corner limits for almost all set of tested design factors X. Capacitor-Trojan does not cause a delay larger than the corners in any situation. This is due to the fact that capacitor value is only $1.4fF$, the minimum load allowed in 65 nm technology. By increasing this capacitance, the HT causes a larger delay in RO and, then, affects more its frequency.

3.3.1.2 Trojan-OFF

It is possible to do the same analysis for Trojan-OFF scenario (see Fig. 3.8). In the case of results being within the corners limits, Trojans are able to pass undetectable by functional logic testing. Simulation results shows that all Trojans are within the corner limits if its design factor X is smaller than 3. However, for a factor larger than 5, transmission-gate Trojan are detectable by functional logic testing since its frequency is lower than the expected for SS corner. Other Trojans are not expect to be detected by functional logic tests by having its frequency variation always smaller than extreme conditions.

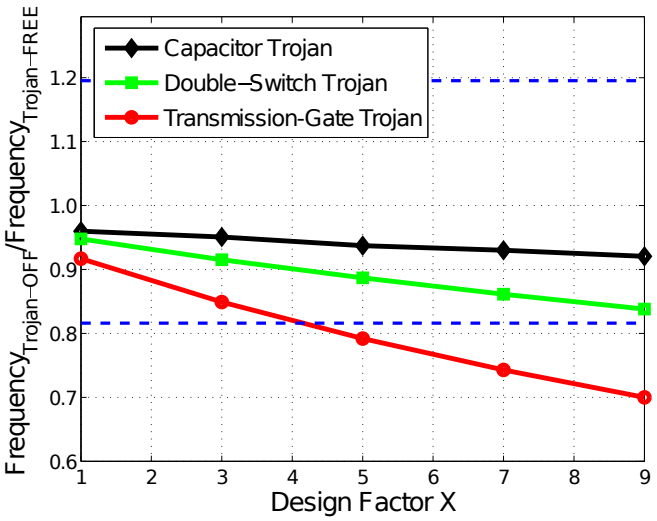


Fig. 3.8: Ratio of Trojan-OFF frequency to Trojan-Free frequency vs. factor X of Trojan versions and corners FF and SS of the circuit Trojan-Free in a 7-stage RO.

3.3.2 Power Consumption Analysis

Power analysis is another important study to be done since Trojans must not consume more power than target circuit in corners condition to avoid being detected by power-based (current-based) side-channel analysis. Simulations were only performed considering the Trojan-OFF case, since it is supposed that the Trojan remains inactive during functional tests. A testing strategy able to trigger the Trojan would likely detect it, however, as its activation depends on side-channel attacks or ultra-rare conditions, it is not considered in the following analysis.

Relations between power consumption of circuit with Trojan-OFF and Trojan-Free are made following the previous delay analysis. Fig. 3.9 shows simulation results.

The power consumption of Trojan-OFF circuits almost does not affect the target circuit overall consumption. In Trojan-OFF case, the power consumption modification is less than 5% for all Trojans. It indicates that Trojans would be very unnoticeable in most of power-based (current-based) side-channel analysis. Moreover, as the tri-state does not allows transistors to switch if trigger is off, the transient current in the Trojan is negligible. Due to the huge frequency attenuation caused by the transmission-gate Trojan, the dynamic power is reduced and, therefore, the global power consumption can be decreased by a factor of approximately 0.99 if the circuit is infected by this HT.

3.3.3 Jitter x Frequency Analysis

As stated in the section 3.1, it is possible to reduce TRNG entropy by decreasing the jitter-frequency product of their RO. Results are presented in Fig. 3.10 to study the impact of Trojans in this relation. Although all HTs reduce circuit frequency, jitter-frequency product is differently

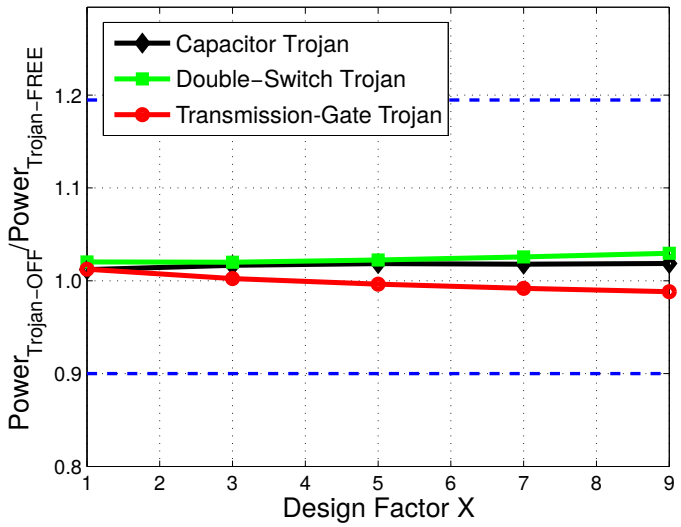


Fig. 3.9: Ratio of Trojan-OFF power consumptions to Trojan-Free power consumptions vs. factor X of Trojan versions and corners FF and SS of the circuit Trojan-Free in a 7-stage RO.

changed by each Trojan. As it is possible to see in the Fig. 3.10, only the transmission-gate Trojan decreases substantially this product. Jitter-frequency product in both SS and FF corners is greater than in TT, and thus, it is not showed in Fig. 3.10.

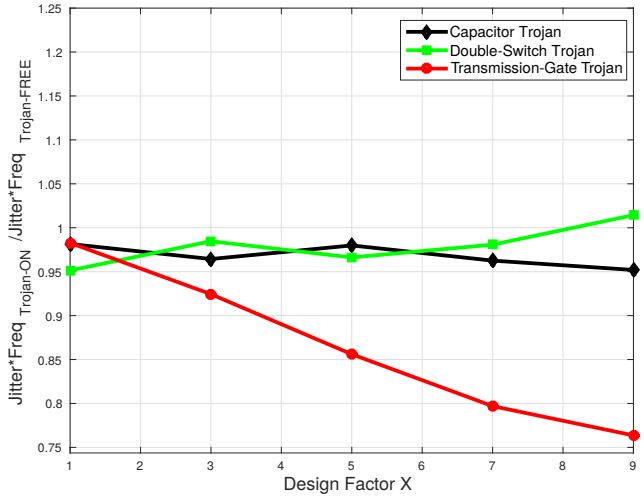


Fig. 3.10: Ratio proportional to jitter-frequency product of Trojan-ON / Trojan-Free vs. factor X of Trojan versions in a 7-stage RO.

3.3.4 Case study

In this subsection, a case study is presented in order to show an illustrative attack using one of these transistor-level Trojans. To this purpose, the transmission-gate Trojan with design factor $X = 3$ is chosen since it presents the best trade-off considering the analyzed parameters. If the HT is off, the circuit power consumption remains unchanged and its frequency is reduced by a factor of approximately 0.85. Thus, in Trojan-OFF scenario, circuit stays within corners limits. If this Trojan is switched on, it is able to reduce the frequency by a factor 0.7 and the jitter-frequency product by a factor of 0.92. Thus, if this Trojan is implemented in all ROs of a TRNG, it will be able to reduce its overall entropy and therefore its randomness by a factor 0.92.

3.4 Conclusions

ROs are fundamental elements for generating high-frequency signals in integrated circuits and ensuring sufficient entropy in TRNG and PUF. This chapter demonstrates the effectiveness of implementing transistor-level HTs in RO able to drastically impose consequences on TRNG, and on any clock-based system. The presented HTs are useful as models and test cases to future studies targeting new detection methods.

The most efficient HT that is demonstrated has only 6 transistors in its payload circuit (Fig. 3.9), and 5 transistors in trigger circuit (cross-coupled inverters and a reset transistor). Assuming simple integrated systems can have thousands of transistors, and complex ones billions, detecting just 11 malicious transistors would not be trivial.

Finally, with the discussed HT, we confirm the concerns announced by cited references in which the major challenges to deal with Trojans is to find new efficient detection methods capable to identify tiny circuit modifications within process corners.

Chapter 4

Bulk Built-in Current Sensors For Detection of Transient Faults

External sources of perturbation are able to disturb integrated systems motivating considerable design challenges. Perturbations due to environmental or intentional attack sources can lead today's circuits to have transient faults, temporarily modifying voltage levels, and provoking soft errors in stored results of operations. Examples of environmental events are alpha particles released by radioactive impurities, and more importantly, neutrons from cosmic rays [68]. On the other hand, intentional perturbation events are usually produced by optical sources such as flashlights or laser beams [51], which maliciously induce transient effects on secure circuits to provide confidential information for cryptanalysis methods or to activate hardware Trojans [17, 73]. The intentional events are thus essential actions of fault-based attacks that retrieve secret data from security applications.

Among the numerous design strategies for detection of transient faults caused by radiation or optical sources, Bulk Built-In Current Sensors (BBICS) [100, 101] offer a promising solution that is perfectly suitable for systems based on CMOS standard cells of commercial libraries [53]. BBICS combine the high detection efficiency of costly fault-tolerance schemes (e.g. duplication with comparison) with the low area and power overheads of less efficient mitigation techniques such as time redundancy approaches [82]. BBICS approach was experimentally validated in bulk CMOS 28 nm and 90 nm chips under the effects of laser sources [28, 135, 151], and designed, moreover, with transistors of carbon nanotubes [116].

In the last 10 years, several BBICS architectures composed of static memories have been proposed [41, 42, 102, 109, 110, 112, 121, 122, 125, 141, 142, 150]. In [125], the authors have compared them in terms of sensitivity for detecting transient faults. More recently, with the aim of reducing area and power overheads, Simionovski and Wirth devised a new class of BBICS constituted of dynamic memories [123, 124, 126]. Unlike previous works and the comparison

study in [125], in this chapter, which was the theme of our recent work [6], we discuss and compare both, static and dynamic, state-of-the-art BBICS architectures, analyzing their area offsets and detection sensitivities in typical and corner conditions. Furthermore, we introduce a new dynamic BBICS architecture with improved detection sensitivity and lower area penalty than its predecessors.

Section 4.1 classifies and describes the different state-of-the-art BBICS architectures and their basic principles. Section 4.2 presents our new dynamic BBICS architecture, and section 4.3 defines what we call as the sensitivity of a sensor or a memory element in detecting single transient faults. Finally, section 4.4 provides comparative results and analysis of BBICS architectures, and section 4.5 concludes this chapter highlighting the main BBICS features and perspectives.

4.1 Architectures of Built-In Current Sensors

Built-in current sensors (BICS) were initially proposed as a mechanism for detecting high increases in the current I_{DDQ} consumed by a CMOS circuit during its quiescent state (i.e. when the circuit is not switching). This type of mechanism enables the testing of CMOS circuits against permanent faults [12]. Further, BICS were also adapted for detecting transient faults – anomalous transient currents produced on the circuit by external perturbation sources [93] [22] [126] (Fig. 4.1). Firstly, BICS schemes for identifying transient faults in memory cells (bit flips) were devised [50, 85, 98, 133]. More recently, efforts were made for monitoring transient currents in combinational logic too [97]. All these techniques connect BICS circuits between the sources of the monitored transistors and the power rails (V_{DD} or GND), targeting on distinguishing anomalous currents from normal currents. Nevertheless, in today's technologies the amplitude of transient currents induced by radiation effects or fault attacks have the same order of magnitude than currents (source to drain or drain to source) normally generated by switching activities of logic circuits. Hence, schemes monitoring transistor sources are very limited for detecting just a restricted range of transient faults.

For overcoming this BICS problem, Neto et al. proposed in [100, 101] the first architectures of bulk built-in current sensors (BBICS). The major innovation of the BBICS is the connection of sensors between the bulks (i.e., body-ties of the target monitored transistors) and the power rails, rather than applying between the transistor sources and the power rails. Thanks to such a difference, BBICS are able to efficiently detect a wider range of transient faults than the classic BICS.

The BBICS-based strategy for the protection of a system is illustrated in Fig. 4.2. A pair of sensors is integrated to monitor pull-up and pull-down CMOS networks of the system blocks, hereinafter respectively PMOS-BBICS and NMOS-BBICS. Melo et al. have analyzed in [91, 92] the robustness of BBICS architectures to substrate noise. Wirth in [141, 142]

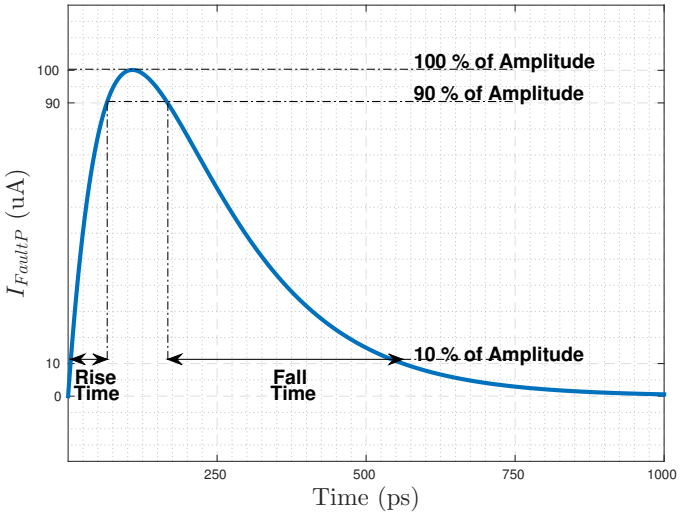


Fig. 4.1: Typical double-exponential profile of a transient fault, which is defined as a transient current generated on the circuit by a external perturbation such as radiation sources or laser beams.

has studied and verified the device-level operation of a BBICS by using TCAD (technology computer-aided design) simulations. In case of an anomalous current such as I_{FaultN} or I_{FaultP}, for instance, it will flow through the junction between the bulk and a reversely biased drain of the disturbed transistor (MOSFET "off"), and the sensors will be able to detect it by considering two phenomena:

1. In fault-free scenarios (i.e., $I_{FaultP} = 0$ and $I_{FaultN} = 0$), the bulk-to-drain (or drain-to-bulk) current is negligible even if the MOSFET is switching due to a new input stimuli;

2. During transient-fault scenarios, I_{FaultP} or I_{FaultN} is much higher than the leakage current flowing through the junction.

The sensitivity of a sensor to identify transient faults declines by increasing the number of transistors under monitoring. Hence, target systems have to be split into several blocks that contain a set of transistors monitorable by a sensor with a sufficient sensitivity in detecting a desired range of transient faults. Fig. 4.2 shows an example of system (chains of inverters) divided into two blocks monitored by two pairs of BBICS.

The range of detectable transient faults is adjustable by calibrating the size of some specific transistors of the sensors. Furthermore, BBICS are designed to latch a flag that indicates the detection of the abnormal currents within a defined range representing a risk of soft errors (i.e., bit flips of memory elements).

With the latch structure responsible for storing the fault indication flag, we classify BBICS architectures into static and dynamic. Static BBICS, which contain a static memory cell, are able to monitor transient faults independently of any periodic signal. In contrast, dynamic

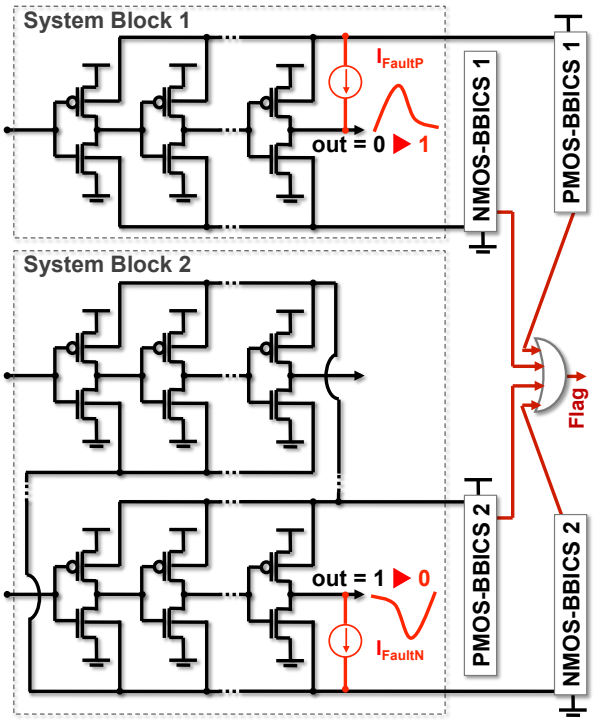


Fig. 4.2: Basic illustrations of BBICS monitoring two system blocks. I_{FaultP} and I_{FaultN} are current sources acting as external perturbations that produce abnormal current effects on the circuit defined as transient faults.

BBICS feature a dynamic memory, which requires by nature a periodic refresh signal to eliminate harmful leakage effects on its voltage output. In the following subsections, we summarize the state-of-the-art BBICS architectures in four types of static sensors and one type of dynamic sensors.

4.1.1 Single BBICS architectures

The sensor architecture illustrated in Fig. 4.3a is the simplest static BBICS in the literature. It counts only 9 transistors, 4 constitute two cross-coupled inverters, i.e., a latch used to register a flag in case of transient faults. This architecture presented in [125] combines concepts proposed in different works [102, 110, 112, 141, 142].

The principle of using a single BBICS circuit to check at the same time both pull-up and pull-down CMOS networks was suggested for the first time in [110, 112] with the aim of saving area. PMOS and NMOS bulk nodes of a single sensor like that in Fig. 4.3a are connected to the monitored body-ties (i.e., bulks of PMOS and NMOS transistors under monitoring) with the help of metal lines. The high ohmic transistors 5 and 7 (with large channel lengths) ensure appropriate voltage bias to the bulk during fault-free scenarios as well as preventing the complete attenuation of the anomalous transient currents in fault contexts. On the contrary, the low-threshold transistors 8 and 6 (with large diffusion widths) are sensing transistors ready to

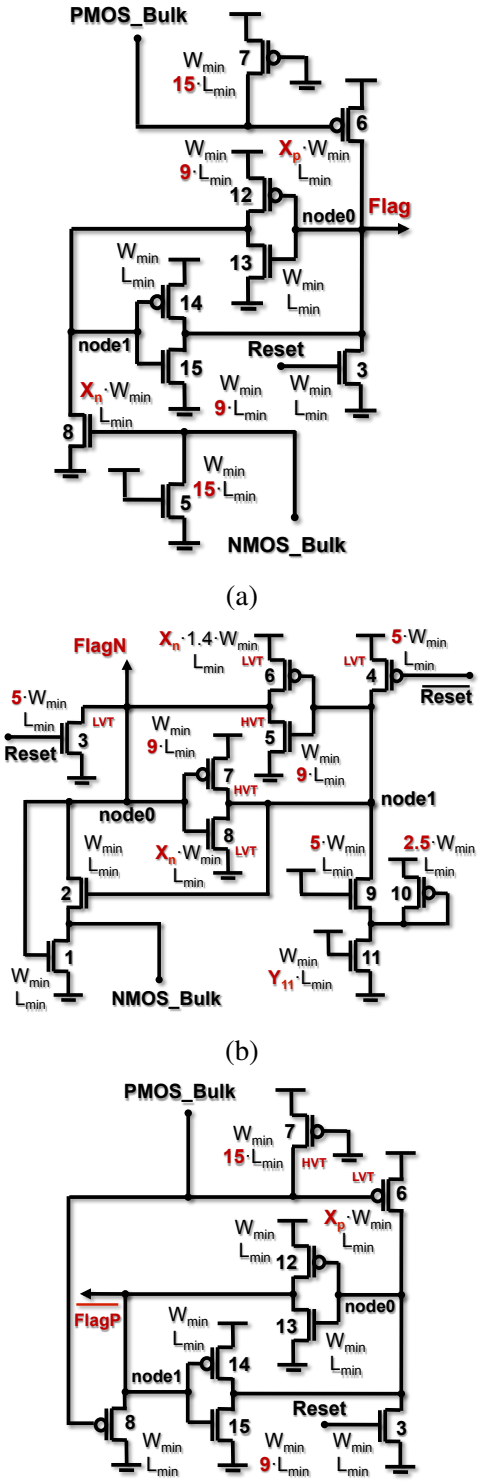


Fig. 4.3: State-of-the-art BBICS architectures: single BBICS [125] (a), NMOS-BBICS of Neto et al. [102] (b), PMOS-BBICS of Zhang et al. [150] (c). W_{min} represents the minimum diffusion width of the transistors, L_{min} is the minimum channel length, and X_n and X_p are design factors used for calibrating the sensitivity of the sensor in detecting transient faults.

quickly switch in case of transient faults, inverting the latch logic. This transistor sizing strategy stated in [102, 141, 142] improves the detection sensitivity of the sensor and makes the leakage power overhead negligible.

In order to enhance even more the sensor sensitivity in detecting transient faults, Dutertre et al. [41, 42] propose replacing standard transistors 5 and 7 by high-threshold voltage transistors (HVT), and transistors 6 and 8 by low-threshold voltage transistors (LVT). Dutertre et al. also highlighted in [42] the importance of using triple-well CMOS technology in networks of NMOS transistors monitored by BBICS. This strategy, which embeds NMOS transistors into P-type wells isolated from P-substrate by N-type well implants, increases the robustness of monitored circuits and considerably improves the sensitivity of the sensor in detecting transient faults in pull-down CMOS networks. As the classic N-type wells in PMOS transistors of pull-up networks, the P-type wells play in monitored NMOS transistors a role of isolation from P-substrate that efficiently helps BBICS in identifying transient faults in pull-down CMOS networks. LVT and HVT transistors as well as the triple-well feature are provided by most of modern commercial technologies.

Champeix et al. in [28] have tested a single BBICS architecture in a bulk CMOS 90 nm chip. Moreover, they have performed fault injection campaigns with a laser facility for validating the approach.

4.1.2 BBICS architectures of Neto et al.

Authors of the first versions of BBICS [100, 101] present in [102] an enhanced architecture formed by a pair of sensors: PMOS-BBICS and NMOS-BBICS. Fig. 4.3b details only the NMOS-BBICS circuit for the sake of simplicity. The illustration omits the PMOS-BBICS and the trimming transistors, which work to compensate process variability in transistors 5 and 7.

The sensor shown in Fig. 4.3b also consists of two cross-coupled inverters that create a latch for fault register. Furthermore, it has additional transistors 9, 10, and 11 acting to increase the sensitivity of the sensor in detecting transient faults. On the contrary, it is evidenced in [109] that the leakage power consumption is considerably grown by including these three transistors and using transistors 2 and 5 between NMOS_Bulk and gnd. As a compensation, a sleep-mode feature dedicated for BBICS is proposed in [109] to reduce the power consumption when the system is left on standby.

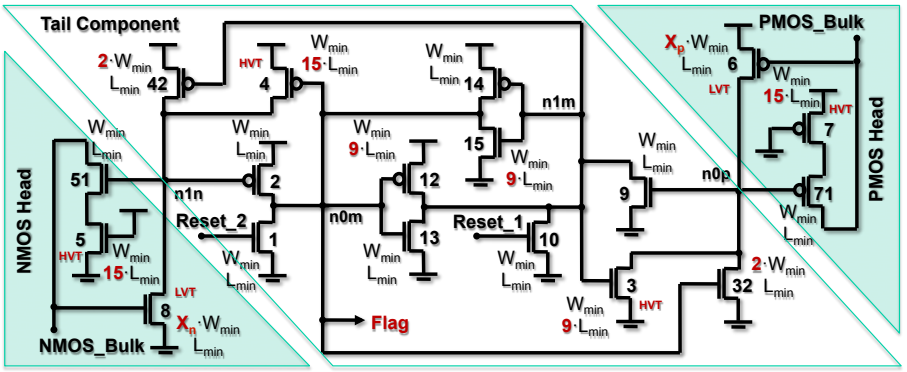
4.1.3 BBICS architectures of Zhang et al.

Zhang et al. [150] propose architectural improvements to BBICS of Neto et al. [102] with the intention of eliminating the leakage penalty. The architecture is also formed by a pair of PMOS-BBICS and NMOS-BBICS. Fig. 4.3c shows the PMOS-BBICS devised by Zhang et al. It is operationally similar to its predecessors, excepting by the presence of PMOS transistor 8. The

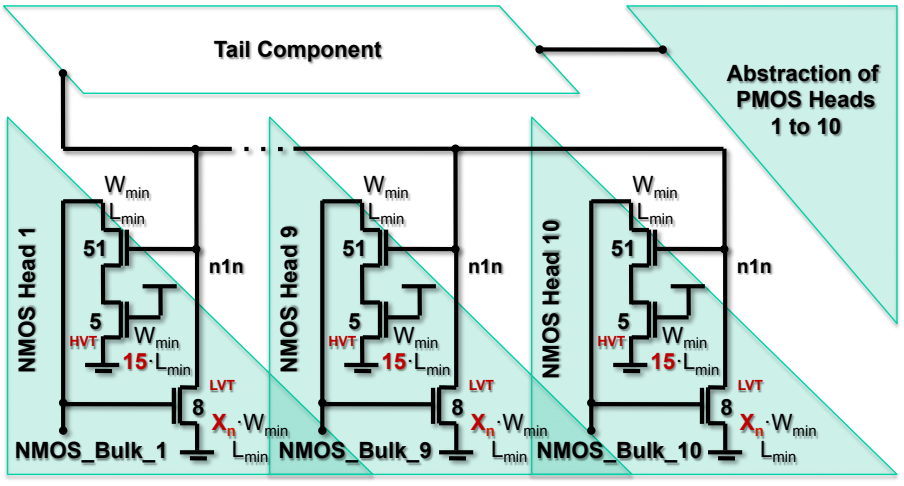
sensor transistors 6 and 7, which make the leakage overhead negligible, were preliminarily studied and suggested by Wirth [141, 142]. The circuit of the NMOS-BBICS of Zhang et al. is complementary to that illustrated in Fig. 4.3c for the PMOS-BBICS.

The architecture of Zhang et al. [150] has been improved in work [29] with the inclusion of CMOS amplifiers. The function of sensing the transient faults on the bulk is attributed to high-gain CMOS amplifiers, such as the previous works [12, 50, 85, 97, 98, 133] have proposed for monitoring and identifying faults on power rails. Even though an amplifier-based solution seems to be promising in terms of sensitivity in detecting transient faults, the sensor [29], built in a bulk 28 nm chip, was experimentally reported in [135] as sensitive to voltage and temperature variations.

Zhang et al. [151] also reported practical results of Fig. 4.3c sensor embedded on bulk CMOS 90 nm chip. The sensor was tested under the effects of laser-based injection sources.



(a)



(b)

Fig. 4.4: BBICS architecture [125] (a) using modular technique [121] (b). W_{min}, L_{min}, X_n, and X_p are defined in caption of Fig. 4.3.

4.1.4 Modular BBICS architectures

In [121, 122], an efficient modular technique was presented for reducing the area overhead introduced by BBICS architectures. The idea is to split the sensor into modules named as tails and heads. Fig. 4.4a details this technique applied on a BBICS architecture proposed in [125]. The sensor could be otherwise designed for monitoring the occurrence of transient faults in 10 pull-up and 10 pull-down CMOS networks, for instance; then the architecture will have 10 NMOS heads, 10 PMOS heads, and a single tail circuit shared by them, see Fig. 4.4b. This modular feature is also able to provide process and temperature robustness to the sensors thanks to the use of the several modules spread on the circuit under monitoring [121, 122].

In addition to take benefit from the modular technique, the circuits of Fig. 4.4a architecture [125] have been devised with attributes (operated by transistors 4 and 3) that facilitate the logic inversion of the latch (transistors 12, 13, 14, and 15). Consequently, it considerably improves the sensitivity of the sensor in detecting transient faults. Negligible power penalty is also reported due to the configurations of transistors 5 and 51 as well as 7 and 71, which ensure respectively the bias GND to the P-type wells and V_{DD} to the N-type wells.

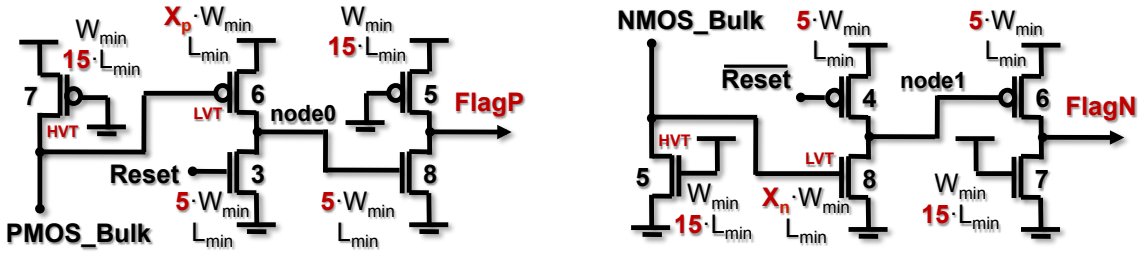
4.1.5 Dynamic BBICS architectures of Simionovski and Wirth

Simionovski and Wirth introduce in [126] the class of the dynamic BBICS architectures. Instead of the conventional latch of the previous static architectures, dynamic memory cells are used for smoothing the switching capacity of the memory node responsible for the fault register. With no feedback circuit wired to the memory node, the sensitivity of the sensor in detecting transient faults is increased and the transistor count of the sensor is reduced. Fig. 4.5 depicts the dynamic circuits featuring the detection of transient faults in pull-up and pull-down CMOS networks.

As any dynamic CMOS circuit, this first version of the dynamic BBICS [126] operates with the help of a reset signal. It periodically refreshes the sensor memory node that is not wired by a feedback circuit. The periodic reset is mandatory to remove accumulative leakage effects on the sensor output, and preventing consequent false alarms indications of fault. Results in [126] show a dynamic BBICS designed on bulk CMOS 130 nm technology is able to properly function by using a short reset pulse with a period of 50 ns. It leaves, therefore, appropriate time for digital systems deal with the fault indication provided by the sensor in case of transient faults. Simionovski and Wirth [123] have experimentally tested a bulk CMOS 130 nm chip with the sensor presented in Fig. 4.5.

The leakage current effects on the dynamic BBICS have been also studied in [124], and a solution for eliminating the periodic reset signal was presented. The strategy proposes to bias the reset transistors of the sensor for operating in the weak inversion region. Accordingly, a steady very low voltage offset is permanently applied on the place of the periodic reset voltage, ensuring, in fault-free scenarios, a stable operation of the dynamic memory nodes. This impor-

tant BBICS feature [124] baptized of self reset copes with the former insensitivity of dynamic sensors in detecting transient faults during the short but periodic phases of reset.



(a) (b)

Fig. 4.5: Dynamic BBICS architectures (a) and (b) of Simionovski and Wirth [126] for monitoring transient faults, respectively, in pull-up and pull-down CMOS networks. W_{min}, L_{min}, X_n, and X_p are defined in captions of Fig. 4.3.

4.2 New Dynamic BBICS Architecture

A new dynamic BBICS architecture is presented in Fig. 4.6. The innovations of the proposed BBICS considerably increment the sensor sensitivity in detecting transient faults at expense of negligible power overhead and with lower transistor count than previous architectures. The new features and differences with regard to the preceding dynamic BBICS [126] illustrated in Fig. 4.5 are:

1. In fault-free scenarios, the high ohmic transistors 7 and 5 detailed in Fig. 4.6 are responsible for biasing the bulks of the monitored PMOS and NMOS transistors, which are made more robust with the use of triple-well CMOS technology. Moreover, unlike other BBICS architectures, transistors 71 and 51 (respectively arranged in series with transistors 7 and 5) have the role of temporally including PMOS and NMOS bulk nodes in a floating state that facilitates the switch of the sensing transistors 6 and 8 during scenarios of transient faults. The detection sensitivity of the sensors are, therefore, improved at the cost of a periodic reset ($pReset$ in Fig. 4.6 and 4.7) on the gates of transistors 71 and 51, which operate to systematically ensure a suitable voltage bias of PMOS and NMOS bulks.

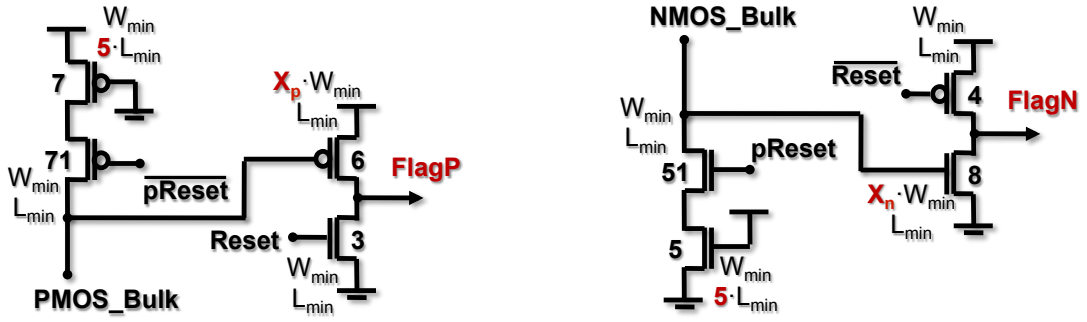
2. The large channel-length transistors 7 and 5 are isolated from the bulks through the minimum-size transistors 71 and 51. The number of parasitic elements connected directly to PMOS and NMOS bulk nodes is thus reduced, and the detection sensitivity of the sensor is enhanced;

3. With the two features described above, the proposed sensor does not need to use special HVT and LVT transistors for obtaining higher detection sensitivity than all previous BBICS architectures;

4. Thanks to the large channel-width transistors 6 and 8, the dynamic memory nodes $FlagP$ and $FlagN$ provide steady voltage signals during enough time to be dealt by other system blocks that have the responsibility of applying recovery actions when transient faults occur.

In addition to the periodic reset $pReset$, the same conventional reset applied on any BBICS architecture for initializing their memory nodes ($Reset$ in Fig. 4.6) is employed on the gates of transistors 3 and 4. This reset signal can be either periodic such as in dynamic architecture [126] or can feature the self-reset property [124] mentioned in previous section.

The operation mode of our dynamic BBICS architecture (Fig. 4.6) is illustrated in Fig. 4.7, which details simulation results of a chain of 10 inverters being monitored by a PMOS-BBICS (Fig. 4.6a) and NMOS-BBICS (Fig. 4.6b). The injected single transient fault, represented by the transient voltage glitch on the PMOS bulk node, is successfully detected by the PMOS-BBICS when the sensor's output $FlagP$ goes to V_{DD}. The pulse on the node $Reset$ is generated by other system block after the end of the procedure that processes the event of fault indication on the node $FlagP$.



(a) (b)

Fig. 4.6: New dynamic BBICS architectures (a) and (b) proposed in this chapter for detecting transient faults in pull-up and pull-down CMOS networks. The bulks of the PMOS and NMOS transistors under monitoring are biased, respectively, by the voltages on PMOS_Bulk and NMOS_Bulk nodes, rather than the voltages on the power rails V_{DD} and GND. W_{min}, L_{min}, X_n, and X_p are defined in captions of Fig. 4.3.

The proposed dynamic BBICS (Fig. 4.6) were designed and verified on a commercial bulk CMOS 65 nm technology. Two cells representing the PMOS-BBICS (Fig. 4.6a) and NMOS-BBICS (Fig. 4.6b) were developed in the same way as the technology standard cells were designed. The BBICS cells are applicable on the circuit under monitoring by replacing the biasing filler cells dedicated for making the body-ties of the standard cells [41]. The layout of the NMOS-BBICS cell is presented in Fig. 4.8, and its design factors X_n and X_p in Table 4.1, which is further explained in section 4.4.

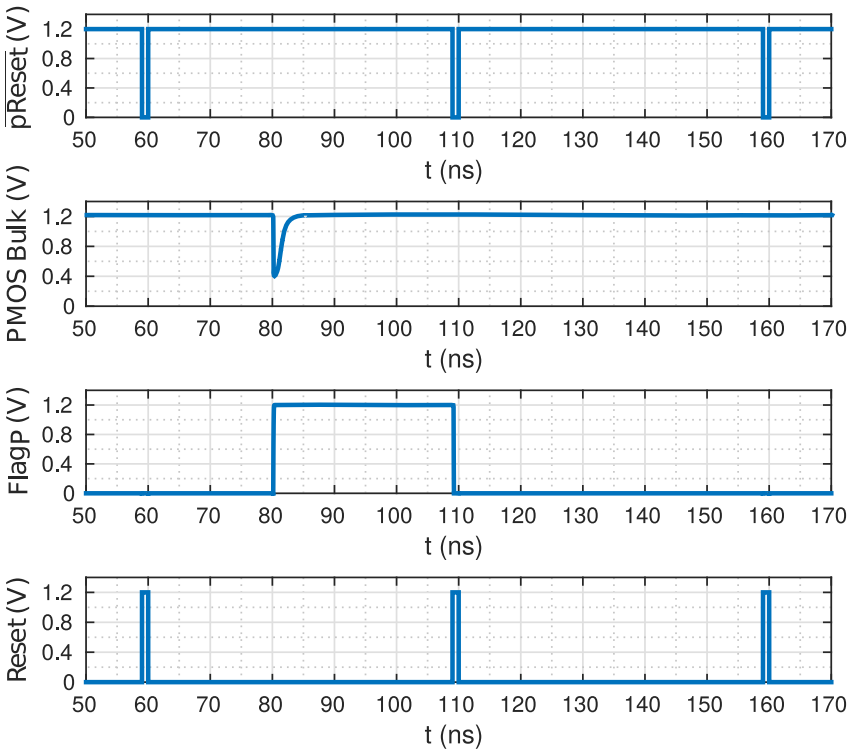


Fig. 4.7: Operation mode of the proposed dynamic PMOS-BBICS detecting the event of a single transient fault on the PMOS bulk node. The fault was injected on a chain of 10 inverters designed on CMOS 65 nm technology.

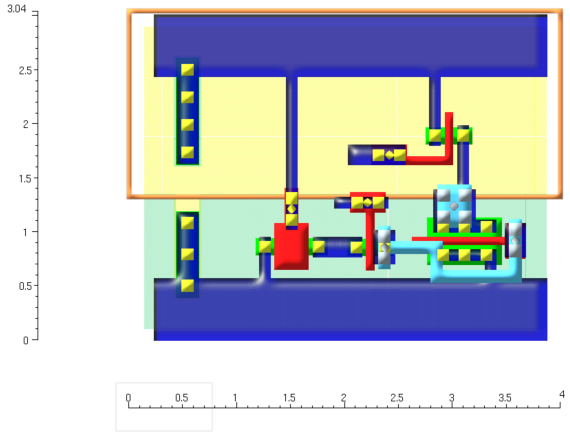


Fig. 4.8: Layout of the new dynamic NMOS-BBICS cell on CMOS 65 nm technology. The divisions of the axis are in μm. The area of the proposed cell is comparable to the sum of three technology NAND cells with minimum drive capabilities. The layout design of the PMOS-BBICS cell, which is not illustrated here, is complementary to this figure.

4.3 Sensitivity of a Flip-Flop in Detecting Transient Faults

Memory elements like flip-flops or latches are sensitive to transient faults that have the capability to reach them and provoke primary transient harmful effects known as: (1) soft errors, which are non-permanent logic inversions of memory elements; or (2) delay errors, i.e. remarkable non-permanent variations on the typical delays of memory elements due to setup time violations.

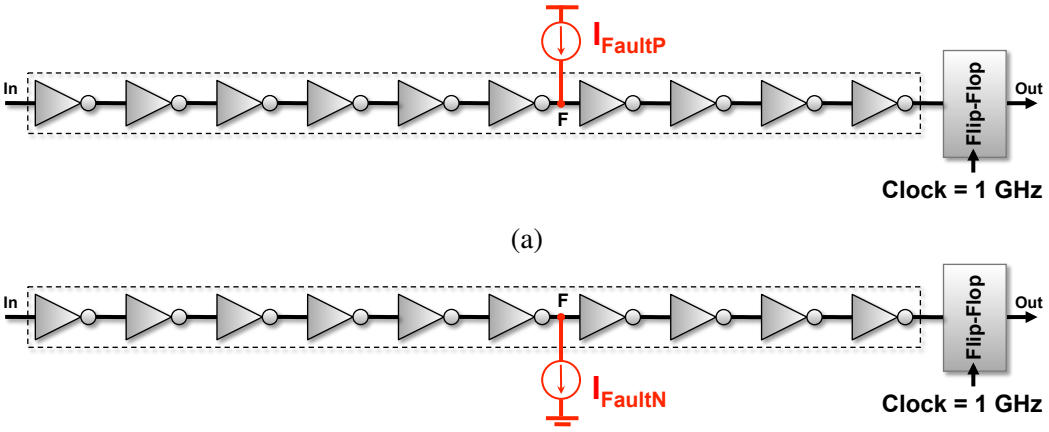
Soft or delay errors will be produced in the circuit depending on the charge of the transient fault – the integral of the current curve in Fig. 4.1. If an anomalous current has a profile (charge) able to overcome electrical, logical, and latching-window masking effects [68] on a circuit, single or multiple soft errors or delay errors will be generated in memory elements. The smallest anomalous current profile that provokes non-permanent errors (soft or delay errors) is defined in this work as the sensitivity of a memory element in detecting transient faults. The threshold at which the memory element becomes sensitive to transient faults is, therefore, the lower bound of the range of transient faults able to induce non-permanent errors in the memory element. The upper bound of this range would be the smallest transient fault that makes permanent errors and can definitely damage the circuit.

4.3.1 Experiments

The sensitivity in detecting transient faults of the flip-flops, as numerous and fundamental memory elements of integrated systems, is a significant reference to determine the smallest profiles of transient faults that need to be detected by schemes like BBICS. Hence, we have studied in this work the sensitivity of the smallest flip-flop cell of a commercial CMOS 65 nm technology. The goal is to evaluate and compare it with the sensitivities of different BBICS architectures. Fig. 4.9 illustrates the circuit used as reference in this study. Electrical-level simulations were initially performed with typical conditions, nominal V_{DD} (1.2 V), 25 ^{o}C, and standard threshold voltage (SVT) transistors. The technology smallest sized standard cells with parasitic elements were applied with the purpose of creating the circuit conditions that produce the smallest profiles of transient faults.

The influence of several different profiles of single transient faults was investigated on the reference circuits (Fig. 4.9) by using the classical transient-fault model for CMOS circuits [93] that is detailed in [22] and [126]. The faults were electrically simulated by injecting either a double exponential current source I_{FaultP} or I_{FaultN} (Fig. 4.1) on the technology most sensitive drain node, which is the drain with the lowest capacitance – i.e. node F(Fig. 4.9) between two inverter cells with the smallest dimensions in the technology standard cell library.

Different profiles of single transient faults were injected by adjusting different current amplitudes and fall times on the parameters of I_{FaultP} (or I_{FaultN}). The rise times were always set on the order of 5 ps to keep the typical shapes of transient faults: short rise time and longer fall



(a)

(b)

Fig. 4.9: Reference circuits of this study: chains of 10 inverters with a flip-flop. It is designed with the target technology's smallest standard cells with the aim of identifying the smallest profiles of transient faults (I_{FaultP} and I_{FaultN}) detectable by a flip-flop.

time [38, 46]. Several electrical-level simulations were thus done by sweeping the parameters of I_{FaultP} (or I_{FaultN}) up to find the smallest profiles of single transient faults that propagate through the inverters and provoke a soft error or a delay error in the flip-flop. In this study we consider as a delay error any flip-flop typical delay variation that is higher than 10%. In addition, as the shape of a transient fault is technology and event dependent, the sweep of the parameters of I_{FaultP} (or I_{FaultN}) has been limited to not create voltage amplitudes higher than 110 % of V_{DD}. This strategy prevents the injection of voltage peaks that could lead the circuit to permanent errors or out of the technology specifications.

4.3.2 Results and analysis

Fig. 4.10 shows the electrical-level simulation results of the circuits in Fig. 4.9. The vertical axis represents the minimum peak-to-peak voltage (on node F and normalized to V_{DD}) that is detectable by the flip-flop after I_{FaultP} (or I_{FaultN}) is injected with a fall time defined on the horizontal axis. For instance, if I_{FaultN} is applied on node F with a fall time of 200 ps (measured between 90% and 10% of the injected current amplitude), the resulting minimum detectable peak-to-peak voltage on node F is around 0.9 V (i.e. 75% of 1.2 V). The flip-flop will thus suffer a soft or delay error if a single transient fault with 200 ps of fall time produces a peak-to-peak voltage on the node F greater or equal to 0.9 V. Fig. 4.10 allows, therefore, identifying the range of single transient faults that reach and produce non-permanent errors in the flip-flop. Note that single transient faults making peak-to-peak voltages on the order of 57% of V_{DD} are still able to provoke soft or delay errors; however they require very long fall times to accomplish it (approximately 2200 ps).

Fig. 4.11 and Fig. 4.12 detail the smallest profiles of transient faults (I_{FaultP} and I_{FaultN}) that produce the peak-to-peak voltages of Fig. 4.10. Thereby, a single transient current injected

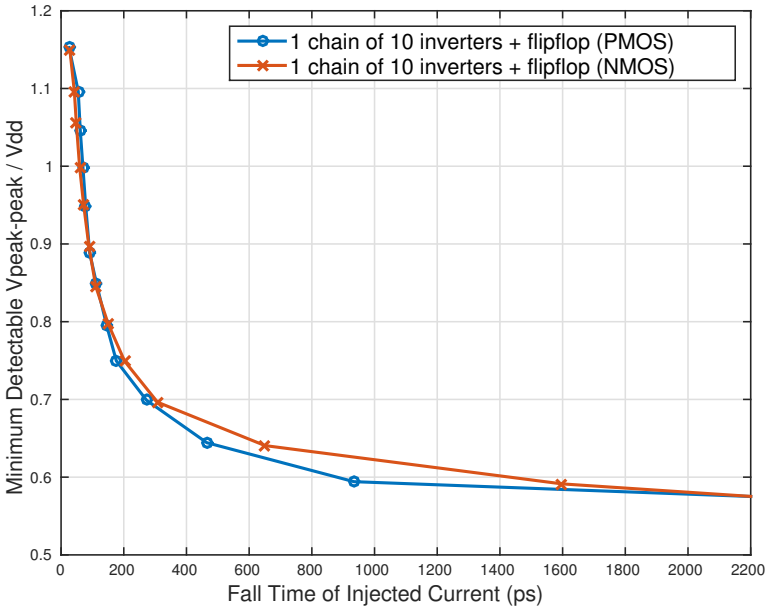


Fig. 4.10: Minimum peak-to-peak voltages (on node F and normalized to V_{DD}) that are detectable by a flip-flop (Fig. 4.9) after the injection of single transient faults (I_{FaultP} or I_{FaultN}) with fall times between 10 ps and 2200 ps; and a rise time on the order of 5 ps.

into node F with a fall time of 200 ps needs at least an amplitude of nearly 120 μA (NMOS case) or 160 μA (PMOS case) to provoke a soft or delay error in the flip-flop. In Fig. 4.12, the respective minimum detectable injected charges (critical charges), which correspond the areas of the injected current curves (Fig. 4.1), are presented on the order of 13 fC (NMOS case) and 17 fC (PMOS case).

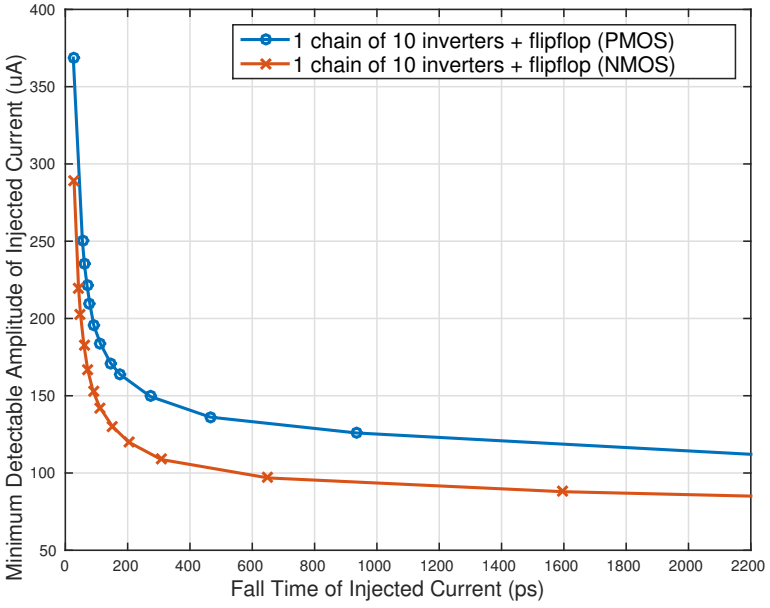


Fig. 4.11: Minimum current amplitudes (injected on node F) that are detectable by a flip-flop (Fig. 4.9). The related injected currents, in function of different fall times (horizontal axis), create the peak-to-peak voltages illustrated in Fig. 4.10.

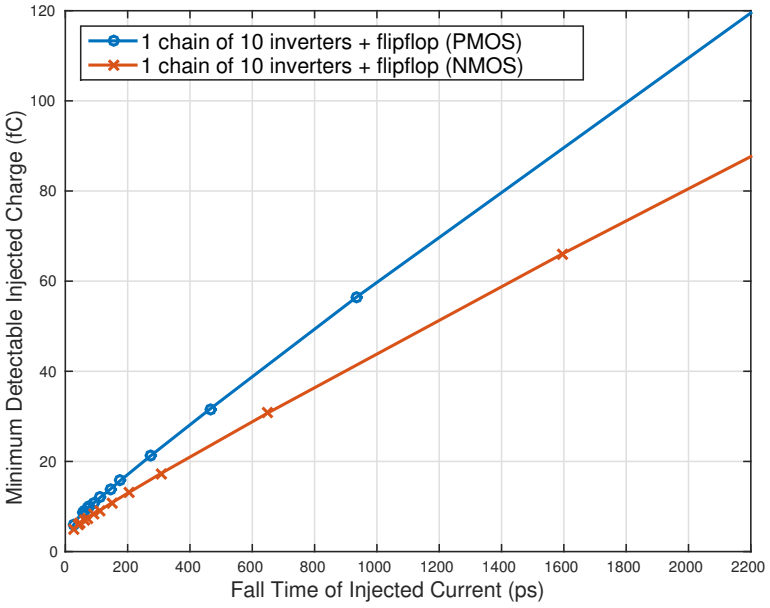


Fig. 4.12: Minimum charges (injected on node F) that are detectable by a flip-flop (Fig. 4.9). The related injected currents, in function of different fall times (horizontal axis), create the peak-to-peak voltages illustrated in Fig. 4.10.

4.4 Results and Analysis of BBICS Sensitivities in Detecting Transient Faults

This section analyzes previously discussed state-of-the-art BBICS architectures in terms of their sensitivities in detecting transient faults. In addition, we compare them with the dynamic BBICS proposed in this chapter.

4.4.1 Experiments for sizing BBICS architectures

All BBICS architectures were electrically simulated monitoring chains of 10 minimum-sized inverters under the same conditions of experiments described in section 4.3.

BBICS in Fig. 4.3a is denominated herein as "sbbics", and its improved version (using HVT transistors instead of SVT transistors 5 and 7, and LVT transistors replacing transistors 6 and 8) is defined as "shsbbics". Furthermore, the sensors in Fig. 4.3c, 4.3b, 4.4a, 4.4b, and 4.5 are named respectively "zbbics", "bbics", "t1hbbics", "t10hbbis", and "dbbics". The dynamic BBICS proposed in this chapter (Fig. 4.6) is labeled with "idbbics". Both NMOS-BBICS and PMOS-BBICS circuits of "zbbics", "bbics", "dbbics", and "idbbics" architectures were taken into account in the analysis of this section.

For the "bbics" architecture, we have set $Y_{11} = 9 \times L_{min}$ in PMOS-BBICS; $Y_{11} = 45 \times L_{min}$ in NMOS-BBICS [102]; and the trimming configuration calibrating the sensors with their best

sensitivities in detecting transient faults. The dynamic architectures "dbbics" and "idbbics" were both simulated with a periodic reset pulse of 500 ps repeated each 50 ns (Fig. 4.7). In addition, even though the original circuit propositions of "bbics" [102], "zbbics" [150, 151], and "dbbics" [123, 126] do not mention the use of LVT and HVT transistors for improving the sensor sensitivity [41, 42], we have used them in the simulated designs of this chapter in order to perform the full potential of such BBICS architectures. Original architectures "zbbics", "bbics", and "dbbics" were, therefore, also enhanced with LVT and HVT transistors in the same way of the other state-of-the-art BBICS analyzed in this chapter with the aim of making a fair comparison of their sensitivities in detecting transient faults. In the proposed "idbbics" architecture, nevertheless, LVT and HVT transistors are not required to calibrate competitive sensitivity, then only SVT transistors were used.

For each BBICS architecture under analysis, a similar transistor sizing strategy was applied, and the optimal values for the design factors X_n and X_p were obtained from several simulations under the effect of a typical single short transient fault [38, 46]. The single fault was injected into the node F (Fig. 4.9) with a rise time of 5 ps, a fall time of 50 ps, and a current amplitude that create a voltage amplitude below 100% of V_{DD} in each simulation scenario. Moreover, the simulations have swept X_n from 1 to 15, and X_p from 1 to 21. The minimum values of X_n and X_p for which the sensors have succeeded in detecting the lowest current amplitude were elected as the optimal. Table 4.1 summarizes the optimal values of the design factors X_n and X_p that were found.

Table 4.1: Taxonomy of BBICS Architectures Analyzed in this chapter: Total Number of Transistors (NMOS-BBICS + PMOS-BBICS Circuits), and Optimal Values for the Design Factors X_n and X_p

BBICS Architecture	Reference	Section	Figure	Class	Number of Transistors	X_n	X_p
sbbics	[110][112][28]	4.1.1	4.3a	Static	9	10	16.8
shsbbics	[110][112][41]	4.1.1	4.3a with HVT and SVT transistors	Static	9	11	16.8
bbics	[102]	4.1.2	4.3b	Static	$11 + 11 = 22$	12	12
zbbics	[150][151]	4.1.3	4.3c	Static	$8 + 8 = 16$	14	16.8
t1hbbics	[125]	4.1.4	4.4a	Static	$1 \cdot 3 + 1 \cdot 3 + 12 = 18$	10	16.8
t10hbbics	[125][121][122]	4.1.4	4.4b	Static	$10 \cdot 3 + 10 \cdot 3 + 12 = 72$	8	12.6
dbbics	[126][123]	4.1.5	4.5	Dynamic	$5 + 5 = 10$	13	14
idbbics	this chapter	4.2	4.6	Dynamic	$4 + 4 = 8$	5	5.6

4.4.2 Experiments for analyzing the sensitivities of BBICS architectures

Several electrical-level simulations were performed such as the experiments described in section 4.3 for a flip-flop; however the goal here was to identify the minimum injected currents that can be detected by a BBICS architecture monitoring the same chain of 10 inverters (Fig. 4.9).

The previously determined curves of minimum injected currents that are detectable by a flip-flop (Fig. 4.11) are also used in this section as references to evaluate the different target BBICS

architectures. These references allow verifying if a sensor is sufficiently sensitive to detect the smallest profiles of transient faults that cause soft or delay errors in the technology smallest flip-flop. Moreover, if a sensor is able to detect these reference profiles of injected currents; currents with larger profiles will be also detectable as they have more charge to overcome the thresholds of the sensor.

4.4.3 Comparative analysis of BBICS detection sensitivities

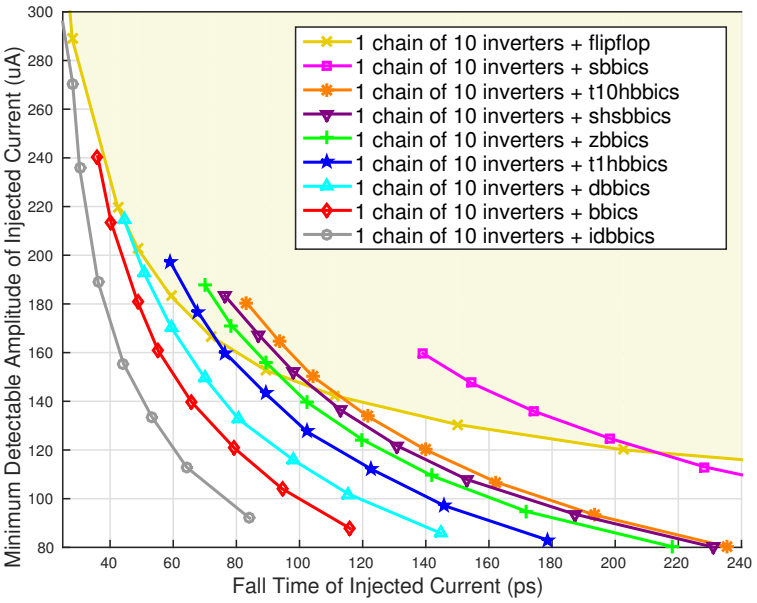
Fig. 4.13a and 4.13b present respectively the curves of minimum injected currents I_{FaultN} and I_{FaultP} that are detectable by the sensors protecting a chain of 10 inverters (Fig. 4.9). The graphics show thus the trends of each sensor in terms of their sensitivities in detecting transient faults. For instance, Fig. 4.13a highlights that a transient fault with 5 ps of rise time, 150 ps of fall time, and 130 μA of amplitude will cause a soft or delay error in the flip-flop; and it will be detected by all BBICS architectures except the "sbbics". Supposing however another scenario in which the fault has also 5 ps of rise time, 150 ps of fall time, and 160 μA of amplitude; even the architecture "sbbics" is able to detect it. Therefore, the lower is the curve of a sensor regarding the reference (flip-flop's curve), the higher is the sensor's sensitivity in detecting transient faults.

Comparing the curves of the different BBICS architectures in Fig. 4.13, we note the proposed dynamic sensor "idbbics" has the lowest curves regarding the flip-flop's references, thereby the highest sensitivity in detecting transient faults. The key difference of such a proposed architecture is indeed the propriety of periodically biasing bulks of monitored PMOS and NMOS transistors on triple-well CMOS technology. The sensor is even able to cover the detection of short and long transient faults that do not provoke soft and delay errors in the flip-flop ("idbbics" curves below flip-flop curves). This extra coverage of transient faults is extremely useful for either advancing preventive security alarm actions against fault injection-based attacks or, as discussed in next subsection, reducing the area overhead to the monitored system.

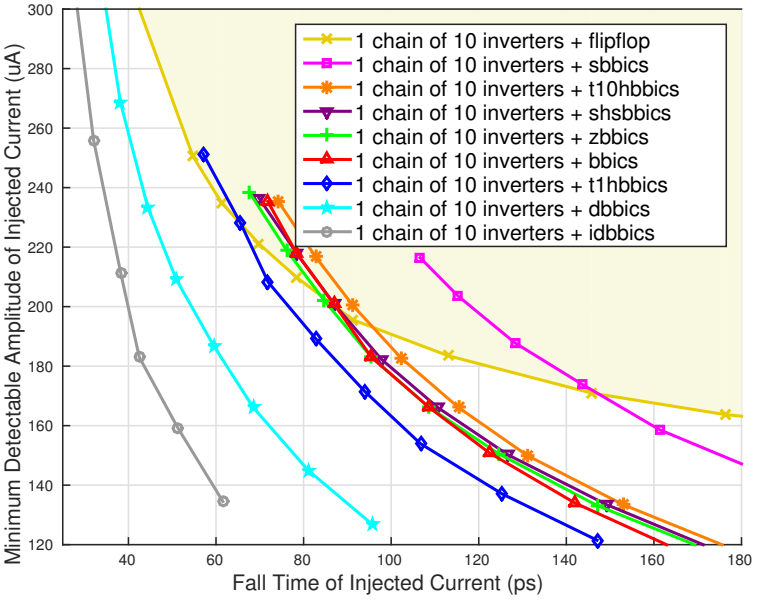
We also observe that only dynamic sensors ("dbbics" and "idbbics") are able to detect all short transient faults (with fall time below 70 ps). In this case, that produce soft or delay errors in the flip-flop (Fig 4.13 "dbbics" and "idbbics" curves below flip-flop curve). Otherwise, all BBICS architectures have the ability of detecting longer transient faults (with fall time above 220 ps).

On the other side, the static architecture "sbbics" reveals having the lowest detection sensitivity, although the application of HVT and LVT transistors [41] (instead of SVT transistors) consistently improves the sensor as Fig. 4.13 notices with the lower curves of the architecture "shsbbics". This result clearly illustrates the effectiveness of HVT and LVT transistors in enhancing the detection sensitivity of BBICS architectures.

Fig. 4.13a also highlights NMOS-BBICS of the static architecture "bbics" with a high detection sensitivity, confirming the contribution of transistors 9, 10, and 11 that create a voltage offset on node1 for reducing the switching efforts of the sensor latch. Nevertheless, the con-



(a)



(b)

Fig. 4.13: Minimum injected currents I_{FaultN} (a) and I_{FaultP} (b) that are detectable by a BBICS architecture monitoring a chain of 10 inverters. Flip-flop's curves from Fig. 4.11 were redrawn here to indicate reference thresholds in which a single transient fault provokes a soft or delay error in the flip-flop.

sequent power consumption of the monitored system, for example, a chain of 10 inverters, is increased by a factor of 80. It is substantially different from all other BBICS architectures that impose negligible power overhead thanks to the direct connection of the sensor high-ohmic and low-threshold transistors to the bulks of the monitored circuit.

Note additionally in Fig. 4.13, the application of the modular technique with multiple heads and a single tail slightly attenuates the detection sensitivity of the sensor (compare "t10hbbics"

and "t1hbbics" curves). This reduction is related to the higher number of transistors monitored by the architecture "t10hbbics", whose tail circuit is influenced by more parasitic elements. Equivalent reduction would happen on the detection sensitivity of the other state-of-the-art BBICS architectures if they were organized in the same way, i.e. with multiple heads and a single tail. The trick of splitting the sensor into multiples heads and a single tail is however useful like an additional parameter to make better trade-offs between detection sensitivity of the sensor and its resulting area overhead to the monitored system.

4.4.4 Influence of the monitored area size

The sensitivity of a BBICS in detecting transient faults is decreased in function of the number of monitored transistors. As more parasitic elements are included in the network monitored by the sensor, the amount of anomalous current (I_{FaultN} or I_{FaultP}) able to reach the sensor is reduced, lowering the sensor ability in identifying it. The results in Fig. 4.14 demonstrate this phenomenon in the cases of only one PMOS-BBICS architecture monitors either 1, 4, or 6 chains. The dynamic architectures ("dbbics" and "idbbics") present similar detection sensitivities when 4 chains of 10 inverters are monitored, however they are lower whether compared with the case of 1 chain of 10 inverters. Besides, the curves show that the detection sensitivity of the static architecture "zbbics" is much more reduced with the number of monitored transistors than its dynamic counterparts. Finally, note that different from all other BBICS architectures, the sensor "idbbics" proposed in this chapter has still design space to improve the results from Fig. 4.14 by increasing the design factors X_n and X_p as well as by using LVT and HVT transistors.

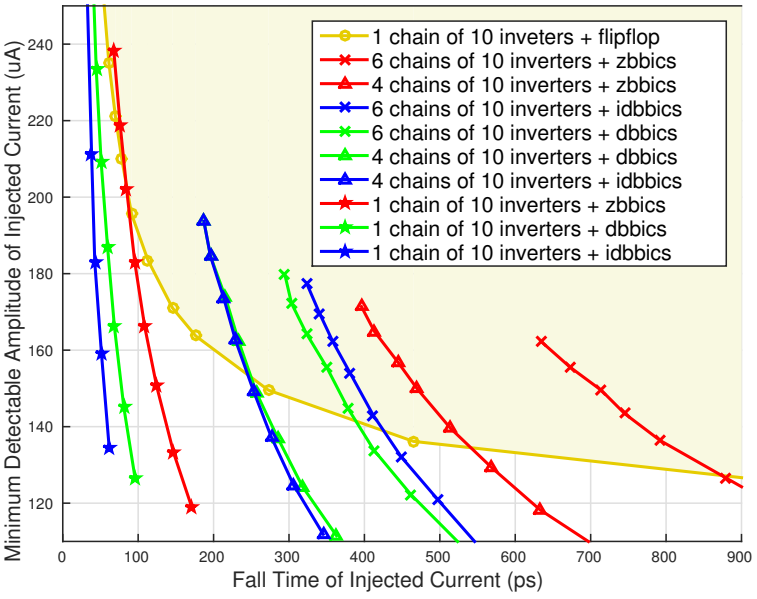


Fig. 4.14: Minimum injected currents I_{FaultP} that are detectable by a PMOS-BBICS architecture monitoring either 1, 4, or 6 chains of 10 inverters.

4.4.5 Estimation of the area overhead imposed by BBICS architectures

Fig. 4.15 estimates the area overhead imposed by each BBICS architecture based on the diffusion areas of the transistors ($W \times L$), which were calculated with help of the design factors discussed in previous subsection 4.4.1. The architecture "idbbics" proposed in this chapter imposes the lowest area overheads on the monitored systems (chains of 10 inverters) thanks to its smaller design factors and the lower number of transistors. For instance, if one PMOS-BBICS and one NMOS-BBICS of the sensor "idbbics" are applied to monitor 6 chains of 10 inverters, the consequent area overhead will be around 12%, while other BBICS architectures will lead to values higher than 36%.

The area overhead can be further reduced, at expense of lowing the sensor detection sensitivity, whether more transistors are included in the network monitored by one sensor (see previous subsection). This strategy would be suitable for applications requiring lower detection sensitivity or with a known range of transient faults to be detected. On the other side, if a single sensor (or a pair of NMOS-BBICS and PMOS-BBICS) is protecting a system block with a size on the order of a chain of 10 inverters, the area overhead might be prohibitive whether all system has to be monitored (see Fig. 4.15); however the detection sensitivity would be much higher. Alternatively, if only the most sensitive parts of the systems are selected to be monitored by BBICS circuits, the overall area overhead can still be significantly reduced.

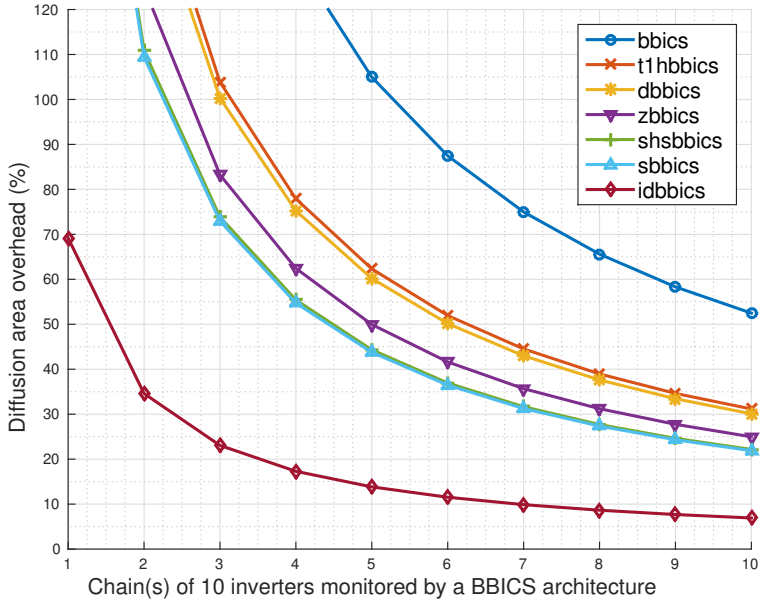


Fig. 4.15: Area overhead included by a BBICS architecture (a single sensor or one PMOS-BBICS and one NMOS-BBICS) that monitors a system with X chain(s) of 10 inverters (X between 1 and 10).

4.4.6 Corner analysis of BBICS architectures

The same electrical-level simulation experiments described in previous section 4.3 were applied for analyzing the BBICS architectures under process and temperature variations.

Normalized results of the minimum injected charges (integral of I_{FaultP} or I_{FaultN}) detectable by the BBICS architectures are presented in Table 4.2 for the following corner conditions: FF 25 ^{o}C; SS 25 ^{o}C; TT 25 ^{o}C; TT 75 ^{o}C; and TT -40 ^{o}C. The charges are normalized to the minimum injected charges able to provoke soft or delay errors in the flip-flop. All normalized charges in the table correspond to the smallest profiles of single transient faults (with 5 ps of rise time) that create a voltage on the order of 80% of V_{DD} (0.96 V) on the node F of a chain of 10 inverters (Fig. 4.9) monitored by a BBICS architecture.

From the table, excepting the architecture "bbics", which requires another on-the-fly trimming bit configuration for compensating the variations in FF and SS corners, all other BBICS architecture are able to operate under process and temperature variations. Nevertheless, depending on the corner condition, the detection sensitivities of the sensors are reduced. The proposed architecture "idbbics" are in all corners either much more sensitive to detect transient faults than the flip-flop or very close to it. This result gives an important margin to reduce the area overhead as discussed in previous subsections 4.4.4 and 4.4.5.

Table 4.2: Normalized Corner Results: Minimum Injected Charges that Are Detectable by the BBICS Architectures When I_{FaultP} (PMOS Case) or I_{FaultN} (NMOS Case) Induces a Voltage on the Order of 80% of V_{DD} on the Node F (Fig. 4.9).

BBICS Architecture	**Normalized Minimum Detectable Injected Charge**									
	PMOS Case					**NMOS Case**				
	FF	**SS**	**TT**	**TT**	**TT**	**FF**	**SS**	**TT**	**TT**	**TT**
	25 ^{o}C	**25 ^{o}C**	**25 ^{o}C**	**75 ^{o}C**	**-40 ^{o}C**	**25 ^{o}C**	**25 ^{o}C**	**25 ^{o}C**	**75 ^{o}C**	**-40 ^{o}C**
flipflop	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
bbics	NO	NO	1.08	0.81	0.71	NO	1.75	0.82	0.67	0.26
sbbics	1.20	1.62	1.40	1.25	1.68	1.47	2.21	1.84	1.64	2.19
shsbbics	0.85	1.30	1.08	0.99	1.26	0.99	1.53	1.27	1.14	1.45
t1hbbics	0.80	1.14	0.97	0.89	1.09	0.85	1.31	1.08	0.98	1.22
t10hbbics	0.90	1.32	1.12	1.04	1.27	1.07	1.58	1.33	1.24	1.45
zbbics	0.85	1.26	1.06	0.97	1.25	0.93	1.45	1.19	1.08	1.39
dbbics	0.35	0.42	0.70	0.60	0.67	0.60	0.59	0.92	1.23	1.31
idbbics	0.40	0.45	0.54	0.57	0.66	0.26	0.27	0.68	1.13	1.19

4.5 Conclusions

This chapter reviews the different types of static and dynamic BBICS architectures by analyzing their sensitivities in detecting single transient faults. Moreover, a new dynamic BBICS archi-

tecture is presented, offering considerable advantages in terms of detection sensitivity and area overhead.

The BBICS application on integrated system designs brings several other complementary benefits. The integration in commercial IC design flows is feasible by simply replacing standard biasing filler cells or well-taps [53]. The reuse of these cell areas, besides making the sensors more sensitive in detecting transient faults, reduces the inherent costs associated with any type of robustness technique. BBICS-based recomputing techniques are applicable for recovering processors from the effects of transient faults [77, 82, 108, 111]. Furthermore, unlike most existing techniques, long-duration and multiple transient faults are also detectable by BBICS [82, 134]. All these features represent important contents for the design of more robust integrated systems in modern technologies.

In the advent of advanced fabrication processes, such as FD-SOI technology for which different body voltages are allowed for managing system performance and power consumption, the design space becomes even wider with the double function of BBICS cells in locally biasing system blocks as well as detecting the occurrence of transient faults.

Chapter 5

Reuse of Bulk Built-in Sensors for Detection of Trojans

This chapter proposes a novel HT-detection method that implies the creation of a new category in the previously mentioned taxonomy of side-channel combined with functional analysis-based techniques. It was the theme of our publications [3, 4].

The proposed method indirectly analyzes HT-induced variations on the electrical impedances of DUTT (design under Trojan test) subcircuits by injecting a train of current pulses into body terminals of their MOSFETs. More precisely, the analyzed side-channel is indeed digital signatures, related to the impedance of the subcircuit substrate and provided by a preexisting BBICS connected to all body terminals of the subcircuit transistors. The similar sensor presented in the section 4 (BBICS), which is originally designed to ensure appropriate bias to the body terminals by replacing the biasing filler cells (well-taps) of the system [53], was only used until now as online-testing devices for detecting radiation- or laser-induced transient currents that may provoke soft errors in memory elements [28, 82, 123].

However, we reuse it here as an offline-testing mechanism, without degrading its run-time feature of detecting transient currents. By applying a short train of current pulses (with different amplitudes) into a DUTT body terminals, each BBICS will detect the pulses in function of their amplitudes and the impedance of the subcircuit substrate, delivering a train of voltage pulses that represents a digital signature of the DUTT. In case of a Trojan-infected subcircuit, the impedance of the subcircuit substrate will be modified with the presence of any HT, altering, consequently, the digital signature of the DUTT, and making the HT detection likely by comparing it with a Trojan-free signature. However, sampling a set of digital signatures by subcircuit is necessary for statistically distinguishing HT-induced variations from process variations.

Unlike most existing side-channel analysis-based techniques, the proposed method requires no switching activity in data paths and no analog measurements. Hence, input test-vector gen-

eration and measurement noise are not issues, and the detection of stealth and tiny HT that have only turned-off transistors and negligible leakage currents are more likely.

Next sections detail the three main innovative contributions of this work: (1) current pulses are injected into body terminals of system subcircuits; (2) built-in current sensors are connected to body terminals for identifying or not the injected currents, providing digital signatures of the subcircuit substrates; (3) resulting digital signatures allow indirect analysis of the impedance of subcircuit substrate, which is modified with the presence of HT, opening a new category of side-channel analysis-based techniques (see Fig. 5.1).

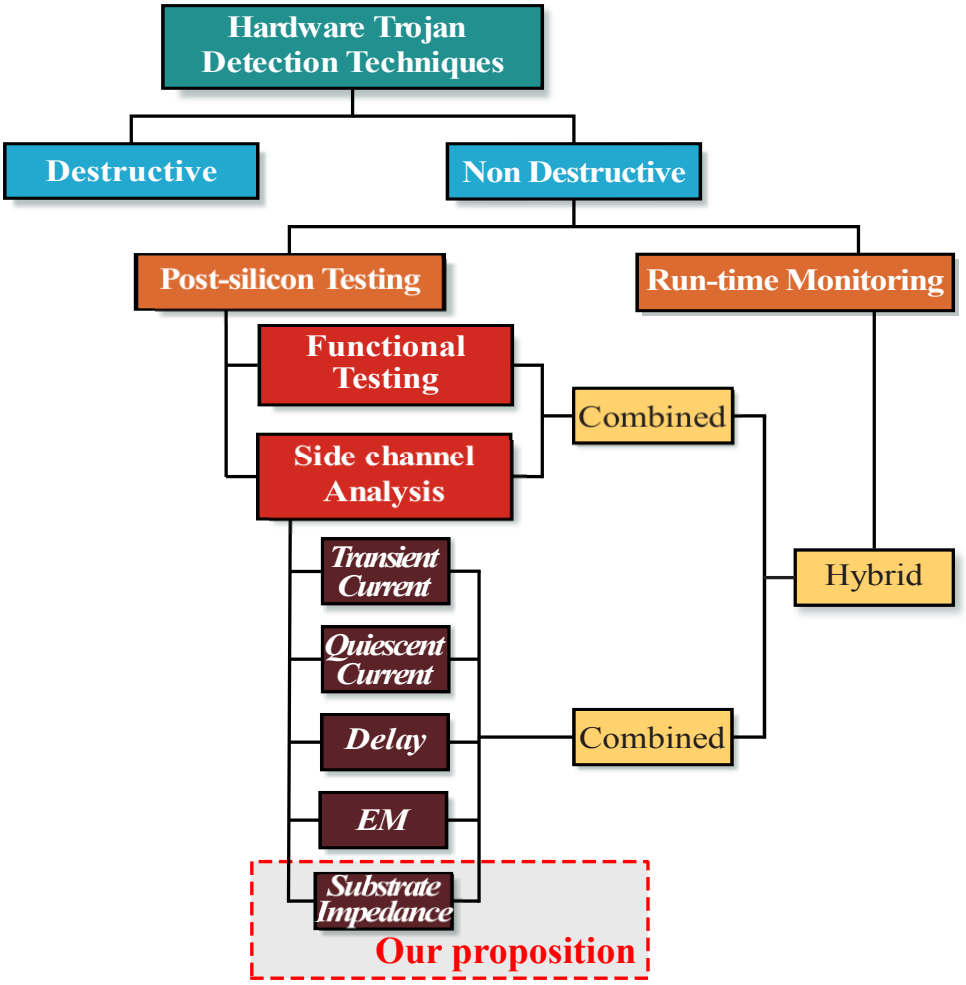
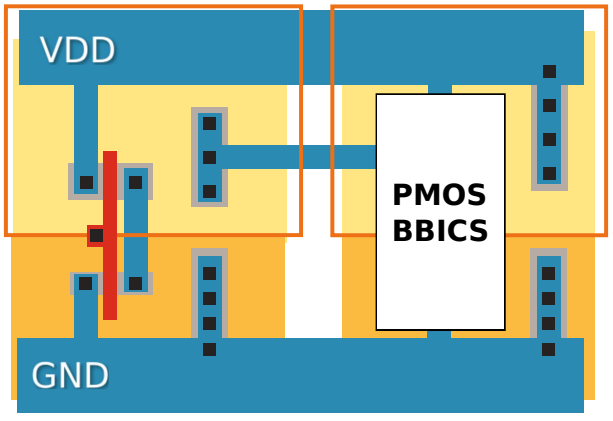


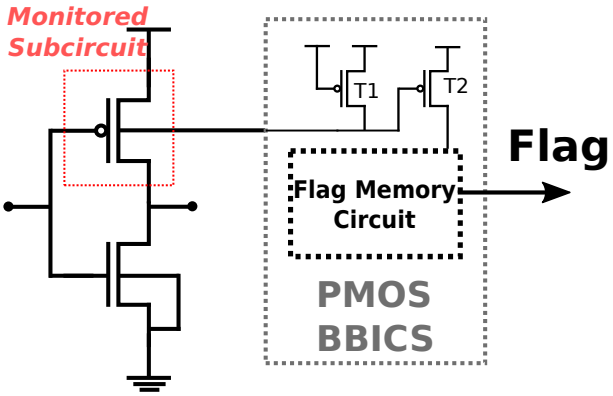
Fig. 5.1: Classification of hardware Trojan detection techniques [48, 64, 69, 94, 130, 132] with a new category of side-channel analysis: the substrate impedance.

5.1 Built-in Current Sensors for Generating Signatures of Subcircuit Substrate

As described in section 4, bulk built-in current sensors (Bulk-BICS or BBICS) connected to MOSFET body terminals explore the presence of an abnormal current peak flowing from the bulk (body) to the drain (or vice versa) of a disturbed transistor if a transient current is induced by radiation or laser sources [28, 82, 123]. Whenever this current is detected, the sensor indicates it by setting its output flag. BBICS are implemented in the original layout by replacing the regular biasing fillers (well-taps) used to bias the circuit body [53]. Therefore, the sensor becomes responsible for biasing and monitoring the target circuit body terminal. Fig. 5.2 synthesizes the sensor implementation showing the layout (Fig. 5.2a) of a single inverter being monitored by a PMOS sensor. Note that the PMOS body terminal of the targeted subcircuit is attached to the PMOS sensor instead of being connected directly to the V_{DD} line. In Fig. 5.2b, its schematic view is depicted presenting the basic BBICS architecture composed by a biasing and a sensitive transistor (T1 and T2). The occurrence of a transient current can lead T2 to its active mode, generating the output flag. The flag memory circuit is responsible for holding and reseting the output.



(a)



(b)

Fig. 5.2: Layout of an inverter monitored by a PMOS BBICS (a) and its schematic view (b).

For monitoring both pull-up and pull-down CMOS networks, the BBICS architecture must be designed in CMOS triple-well technology [42] with its body terminal attached to the NMOS sensor. As the scope of this chapter is not precisely detecting transient currents, the NMOS BBICS is omitted from our study in order to simplify the analysis; however its operation is similar to the PMOS. PMOS BBICS will be applied in this chapter to detect gate- and transistor-level Trojans as well as layout-level Trojans modifying PMOS networks. However, if NMOS BBICS is also applied, even layout-level Trojans in NMOS network are able to be detected.

The basic BBICS operation is presented in Fig. 5.3 if a sequence of current pulses are applied in its terminals. It causes a disturbance in the PMOS body terminals voltage and BBICS can be able to detect it depending on its sensitivity in detecting transient currents [92, 125]. Thus, BBICS generates a flag indicating the incoming current if its amplitude exceeds the BBICS detection threshold.

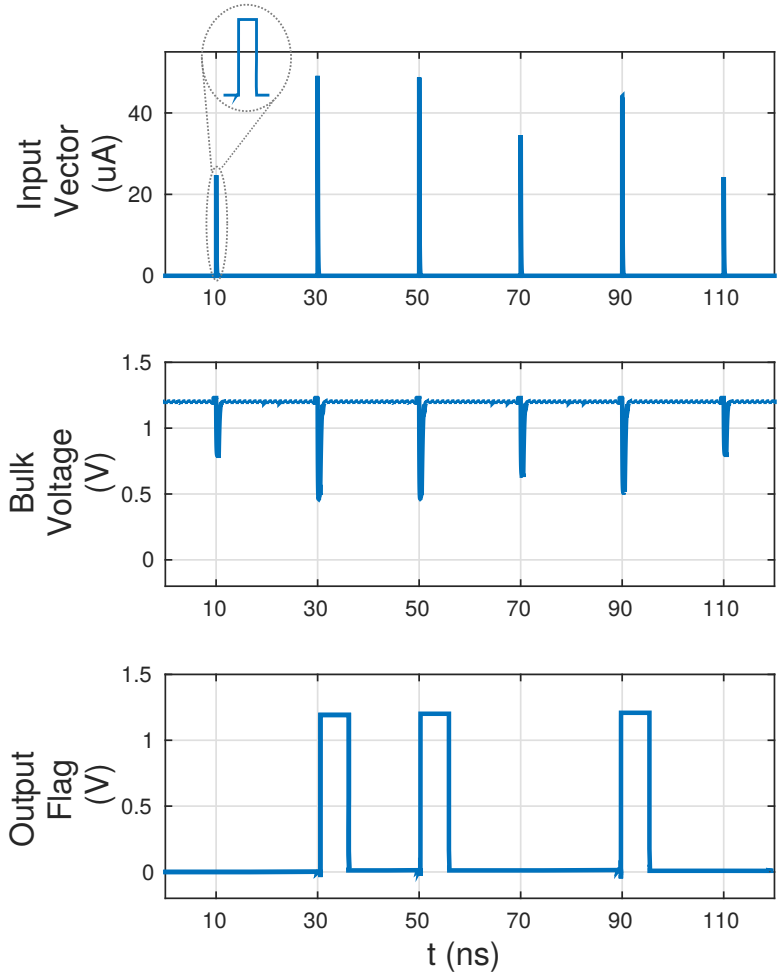


Fig. 5.3: Representation of current injections in the PMOS bulk, its impact in the PMOS bulk voltage and the output flag generated by the BBICS.

As discussed in chapter 4, the sensitivity in detecting transient currents for a given BBICS architecture essentially depends on two conditions: the transient current profile and the monitored circuit. Indeed, considering the same profile of transient current, the lower the amount of transistors under monitoring by the sensor, the better the BBICS sensitivity. As a circuit in today's technology has about billions of transistors, as a way to improve the BBICS sensitivity, the designer can split the circuit into smaller subcircuits. Each subcircuit must have a BBICS monitoring it. Therefore, the designer can conveniently choose the amount of transistors that the BBICS will monitor in each subcircuit, depending on the desired sensitivity.

5.2 Proposed HT Detection Method

The proposed detection method relies on the assumption that the addition or modification of instances from the target circuit leads to physical alterations in the original substrate impedance. Moreover, it is even possible to enhance the HT effects if the number of transistors placed in this substrate is reduced, facilitating its detection. To this end, the substrate is partitioned into several subcircuits composed of a group of transistors sharing the same subcircuit PMOS body terminal. Therefore, the method proposes monitoring these body terminals with BBICS. By injecting a current pulse in the body terminal, the BBICS allow tracking the induced voltage peak. If a HT is inserted in the same substrate, the voltage peak is altered and the BBICS may no longer to detect it. Thereby, the response of the BBICS for a subcircuit to a given set of different current pulses, also named signature, may indicate the presence of a HT in the original layout. To classify a population of DUTTs according to the HT infection, a statistical comparison between a certified Trojan-free IC and a DUTT signatures can be performed.

5.2.1 Injection of Current Pulses into MOSFET Body Terminals

In order to inject a current in the body terminal of the system subcircuits, three possible alternatives are highlighted: internal or external direct current injection and laser attack to induce current. The direct insertion of a current peak in the body terminal by an external source gives total controllability of the current profile and is simply integrated to the testing phase. Despite featuring such properties, it requires an extra analog pin in the IC, which could be an issue, depending on the design constraints. On the other hand, an internal source can be implemented to produce current pulses with controllable amplitude set by a digital input vector. Although the addition of a new analog pin is not needed, the range of possible amplitudes is limited by the size of the input vector as well as other waveform parameters. If the system is divided into several subcircuits, such approaches require at least an additional transistor to replicate the incoming current to each subcircuit, resulting in an extra area overhead. To avoid this area issue, a laser can induce a current in the body terminal. Even though this approach does not require extra

on-chip circuitry, it presents some inconveniences such as delay, cost and complex integration with the regular testing phase.

Designers can choose conveniently the current injection strategy that better fits according to their needs. Furthermore, the current amplitude must be set in such a way to avoid latch-up. Some other waveform parameters such as the pulse width can also be exploited to evaluate the BBICS sensitivity. In the following analysis, the method is evaluated with an external current source generating pulses with different amplitudes. Fig. 5.4 shows the current injection set-up used in the subsequent analysis. In this topology, the current generated by the external source is replicated to each body terminal of the subcircuits. Nevertheless, the same analysis can also be applied for other current injection techniques.

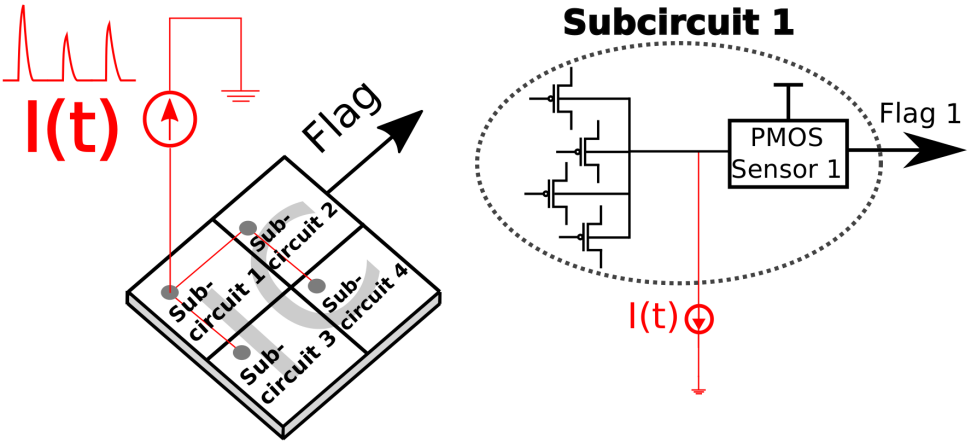


Fig. 5.4: Current insertion topology and its schematic view: an external current source able to insert subsequent peaks in the PMOS body terminal.

5.2.2 Monitoring of Current Sensors Built in System Subcicuits

To track the whole IC substrate, each subcircuit must have its body terminal attached to its own BBICS. Initially, the BBICS is designed to detect the current pulses generated by the source presented in Fig. 5.4. Basically, the detection is achieved whether the injected current (e.g. its amplitude) is larger than the sensor original threshold. Therefore, if a set of subsequent current pulses with different amplitudes are injected in the body terminal, only the peaks that exceed sensor threshold are detected. Fig. 5.3 shows an example of a train of current pulses that can be employed. In this case, the BBICS is able to detect only the pulses inserted at 30, 50 and 90 ns, indicating it by pulses in its output flag. Assuming a given sequence of numerous current pulses with different amplitudes, the generated BBICS output gives the digital signature of the design. As the BBICS sensitivity also depends on the target circuit characteristics, the digital signature is unique for each design.

The area overhead caused by the BBICS results from the ratio between the sensor and the target circuit area. The designer can set a target overhead area to define the number of sensors to be inserted in the design and consequently, the number of subcircuits. For instance, considering a built-in sensor architecture that occupies an area approximately equivalent to 3 minimum-sized standard cell inverters, the designer can define a target overhead of 10% and place each BBICS to monitor 30 inverters from the original design. However, as distinct gates have different areas, the BBICS can also be placed according to the number of the transistors composing a subcircuit. As the BBICS sensitivity is improved in smaller circuits, the number of transistors monitored by each BBICS must be chosen in order to ensure that the effects of process variation (PV) are less significant than HT implementation. Therefore, the BBICS minimum area overhead is limited by its sensitivity and PV. In any case, as the method proposition is reusing these built-in BBICS to detect Trojans, a system already covered by this on-line testing technique to monitor transient currents does not require overhead area dedicated for HT detection.

5.2.3 Compilation of Signatures Collected from Subcircuit Substrate

As the BBICS provides signatures for each design, the compilation of the obtained signatures is needed in order to analyze a possible alteration in its original value, indicating the presence of a HT. For instance, if the BBICS presents a certain amount of detections in a genuine design, it must provide the same result for a Trojan-free DUTT, varying according to PV. As a mean to evaluate the generated signature, the number of detections for a given train of current pulses can be analyzed. Therefore, the sensor efficiency is defined as the ratio between the number of achieved detections and the total number of pulse injections as show in (5.1). If the obtained signature is altered, the sensor efficiency is also modified and thus, a HT can de detected.

$$Sensor\ Efficiency = \frac{\#of\ Detections}{\#of\ Injections} \times 100\% \tag{5.1}$$

5.2.4 Statistical Analysis for Detecting HTs in System Subcircuits

The analysis of the sensor efficiency can be extended for a more realistic case, in which PV is considered. For a given train of current pulses, the PV causes a deviation in the expected sensor efficiency. For example, the graph in Fig. 5.5 depicts the histogram of the sensor efficiency considering a set of Trojan-free circuits under PV. Moreover, a curve surrounding the obtained histogram is presented to provide a possible parameterization for the obtained results, which represents the Trojan-free design profile. If a HT is placed in the original substrate, the alteration in the original sensor efficiency leads to a new profile curve (Trojan-infected DUTT profile), as shown in Fig. 5.5. The statistical analysis of the obtained profiles may indicate that, effectively, the DUTTs are Trojan-infected.

For classifying whether the DUTTs belong to the Trojan-free class, it is necessary to verify

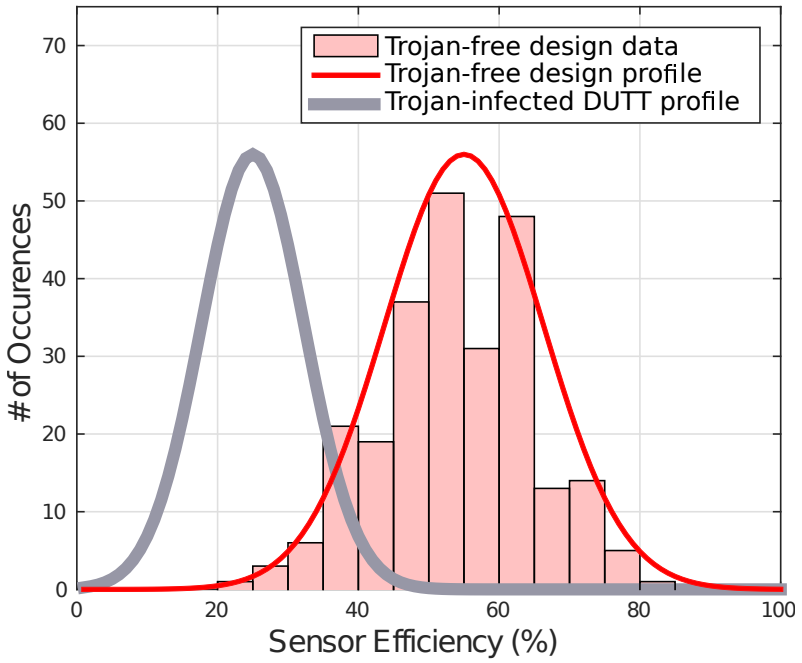


Fig. 5.5: Histogram of the sensor efficiency for Trojan-free design surrounded by its profile curve and another one of a Trojan-infected DUTT.

if the two obtained set of data came from the same distribution. Thus, two hypotheses are considered. H0: DUTTs belong to the Trojan-free class; and H1: DUTTs do not belong to the Trojan-free class. A hypothesis test is chosen in order to classify the DUTT according to its class. In this analysis, the Kolmogorov-Smirnov (KS) test [90] is used since it is non-parametric and can widely be found in common numerical computing libraries and environments. The p-value calculated by the KS-test indicates the possibility of belonging to H0. In order words, the closer p-value is to zero, the stronger the hypothesis of DUTTs being Trojan-infected.

5.3 Simulation Results and Analysis

5.3.1 Description of Simulation Experiments

The effectiveness of the proposed method is evaluated by simulating some DUTTs from benchmarks ISCAS'85, ITC'99, and chains of inverters, all monitored by one or more built-in sensors. The DUTTs were designed with CMOS 65 nm technology, standard cells, and SVT transistors. Simulations were done in nominal conditions (1.2 V and 27°C) by performing Monte Carlo analysis with technology local and global process distributions using Spectre simulator in the Cadence environment.

A Trojan-free DUTT was simulated by analyzing 250 Monte-Carlo runs in order to generate the golden IC data. DUTT's primary inputs were set to V_{DD}, since our method takes no account of the system's switching activity. A train of 31 current pulses with different amplitude was

injected such as presented in Fig. 5.3 in order to evaluate the sensor efficiency. Subsequently, the efficiency was calculated for each run considering different PV. At the end of this first test, results of the Trojan-free DUTT were gathered, producing a distribution of the sensor efficiency with 250 samples.

Afterwards, Trojan-infected DUTTs were also simulated under the same conditions, delivering a distribution of the sensor efficiency with 250 samples. Finally, with the two sets of data (Trojan-free and Trojan-infected distributions), the KS-test was performed by taking Trojan-free data as the reference distribution and calculating the p-value for each one of the 250 samples.

5.3.2 Targeted HT Implanted in DUTTs

To ensure that our HT detection method is effective for tracking different types of Trojans, the technique was tested under worst case scenarios. For this reason, small Trojans were chosen to be implemented in this analysis since they induce negligible modifications on the side-channel parameters of DUTTs and even so, being able to provoke critical consequences in security systems, as stated in [5, 17, 73]. On the other hand, bigger HTs induce more variations on the side-channel parameters and are naturally more noticeable. Therefore, the success on tracking small hardware modifications indicates that this technique is also able to identify other more sophisticated HTs. Hence, three different minimalist HTs were applied on DUTTs to evaluate our method: (1) a minimum-sized inverter as the technology smallest gate (minimum drive capability) that emulates the worst gate-level Trojan case; (2) a single PMOS transistor with minimum width and length as a layout-level Trojan in which the detection is more difficult to be achieved than the previous mentioned gate-level Trojan due to its smaller dimensions; (3) a reduction by a factor of 1000 on the channel doping concentration of a PMOS transistor as another layout-level Trojan, emulating the HTs presented in [17, 73], able to make cryptographic systems more vulnerable to leak information. The detection of these 3 HTs indicates that the proposed method is able to detect any HT types composed of one or more gates (1); one or more transistors (2); or parametric HT making modifications on the channel doping concentration (3).

5.3.3 DUTTs Used to Generate Simulation Results

Table 5.1 summarizes the set of performed simulations and the obtained results. The first case studied was a DUTT composed of a chain of 10 inverters with one minimum-sized PMOS transistor inserted as HT. This system is monitored by one BBICS that produces an estimated area overhead of 28.5%. As long as the number of DUTT samples was increased, the p-value was calculated, as show in Fig. 5.6. This figure presents the progress p-value in relation to the number of tested devices. The analysis of the p-value results leads to the conclusion that a few DUTT samples are needed to detect the Trojan. The HT insertion upsets the original sensor efficiency distribution in such a way that it was possible to conclude with a probability

of 99% (p-value < 0.01) that approximately 11 ICs are needed to classify DUTT as Trojan-infected. This circuit can be even interpreted as a subcircuit of the DUTT in the second line of Table 5.1, composed of 10 chains of 10 inverters monitored by 10 sensors. If the same HT is inserted in one of their 10 subcircuits, the performance of the method remains unchanged since each individual BBICS works independently. Consequently, the ratio between HT and DUTT size (HT area overhead) is substantially reduced if DUTT is composed of several subcircuits. Therefore, all DUTTs in the Table 5.1 could be interpreted as subcircuits of a main system.

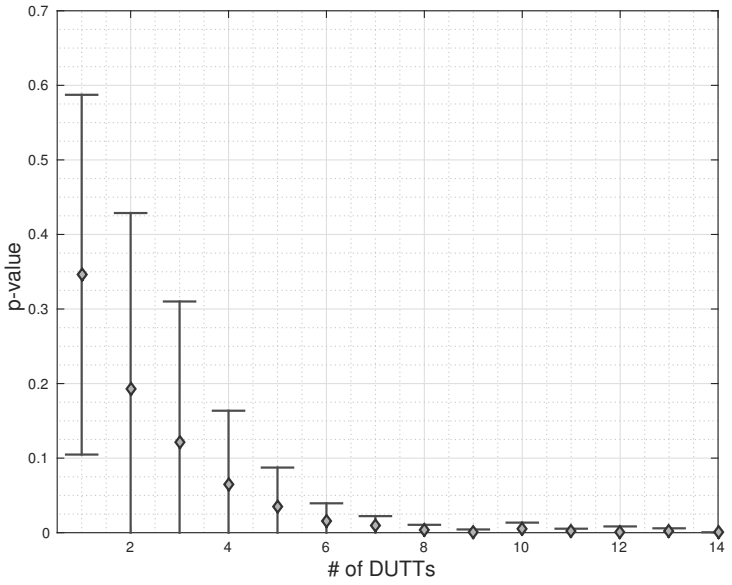


Fig. 5.6: Progress of p-value (with accuracy of $\pm\sigma$) in function of the number of DUTTs.

Results obtained from chains of 50 inverters, considering a transistor and an inverter as HT, illustrate the method effectiveness, even if the amount of gates monitored by the same BBICS is considerably increased. Benchmarks b01 and c17 are also considered in the analysis to generate results for DUTTs composed by other gates than inverters. At last, a chain of 10 inverters with a reduce doping concentration (dopant Trojan) is analyzed to emulate the HT presented in [17, 73], showing that our method is also suitable for this type of threat.

5.3.4 System Area Overhead and Number of Required Samples

The area overhead and the number of required samples are two balanced parameters chosen by the designers. As the overhead area is directly derived from the number of sensors implemented in the circuit, it is defined during the design phase. The designer can reduce this area offset if a large number of DUTT samples are available for testing. In order to illustrate it, results in Tab. 5.1 show that in a chain of 10 inverters only 11 DUTT samples are required to detect a HT. However, the BBICS used to monitor this circuit increases the total area of 28.5%. By enhancing the number of inverters to 50, the overhead area is reduced to 5.7% whereas 136 samples are needed. The other results in Tab. 5.1 show that no more than 150 DUTT samples

Table 5.1: Monte Carlo simulation results generating a detection probability of 99% over 250 data obtained from a Trojan-free DUTTs. Sensor and HT area overheads imposed on the total cell area of the Trojan-free DUTT.

DUTT	Number of Gates	Number of Transistors	Number of Built-in Sensors	System Area Overhead	Implanted HT	HT Area Overhead	Number of Required DUTT's samples
Chain of 10 Inverters	10	20	1	28.5%	1 PMOS	5%	11
10 Chains of 10 Inverters	100	200	10	28.5%	1 PMOS	0.5%	12
Chain of 50 Inverters	50	100	1	5.7%	1 PMOS	1 %	136
Chain of 50 Inverters	50	100	1	5.7%	1 Inverter	2%	101
c17	6	24	1	25%	1 Inverter	8%	8
b01	28	306	4	20%	1 Inverter	0.6%	93
Chain of 10 Inverters	10	20	1	28.5%	Dopant HT in all Inverters	0%	16

are needed if the overhead is larger than 5%. Moreover, if the circuit already features this BBICS to its original purpose (transient current detection) the area overhead needed for HT detection is negligible since it uses the same preexisting BBICS.

5.4 Conclusions and Perspectives

In this chapter, we presented a new HT detection method able to detect gate-, transistor- and layout-level Trojans even if the HT presents negligible variations on the classic side-channel signals (i.e. I_{DDT}, I_{DDQ}, delay, and EM-emissions). The impacts caused by a HT on the impedance of the DUTT substrate allow the detection of minimal alterations. Moreover, the effectiveness of the method on detecting stealth and small HT can be improved whether the whole system is split into smaller subcircuits, making the identification of the HT location also possible. The area overhead is negligible if the same sensors are reused as online-testing mechanisms for detecting transient currents. In addition, even if attackers are able to implant HTs on the BBICS circuit, the method offers the possibility of being self-monitored. For this purpose, besides monitoring its subcircuit, each BBICS also has to monitor the subsequent sensor. Consequently, all sensors would be monitored as well as the original circuit, avoiding HT insertion. The method is still applicable in combination with other side-channel techniques such as path delays and current leaks in order to further increase the HT detection coverage.

Chapter 6

Detection of Trojans in Asynchronous Circuits

The Internet-of-Things (IoT) is a trend in nowadays industry that renders a variety of physical devices able to collect and process data, and even to act in real-time in its environment. Thereby, designs with low-power consumption and high trustworthiness are crucial for the proper development of all the benefits that IoT can provide. To be competitive in this market, modern architectures must match robustness against attacks and faults with low power and performance.

Clockless circuits, also known as asynchronous circuits are an interesting alternative to deal with power consumption without compromising system performance [104]. Those architectures employ local communication protocols instead of a global clock for data synchronization, avoiding unnecessary dynamic power consumption in parts of the circuit that have no data to process at a certain point in time. Thanks to these protocols, the blocks synchronize themselves. A block may instantaneously initiate its operation after the execution of a previous block. This feature makes asynchronous designs faster than their synchronous counterparts. Moreover, the robustness of Quasi-Delay-Insensitive (QDI) class of asynchronous circuits against Differential Power Analysis (DPA) [57], transient-faults [95], and EM emissions [127] makes it also a good solution from the security point a view.

Some works were recently published showing that it is possible to implement Trojans in asynchronous circuits [55, 71], while others [86, 87] present strategies to detect threats in mixed macro-synchronous micro-asynchronous systems. However, for the best of our knowledge, none have ever studied the HT impacts in the side-channel signals of QDI asynchronous circuits nor proposed methods devoted for detecting Trojans in this type of circuit.

In this chapter, used as a base for the publication [1], we propose a testing technique for Trojan detection dedicated to QDI asynchronous circuits that exploits the pair of side-channel signals – transient current (I_{DDT}) and path delay (Δt) – without adding any extra circuitry nor

modifying the original post-silicon test set-up to statistically compare patterns from genuine ICs and DUTT. The proposed technique takes advantage of intrinsic supply current of asynchronous circuits, which produces separate traces from different blocks that compose the system. Therefore, we show that it is only required measuring global power supply current I_{DD} from the V_{DD} pin to obtain isolated side-channel signals from each block of the circuit. Moreover, a V_{DD} calibration procedure is presented to compensate PV masking effects, enhancing the HT detectability.

This chapter is organized as follows: Section 6.1 presents the background on asynchronous circuits. Section 6.2 introduces aspects of side-channel signals applied for HT detection in QDI asynchronous circuits. In section 6.3, we present our testing technique, while in section 6.4, simulation experiments are discussed. Section 6.5 concludes this chapter.

6.1 Fundamentals of QDI Asynchronous systems

The circuits based on a local communication protocol for data synchronization, instead of a global clock, are known as asynchronous circuits. In such architectures, a certain block (sender) only outputs a signal to the following one (receiver) if all its output channels are empty. The receiver will only start processing new data when all the necessary inputs are available. These two directives are the basis of a communication protocol. Respecting them is thus crucial for a correct functioning of asynchronous circuits.

A typical representation of an asynchronous system is shown in Fig. 6.1b. If compared to its synchronous equivalent, the basic difference noted is the absence of a clock signal and the addition of an acknowledgement signal. The latter is part of the local protocol that enables synchronization in asynchronous systems. It signalizes to the previous stage that the calculation is completed and new data can be processed.

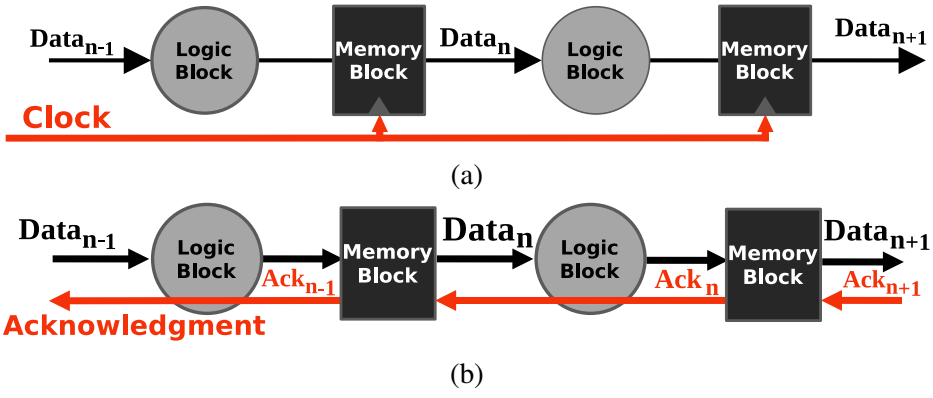


Fig. 6.1: Typical representations of synchronous (a) and asynchronous (b) systems.

Different strategies to conceive asynchronous circuits exist. They differ from each other in three basic aspects: the type of communication protocol implemented between blocks, the

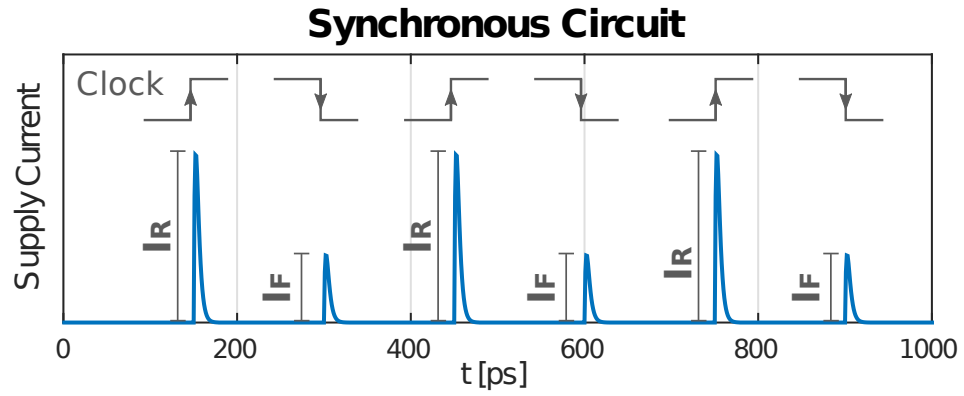
method used for encoding data, and the number of timing assumptions necessary for proper function [89, 104, 105]. In this context, QDI asynchronous circuits can operate correctly with only a few assumptions on some forks [89]. The reduced number of timing constraints requires a robust data encoding, allowing data validity to be signaled by the encoding itself. Therefore, dual-rail encoding is used for this purpose. In this case, the protocol validity signal (request) is merged into data signals. Thus, there is no physical difference between data and the communication protocol signal. This type of encoding is particularly robust against DPA based attacks, due to its power balance property which masks internal states [57]. Another advantages of QDI circuits have also been shown by numerous studies. For instance, lower-power consumption, EM emission, no clock distribution, nor skew issues [127]. Moreover, this class of circuits also features robustness against fault attacks, as discussed in [95]. Consequently, QDI asynchronous architectures are an attractive solution in terms of energy efficiency and security.

6.2 Side Channel Analysis Applied to QDI Asynchronous Circuits

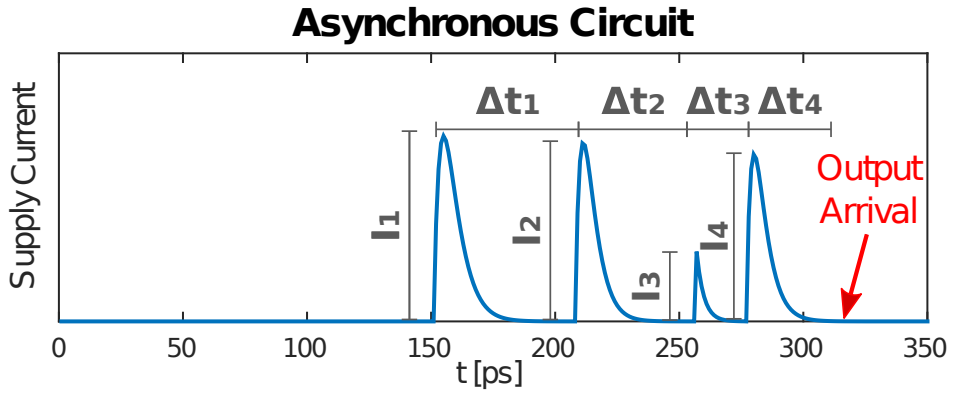
6.2.1 Exploiting Side-Channel Signals in QDI Asynchronous Circuits

As presented in section 6.1, local handshake pulses control data propagation through asynchronous circuits. Their generation, which occurs at any time, are governed by the latency of the successor and the predecessor blocks. Therefore, these pulses tend to be randomized over time, resulting in smoother supply current curves [127], without the large di/dt spikes as in synchronous circuits, as shown in Fig. 6.2a.

The asynchronous protocol results in a current trace as the one depicted in Fig. 6.2b. This example represents the current response of a single input vector passing through a 4-stage pipelined asynchronous circuit measured in the global power supply. As previously explained, as soon as one stage of an asynchronous circuit finishes its calculation, it acknowledges the end of processing to the previous stage, admitting new data to be processed if inputs are available. In case of a single input vector test, whenever the first stage outputs data to be processed by the second stage, the former will no longer have new calculations to do, and will become idle. The same behavior will be observed in every following stage of the asynchronous pipeline until all stages turn inactive. Therefore, only one stage of the pipeline is in fact active at a certain moment, while the other stages stand idle, waiting for new data to be processed. Thus, each peak in Fig. 6.2b corresponds to the operation of a single pipeline stage. For this reason, a Trojan inserted in an asynchronous circuit directly impacts the current peak that corresponds to the stage in which it has been inserted. Conversely, in synchronous circuits the global clock governs the switching activity of all pipeline stages simultaneously. Hence, the current peaks depicted in 6.2a represent the sum of the individual contribution of all elements that composes



(a)



(b)

Fig. 6.2: Abstraction of the supply current from synchronous (a) and asynchronous (b) circuits.

the circuit. The insertion of a Trojan in this case would impact the supply current response of the system as a whole, not only the pipeline stage in which it has been inserted.

The analysis above leads to two insights: the first one is that global supply current of asynchronous is more sensitive to Trojan effects than in synchronous circuits. The second one is that it is possible to obtain the transient current of each separate pipeline stage and the path delay only with the global supply current trace. The following paragraphs discuss how to obtain such parameters.

6.2.1.1 Global Path Delay (Δt)

In QDI asynchronous circuits, the delay of a pipeline stage is measured by the difference Δt_i between two current peaks (see Fig. 6.2b). As the global delay is given by $\Delta t = \sum_{i=1}^{n} \Delta t_i$, the deviation caused by a Trojan in one of its n pipeline stage is propagated to others subsequent blocks, delaying the original output. Therefore, measuring the delay Δt between the primary inputs and its arrival at the system output, is sufficient for malicious circuitry tracking. Obtaining path delay in asynchronous circuit requires only non-intrusive current measures on the power supply pin, seamlessly fitting in the regular testing phase with no extra cost in terms of design.

Although a small Trojan may increment only a few ps in the global path delay, nowadays on-the-fly Time to Digital Converter (TDC) based sensors can reach resolutions in the order of ps [43], rendering delay measurements possibles. Other sensors can reach even smaller resolutions in the order of hundreds of fs [45] at the price of multiple measurements. Moreover, as gate delays is inversely proportional to power supplies levels, two simple actions can increase feasibility of path delay measurements: (1) reduction of the power supply level V_{DD} and; (2) reduction of the substrate voltage V_{DDS}.

6.2.1.2 Transient Current (I_{DDT})

Since only the gates from a specific pipeline stage switches simultaneously, if a Trojan is inserted in a certain stage, its relative current peak is increased, making the Trojan impact highlighted. Therefore, a HT insertion in a given stage is recognized by the variation in its current amplitude. Some strategies such as clock gating or power gating could be implemented in synchronous circuits for the same purpose of isolating the switching activity of specific subcircuits. However, for that matter, additional circuitry would be needed to incorporate such a property, already available in asynchronous circuits.

Path delays and transient currents are correlated and mutually affected by PV. If the variation increases the current, it decreases the delay and vice-versa. The measurement of one variable allows deducing the range of the other, thus reducing the range of possible values of the other one. Therefore, the evaluation of the variables I_{DDT} and Δt allows reducing the impact of PV effects, as similarly demonstrated in [96].

6.2.2 Trojan Impacts on Side-Channel Signals

We propose to illustrate the Trojan effects on the current trace by a preliminary test considering PV effects in FD-SOI 28 nm technology. Figure 6.3 show an example of supply current traces obtained from 3-stage pipelined devices with and without a Trojan implanted in its second stage. The result figure depicts the impacts caused by Trojans (of approximately 1.3% of the original circuit area) in the current trace. It illustrates the motivation for the Trojan detection technique presented in this work. Note that the current in the first stage remains unaltered, whereas the current peaks in the second stage are clearly increased, and delayed in the third due to the Trojan effects. The necessary steps to perform Trojan detection are thus discussed in the following section.

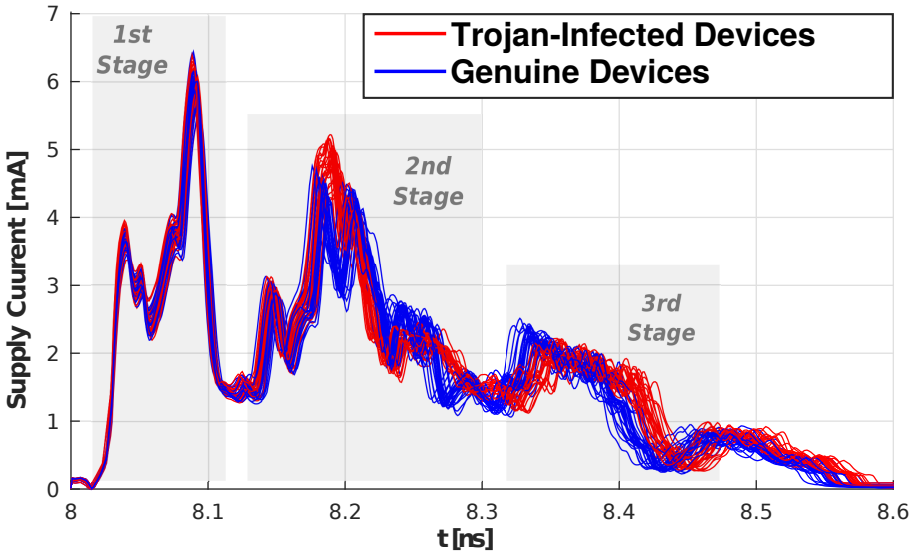


Fig. 6.3: Current curves in a 3-stage pipelined QDI asynchronous circuit obtained with 50 runs of Monte Carlo simulations. Blue traces were generated by genuine and red by Trojan-infected devices.

6.3 Proposed HT-Detection Technique for QDI Asynchronous Circuits

6.3.1 Test Procedure: Collecting I_{DDT} and Δt

Initially, the proposed test procedure requires collecting side-channel signatures from genuine devices (golden data) as in other methods. Subsequently, the same procedure is applied on each DUTT in order to produce results that will be statistically compared to the golden data.

As explained in section 6.2.1, the supply current trace is enough to obtain I_{DDT} and Δt, which are the necessary parameters for the proposed analysis. Thus, the current trace is directly measured from the global V_{DD} pin of each golden device using a test input vector X, from which the I_{DDT} peaks of each pipeline stage (I_{DDT1}, I_{DDT2}, I_{DDT3}) are extracted and stored. The Δt is obtained with the same test, however using a reduced V_{DD} level to facilitate the delay measurement, as discussed in 6.2.1.1. This test is repeated for each available genuine device. By the end, due to the PV effects, the collected parameters form a statistical distribution. Defining limits to such a distribution results in a data range in which measurements from genuine devices are expected to lay on. By performing the same test in each DUTT, it is possible to verify if its parameters belong to the generated distribution, thus classifying it as Trojan-free if the assumption is true, otherwise the DUTT is classified as Trojan-infected.

6.3.2 Voltage Tunning to Increase PV Compensation

Inter-die PV-induced V_{th} fluctuations are compensable by tuning each device supply voltage. The consequent reduction of PV masking effects decreases data dispersion, thus increasing Trojan detectability. Such a technique must be applied to both golden devices and DUTTs. The calibration method consists in slightly varying the original device V_{DD} value in order to produce an arbitrary preset global path delay Δt_o in each of the tested devices. Once the value Δt_o is obtained, the current peaks I_{DDT} are collected. Similarly, a test to obtain Δt is performed by tuning the V_{DD} value that produces a current peak I_{DDT_n} in a specific pipeline stage n for each device. At the end of these tests, a new pair I_{DDT} and Δt is obtained for each device, characterizing it.

The asynchronous architectures also allow strategies to compensate intra-die PV. For instance, the method in [49], also applicable to asynchronous circuits, propose splitting the substrate into several islands and applying different body bias (V_{DDS}) in each one, compensating local PV. However, this technique requires a mechanism to generate body biasing, which implies on area overhead. As the goal of this work is to present a technique without modifications on the original design and test set-up, this scheme is not considered in the result section. In any case, systems already featuring body biasing schemes for its original purpose are expected to be more effective for the Trojan detection purpose.

6.3.3 Minimum Necessary Test Vectors

In order to ensure that the whole circuit has been covered by the test vectors, a test methodology must cover all path branches of the system. The Fig. 6.4 shows a generic example of a n-stage pipelined QDI asynchronous system. In this diagram, the clouds and rectangles,respectively represent logic and memory blocks. In QDI asynchronous circuits, multiplexers and demultiplexers selectively activate or deactivate concurrent parts of the circuit depending on the data flow. For instance, a certain input test vector would lead data to pass exclusively by the blocks $C_{1,1}$ and $C_{n,1}$ of Fig 6.4, while the other blocks remains inactive. Therefore, if a Trojan is located in one of these inactive blocks, it will not affect the current trace in this test leading to a failed detection. To avoid this undesirable condition, a set of input test vectors that guarantees the activation of every concurrent blocks must be chosen. Since the same test covers blocks from different pipeline stages, the minimum number of required test vectors N_T is stated as the maximum amount of parallel branches in the system as presented in Eq. 6.1.

$$N_T = \max(K_i), \quad \forall\, i \in \mathbb{N}^* \leq n \tag{6.1}$$

where K_i is the number of concurrent blocks in the stage i and n is the number of pipeline stages in the system.

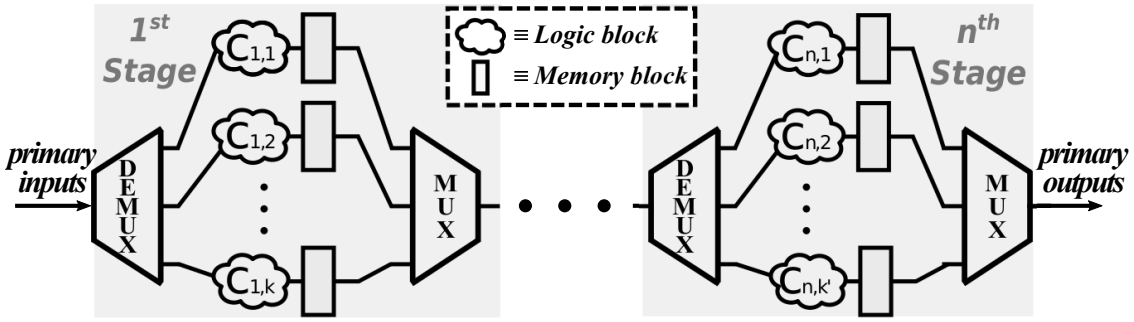


Fig. 6.4: Diagram of a pipelined system with n stages.

6.4 Experiments, Results, and Analysis

The techniques presented in section 6.3 were applied to the case-study circuit reported in 6.4.1. The Trojan used to infect DUTTs is described in 6.4.2. Both case-study and Trojan were synthesized with low threshold voltage transistors in FD-SOI 28 nm technology. The results were obtained by using 400 runs of Monte Carlo simulations performed at a reference temperature of 25^oC. Intra- and inter-die PV, and mismatch variations have been considered.

6.4.1 Target Case-Study: An 8-bit ALU

The QDI asynchronous 8-bit ALU proposed in [37] is the case-study circuit chosen for this study. It has 3 stages of pipeline and a total of 506 logic gates. Figure 6.5a depicts an abstraction of it. The mux and demux illustrate the existence of two concurrent branches in this architecture, which imposes a minimum of 2 input vectors for testing completion according to equation 6.1. These building blocks that compose the 8-bit ALU are also found in the AES representation of Figure 6.5b, a typical security-oriented circuit. Having a similar composition leads to the conclusion that side-channel signals from both AES and ALU would be similarly impacted by a Trojan. Consequently, the detection procedure presented in this chapter can be used to detect Trojans in other security-oriented QDI asynchronous circuits.

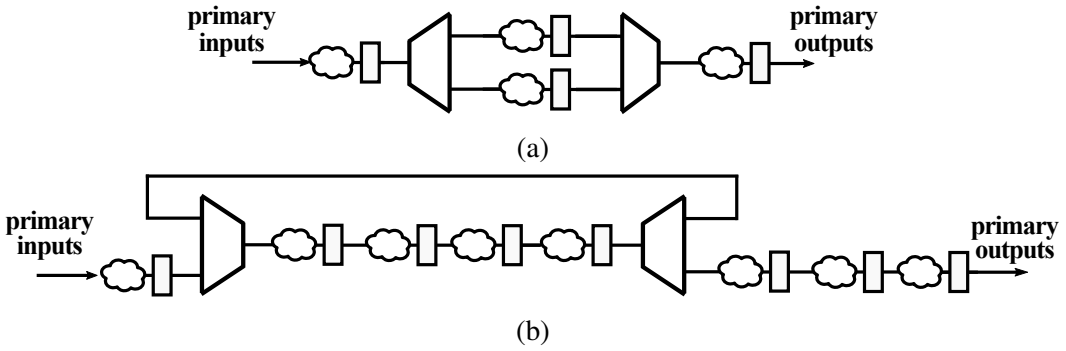


Fig. 6.5: Representation of the circuit pipeline stages from the ALU proposed by [37] (a) and a typical AES [7] (b).

6.4.2 Target Hardware Trojan

The Trojan model is a gate-level trigger that alters data flow whenever the input is set to a specific value, which is never tested during our simulations to make the detection more challenging. Therefore, the Trojan remains inactivated during all performed tests, which implies that its architecture is not pertinent for the detection. Although they are not active during the testing phase, its non-null side-channel leakages, that depends on their size, enable the detection. Simulations were done with Trojans representing 1.7%, 1.3% 1%, and 0.8% area of the design.

6.4.3 Test Description

Two simulations are performed in order to generate results for the technique proposed in 6.3: one to obtain I_{DDT} with supply voltage V_{DD} of 1V, and the second one to obtain global Δt, with V_{DD} of 0.6V. The latter is performed to simulate a real sensor limitation scenario, in which the available time resolution is $1ps$.

To perform the calibration technique presented in 6.3.2 a two-step procedure is executed: first, an arbitrary Δt_o of $371ps$ was chosen. Each device in the Monte Carlo simulation needs to have its V_{DD} set to a value that leads to the chosen Δt_o. The I_{DDT} value for each calibrated device is then stored. The second step, needed to obtain the global path delay, consists of setting V_{DD} to a value that generates an arbitrary current peak I_{DDT_o} of $950\mu A$ in the first pipeline stage of each device. The Δt value for each calibrated device is then stored. Each device (genuine and DUTT) is characterized by their current peaks I_{DDT}, obtained for a pre-defined Δt_o, and a global delay Δt, measured for a reference I_{DDT_o}.

Data collected from Trojan-free devices produce the signature of golden ICs by following the procedure explained in section 6.3 and the aforementioned calibration. Afterwards, the same design is infected with different Trojans, as presented in 6.4.2, to produce results that are statistically compared with the golden data. The parameter used to evaluate the results is the detection rate, defined as the percentage of Trojan data not pertaining to the Trojan-free distribution.

6.4.4 Results and Analysis

Results from Monte Carlo simulations performed with genuine and Trojan-infected devices are shown in Fig. 6.6. In these simulations, the Trojans have a total area of 1.6% of the original circuit area and were inserted in the second pipeline stage. An error ellipse with 95% of significance level surrounds the golden data, indicating a region in the parameters space where a circuit is accepted as Trojan-free. Note that the impact of the Trojan shifts the data from its original value to a greater current peak and global delay, as previously signaled in Fig. 6.3. As

no point from the Trojan-infected circuit is enclosed by the Trojan-free ellipse, the detection rate is 100%.

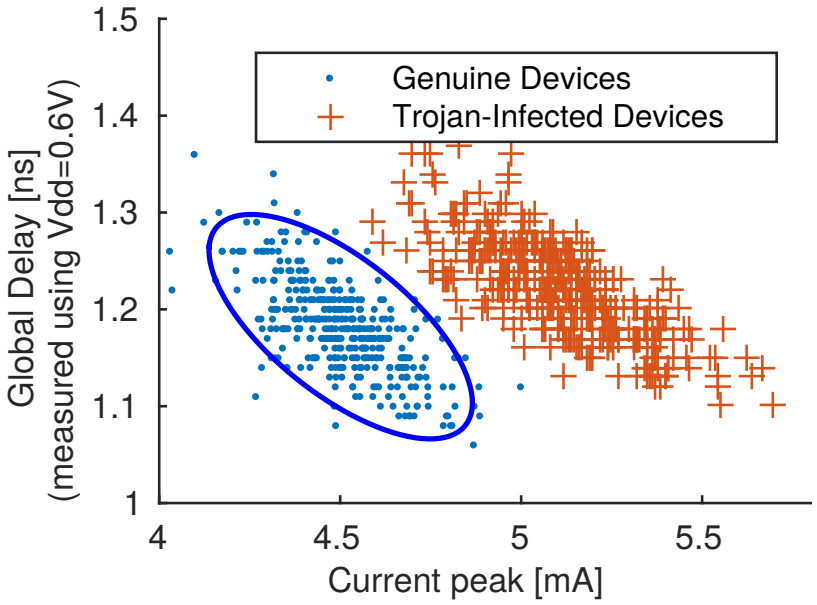


Fig. 6.6: Current peak in the second stage and the global delay obtained from Monte Carlo simulations in the Trojan-free and Trojan-infected ALU. The error ellipse surrounds the data from genuine devices with a significance level of 95%.

The same study is extended to other scenarios in which the Trojan size and location vary. Fig. 6.7 depicts the curves of detection rate versus Trojan size. The three graphs represent the results for a HT inserted in the first, second, and third stages respectively.

Results in Fig. 6.7 show that it is possible to detect Trojans representing 0.8% of the original circuit area with a detection rate of 100% if the Trojan is inserted in the first pipeline stage, 35% in the second, and 39% in the third using fixed V_{DD} values. The V_{DD} tuning process is capable of increasing the detection rate to 100%, 88%, and 47%, since the chip-to-chip PV are compensated. The curves in different stages are different, since the total number of gates vary in each stage. If the stages were equality balanced, the total dynamic consumption would be divided equally in all stages and, thus the curves would be more homogeneous. Furthermore, systems with more pipeline stages would present its dynamic power divided in more stages, enhancing the relative Trojan overhead in one of their stages. In order to enhance the detection rate, modifications in the original circuit could be made to increase the number of pipeline stages. Moreover, our technique can also be used combined with other methods for further enhancements. For instance, the use of extra power pins as in other methods [8, 114, 136] would improve the Trojan isolation, resulting in better detection rates.

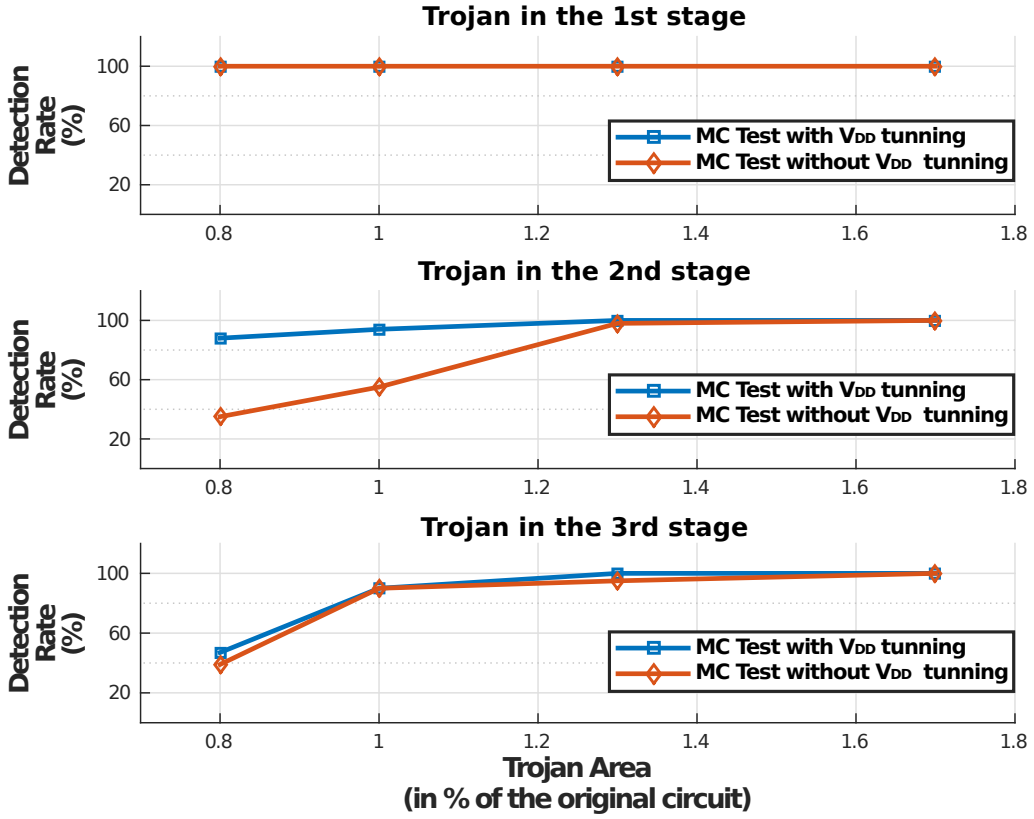


Fig. 6.7: Detection rate obtained using the techniques without (red) and with V_{DD} calibration technique (blue), in which different pipeline stages are infected by a Trojan.

6.5 Conclusions

In this chapter, we presented an efficient Trojan detection technique dedicated to QDI asynchronous circuits exploiting its inherent transient current and path delay characteristics. The distributed current peaks, intrinsic of asynchronous circuits, make them more sensitive to side-channel deviations than synchronous circuits, thus enhancing HT detection potential. Using a V_{DD} tuning process in order to reduce the impact of PV, it is possible to detect modifications smaller than 1% of the original circuit with a detection rate close to 100% without requiring any extra-circuitry. Moreover, the testbench set-up employed in the regular post-silicon testing phase can be reused for this purpose. Still, this technique can also be employed combined with any other methods proposed in the literature in order to enhance the obtained results, thus allowing the detection of even smaller Trojans.

Chapter 7

Conclusions and Perspectives

In modern IC production, designers must have means to ensure the trustworthiness of systems, even composed of untrusted components. Thus, HT detection techniques aim to verify and validate systems even if one or more steps in the production is not trusted. However, the process of certifying systems against HTs must be worthwhile to compensate the trade-off between detecting them and moderating the design and test costs. From there comes the main motivation for this thesis: reducing the costs dedicated to the HT detection by reusing architectures and mechanisms originally designed to deal with other security aspects.

Chapter 2 of this thesis discussed the state-of-the-art of Trojan implementation and detection, taking into account the different abstraction levels of designs to analyze differences between the existing detection techniques. Transistor-level Trojans were proposed in in chapter 3 in order to present possible attacks using only a few transistors. The tri-state logic makes HTs operate with only cut-off transistors, making its dynamic and quiescent power consumption negligible, and, therefore, rendering these Trojans almost imperceptible by current-based side-channel analysis.

Scalable sensors originally designed to detect transient faults – the BBICSs – have been analyzed and compared in chapter 4, moreover a new BBICS architecture with reduced area and high sensitivity has been proposed. Thereafter, in chapter 5, intrinsic properties of these sensors have been exploited to evaluate the substrate of circuits and detect Trojans. With no hardware modifications, a new off-line testing feature has been revealed to this sensor, which is indeed able to detect very small Trojans, on the order of the CMOS 65 nm technology's smallest gates and transistors. Furthermore, physical-level Trojans such as alterations in the channel doping concentration are also detectable. As a result, a U.S. patent specification of this technique has been recently described in partnership with a company.

In the context of QDI asynchronous systems, chapter 6 has shown that this class of circuit is intrinsically capable to generate current signatures with valuable timing and power consumption information to perform the detection of Trojans. The proposed HT detection technique only collects current signatures from a single power supply port requiring no hardware modification

in the original design.

The two HT detection techniques introduced in this thesis cover different aspects in HT detection. While the first technique is able to detect very-small absolute area HTs using multiple BBICS distributed around the design, the second one detects small relative area HTs –about 1% of the design–. Therefore, by using the capabilities of tracking side-channel and fault attacks at run-time from original architectures combined with our new off-line testing techniques able to detect HTs, we can deliver circuits with a very high protection against the most important types of IC hacking [54]. Consequently, adding this new HT detection feature amortizes the costs of providing secure systems.

Nevertheless, improvements have to be done in order to optimize these techniques. In the case of the BBICS-based technique, the methods and architectures for injecting current pulses with high accuracy amplitudes and resolution must be developed. Although a theoretical analysis provide a positive outlook, the efforts ensuring its feasibility still needs practical tests. Moreover, architectures able to compile and process the data generated by the multiple BBICS in the design still need to be developed. In the case of the second technique, the lack of available tools, libraries, standard benchmarks, and difficult integration in FPGAs hampers evaluating our method. The evaluation of our HT detection technique in silicon, considering different design implementation will provide directions to be followed in this study and further challenges.

Both techniques were integrated in a recent tapeout in FD-SOI 28 nm technology in partnership with ST Microelectronics and Tiempo. We designed different IPs in order to validate: (1) the overall behavior of our dynamic BBICS; (2) its sensitivity; (3) its capability for detecting Trojans in off-line mode and transient faults in on-line mode; (4) the detection technique based on the global current of QDI asynchronous circuits. These ICs will be delivered by the end of 2017 and the tests in the devices will allow to validate all the security approaches proposed in this thesis.

In future works, we envisage combining our BBICS with body-biasing cells, in order to generate a single structure able to:

1. Detect HTs
2. Detect side-channel attacks and transient faults
3. Control of body-biasing to compensate PVT and aging
4. Control of body-biasing to optimize the trade-off between performance and power consumption.

Thus, the integration of this hard IP in the design flow by replacing the traditional biasing filler cells will allow the automatic control of the body-biasing to balance power and performance, while monitoring the bulk will provide system protection against HTs, side-channel attacks, and transient faults.

In addition, the technique proposed in chapter 6, originally developed for detecting HTs in QDI asynchronous circuits will be suitable for other asynchronous circuit classes. Hence, we aim to expand our presented technique to obtain a generic methodology able to fit for all the different classes of asynchronous circuits with the same principle. This approach do not require any modification on the design for the HT detection purpose. With these works, we expect to finally confirm our assumption that asynchronous circuits are indeed more practical to detect Trojans than their synchronous counterparts. Ultimately, a proposition combining the BBICS with body-biasing cells structures in asynchronous circuits could detect different HT levels and feature all security and performance properties that these mechanisms and architectures can provide.

In the upcoming years, the design companies will become more and more specialized and will only provide some dedicated steps in the IC design flow. The largest semiconductor companies will integrate this advanced know-how at the risk of untrusted IPs or design flow steps. Indeed, outsourcing is risky during the design phase but also the fabrication. These risks create new threats and vulnerabilities, which will eventually be exploited for instance in military devices, secured smart cards or database servers in order to retrieve confidential information. Even worse - in the IoT world - hacking the control of cars or planes is also a possible scenario. Therefore researches have been devised for about ten years to detect hardware Trojans in order to develop trusted systems for the most sensitive applications.

With the proposed solutions, it is shown that the HT detection is viable, even if the Trojans are implanted during the fabrication phase by adding few transistors or modifying the dopant concentration. This is really encouraging for the future because the HT detection mechanisms based on BBICS open new trends for protecting complex chips with several third parties implied in the fabrication chain.

Moreover, the excellent properties of asynchronous circuits already used in cryptoprocessor designs and, more generally, in secured systems seems to offer real trends for fighting these threats. In fact, the proposed approach is really attractive because it offers the ability of detecting HT with no extra hardware, but at the price of an adapted test method.

Finally, all the proposed methods require a golden model and its definition is also challenging. Indeed, we have to think ways of building trusted systems with untrusted components.

Glossary

AES Advanced Encryption Standard.

ALU Arithmetic Logic Unit.

BBICS Bulk Built-in Current Sensor.

BICS Built-in Current Sensor.

CAD Computer-Aided Design

CMP Chemical Mechanical Polishing

CRC Cyclic Redundancy Check.

DARPA Defense Advanced Research Projects Agency.

DoS Denial-of-Service.

DPA Differential Power Analysis.

DUTT Design Under Trojan Test.

EM Electromagnetic.

FF Fast-Fast (process corner).

HT Hardware Trojan.

HVT High-threshold Voltage Transistors.

IC Integrated Circuit.

IoT Internet-of-Things.

IP Intellectual Propriety.

$\mathbf{I}_{DDT}$ Transient Current.

$\mathbf{I}_{DDQ}$ Quiescent Current.

KL Karhunen–Loève.

KS Kolmogrov-Smirnov (hypothesis test).

LVT Low-threshold Voltage Transistors.

NSF National Science Foundation.

PCB Printed Circuit Board.

PUF Physical Unclonable Function.

PV Process Variation.

PVT Process Voltage Temperature.

QDI Quasi Delay Insensitive.

RNG Random Number Generator.

RO Ring Oscillator.

RT Register Transfer.

SEQ Sequential Equivalence Checking.

SS Slow-Slow (process corner).

SVT Standard Threshold Voltage.

TCAD Technology Computer-Aided Design.

TDC Time to Digital Converter.

TRNG True Random Number Generator.

TT Typical-Typical (process corner).

$\mathbf{V}_{th}$ Threshold Voltage.

Bibliography of Author's Publications

[1] L. A. Guimarães, T. F. de Paiva Leite, R. Possamai Bastos, and L. Fesquet. Non-intrusive testing technique for detection of trojans in asynchronous circuits. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018.

[2] L. A. Guimarães, R. Possamai Bastos, and L. Fesquet. A new proposition on hardware trojan activation. In *18èmes Journées Nationales du Réseau Doctoral en Micro-nanoélectronique (JNRDM)*, 2015.

[3] L. A. Guimarães, R. Possamai Bastos, and L. Fesquet. Detection of layout-level trojans by injecting current into substrate and digitally monitoring built-in sensors. Work-in-Progress (WIP) session, 54nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2017.

[4] L. A. Guimarães, R. Possamai Bastos, and L. Fesquet. Detection of Layout-Level Trojans by Monitoring Substrate with Preexisting Built-in Sensors. In *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 290–295, July 2017.

[5] L. A. Guimarães, R. Possamai Bastos, T. F. de Paiva Leite, and L. Fesquet. Simple tri-state logic Trojans able to upset properties of ring oscillators. In *2016 International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS)*, pages 1–6, April 2016.

[6] R. Possamai Bastos, L. A. Guimarães, F. Sill Torres, and L. Fesquet. Architectures of bulk built-in current sensors for detection of transient faults in integrated circuits. *Microelectronics Journal*, 71:70 – 79, 2018.

References

[7] FIPS PUB 197, Advanced Encryption Standard (AES), 2001. U.S. Department of Commerce/National Institute of Standards and Technology.

[8] AARESTAD, J., ACHARYYA, D., RAD, R., AND PLUSQUELLIC, J. Detecting trojans through leakage current analysis using multiple supply pad I_{ddq} s. *IEEE Transactions on Information Forensics and Security 5*, 4 (Dec 2010), 893–904.

[9] ADEE, S. The hunt for the kill switch. *IEEE Spectrum 45*, 5 (May 2008), 34–39.

[10] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *SP* (2007), pp. 296–310.

[11] ALKABANI, Y., AND KOUSHANFAR, F. Consistency-based characterization for ic trojan detection. In *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers* (Nov 2009), pp. 123–127.

[12] ATHAN, S. P., ET AL. A novel built-in current sensor for iddq testing of deep submicron cmos ics. In *Proc. IEEE (VTS'96)* (1996), pp. 118–123.

[13] BA, P. S., DUPUIS, S., PALANICHAMY, M., MARIE-LISE-FLOTTES, NATALE, G. D., AND ROUZEYRE, B. Hardware trust through layout filling: A hardware trojan prevention technique. In *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (July 2016), pp. 254–259.

[14] BALASCH, J., GIERLICHS, B., AND VERBAUWHEDE, I. Electromagnetic circuit fingerprints for hardware trojan detection. In *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)* (Aug 2015), pp. 246–251.

[15] BANGA, M., AND HSIAO, M. S. Trusted rtl: Trojan detection methodology in pre-silicon designs. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (June 2010), pp. 56–59.

[16] BAYON, P., BOSSUET, L., AUBERT, A., FISCHER, V., POUCHERET, F., ROBISSON, B., AND MAURINE, P. Contactless electromagnetic active attack on ring oscillator based

true random number generator. In *Constructive Side-Channel Analysis and Secure Design*, vol. 7275. Springer Berlin Heidelberg, 2012, pp. 151–166.

[17] BECKER, G. T., REGAZZONI, F., PAAR, C., BURLESON, WAYNE P.", E. G., AND CORON, J.-S. *Stealthy Dopant-Level Hardware Trojans*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 197–214.

[18] BHASIN, S., DANGER, J.-L., GUILLEY, S., NGO, X., AND SAUVAGE, L. Hardware trojan horses in cryptographic ip cores. In *FDTC* (2013), pp. 15–29.

[19] CAKIR, B., AND MALIK, S. Hardware trojan detection for gate-level ics using signal correlation based clustering. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2015), pp. 471–476.

[20] CAO, Y., CHANG, C. H., AND CHEN, S. A cluster-based distributed active current sensing circuit for hardware trojan detection. *IEEE Transactions on Information Forensics and Security 9*, 12 (Dec 2014), 2220–2231.

[21] CHA, B., AND GUPTA, S. K. Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2013), pp. 1265–1270.

[22] CHA, H., AND PATEL, J. A logic-level model for alpha-particle hits in cmos circuits. In *Proc. IEEE International Conference on Computer Design (ICCD'93)* (1993), pp. 538–542.

[23] CHAKRABORTY, R. S., AND BHUNIA, S. Security against hardware trojan through a novel application of design obfuscation. In *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers* (Nov 2009), pp. 113–116.

[24] CHAKRABORTY, R. S., NARASIMHAN, S., AND BHUNIA, S. Hardware trojan: Threats and emerging solutions. In *2009 IEEE International High Level Design Validation and Test Workshop* (Nov 2009), pp. 166–171.

[25] CHAKRABORTY, R. S., PAUL, S., AND BHUNIA, S. On-demand transparency for improving hardware trojan detectability. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (June 2008), pp. 48–50.

[26] CHAKRABORTY, R. S., SAHA, I., PALCHAUDHURI, A., AND NAIK, G. K. Hardware trojan insertion by direct modification of fpga configuration bitstream. *IEEE Design Test 30*, 2 (April 2013), 45–54.

[27] CHAKRABORTY, R. S., WOLFF, F., PAUL, S., PAPACHRISTOU, C., AND BHUNIA", S. *"MERO: A Statistical Approach for Hardware Trojan Detection"*. "Springer Berlin Heidelberg", "Berlin, Heidelberg", "2009", pp. "396–410".

[28] CHAMPEIX, C., ET AL. Experimental validation of a bulk built-in current sensor for detecting laser-induced currents. In *IOLTS* (2015), pp. 150–155.

[29] CHEN, L., ET AL. Methods and devices for detecting single-event transients. In *U.S. patent no. 8,451,028.* (2013).

[30] CHEN, X., LIU, Q., YAO, S., WANG, J., XU, Q., WANG, Y., LIU, Y., AND YANG, H. Hardware trojan detection in third-party digital intellectual property cores by multi-level feature analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems PP*, 99 (2017), 1–1.

[31] CHEN, X., WANG, L., WANG, Y., LIU, Y., AND YANG, H. A general framework for hardware trojan detection in digital circuits by statistical learning algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36*, 10 (Oct 2017), 1633–1646.

[32] CHERKAOUI, A., FISCHER, V., AUBERT, A., AND FESQUET, L. A self-timed ring based true random number generator. In *Asynchronous Circuits and Systems (ASYNC), 2013 IEEE 19th International Symposium on* (May 2013), pp. 99–106.

[33] CHIANG, C., AND KAWA, J. *Design for Manufacturability and Yield for Nano-Scale CMOS*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[34] COURBON, F., LOUBET-MOUNDI, P., FOURNIER, J. J. A., AND TRIA, A. A high efficiency hardware trojan detection technique based on fast sem imaging. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2015), pp. 788–793.

[35] CUI, X., MA, K., SHI, L., AND WU, K. High-level synthesis for run-time hardware trojan detection and recovery. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)* (June 2014), pp. 1–6.

[36] DAVOODI, A., LI, M., AND TEHRANIPOOR, M. A sensor-assisted self-authentication framework for hardware trojan detection. *IEEE Design Test 30*, 5 (Oct 2013), 74–82.

[37] DE PAIVA LEITE, T. F., BASTOS, R. P., JADUE, R. I., AND FESQUET, L. Comparison of low-voltage scaling in synchronous and asynchronous fd-soi circuits. In *2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)* (Sept 2016), pp. 229–234.

[38] DODD, P., ET AL. Production and propagation of single-event transients in high-speed digital logic ics. *IEEE Trans. Nuclear Science 51*, 6 (2004), 3278–3284.

[39] DUBROVA, E., NÄSLUND, M., CARLSSON, G., AND SMEETS, B. Keyed logic bist for trojan detection in soc. In *2014 International Symposium on System-on-Chip (SoC)* (Oct 2014), pp. 1–4.

[40] DUPUIS, S., DI NATALE, G., FLOTTES, M.-L., AND ROUZEYRE, B. Identification of Hardware Trojans triggering signals. In *First Workshop on Trustworthy Manufacturing and Utilization of Secure Devices* (Avignon, France, May 2013).

[41] DUTERTRE, J.-M., ET AL. Sensitivity tuning of a bulk built-in current sensor for optimal transient-fault detection. *Microelectronics Reliability 53*, 9 (2013), 1320–1324.

[42] DUTERTRE, J.-M., POSSAMAI BASTOS, R., POTIN, O., FLOTTES, M.-L., ROUZEYRE, B., DI NATALE, G., AND SARAFIANOS, A. Improving the ability of Bulk Built-In Current Sensors to detect Single Event Effects by using triple-well CMOS. *Microelectronics Reliability 54*, 9-10 (2014), 2289–2294.

[43] EL-HADBI, A., CHERKAOUI, A., ELISSATI, O., SIMATIC, J., AND FESQUET, L. On-the-fly and sub-gate-delay resolution tdc based on self-timed ring: A proof of concept. In *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)* (June 2017), pp. 305–308.

[44] ELLIS, S., GAO, Y., AND WANG, C. Tsmc ready to spend $20 billion on its most advanced chip plant. October 2017, [Online]. Available: https://www.bloomberg.com/news/articles/2017-10-06/tsmc-ready-to-spend-20-billion-on-its-most-advanced-chip-plant, Accessed 10 October 2017.

[45] ELSHAZLY, A., RAO, S., YOUNG, B., AND HANUMOLU, P. K. A noise-shaping time-to-digital converter using switched-ring oscillators—analysis, design, and measurement techniques. *IEEE Journal of Solid-State Circuits 49*, 5 (May 2014), 1184–1197.

[46] FERLET-CABROIS, V., ET AL. Statistical analysis of the charge collected in soi and bulk devices under heavy ion and proton irradiation—implications for digital sets. *IEEE Trans. Nuclear Science 53*, 6 (2006), 3242–3252.

[47] FORTE, D., BAO, C., AND SRIVASTAVA, A. Temperature tracking: An innovative run-time approach for hardware trojan detection. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (Nov 2013), pp. 532–539.

[48] FRANCQ, J., AND FRICK, F. Introduction to hardware trojan detection methods. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2015), pp. 770–775.

[49] GARG, S., AND MARCULESCU, D. System-level leakage variability mitigation for mpsoc platforms using body-bias islands. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 20*, 12 (Dec 2012), 2289–2301.

[50] GILL, B., ET AL. An efficient bics design for seus detection and correction in semiconductor memories. In *Proc. IEEE (DATE'05)* (2005), pp. 592–597.

[51] GODLEWSKI, C., ET AL. Electrical modeling of the effect of beam profile for pulsed laser fault injection. *Microelectronics Reliability 49*, 9-11 (2009), 1143–1147.

[52] GOLIC, J. D. J. New methods for digital generation and postprocessing of random data. *IEEE Transactions on Computers 55*, 10 (Oct 2006), 1217–1229.

[53] GUIMARÃES, M. V., AND TORRES, F. S. Automatic layout integration of bulk built-in current sensors for detection of soft errors. In *2016 29th Symposium on Integrated Circuits and Systems Design (SBCCI)* (Aug 2016), pp. 1–6.

[54] HAMDIOUI, S., DANGER, J. L., NATALE, G. D., SMAILBEGOVIC, F., VAN BATTUM, G., AND TEHRANIPOOR, M. Hacking and protecting ic hardware. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2014), pp. 1–7.

[55] HASAN, S. R., MOSSA, S. F., PEREZ, C., AND AWWAD, F. Hardware trojans in asynchronous fifo-buffers: From clock domain crossing perspective. In *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)* (Aug 2015), pp. 1–4.

[56] HE, J., ZHAO, Y., GUO, X., AND JIN, Y. Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25*, 10 (Oct 2017), 2939–2948.

[57] HO, W. G., PAMMU, A. A., LIU, N., NE, K. Z. L., CHONG, K. S., AND GWEE, B. H. Security analysis of asynchronous-logic qdi cell approach for differential power analysis attack. In *2016 International Symposium on Integrated Circuits (ISIC)* (Dec 2016), pp. 1–4.

[58] HOQUE, T., MUSTAPA, M., AMSAAD, F., AND NIAMAT, M. Assessment of nand based ring oscillator for hardware trojan detection. In *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)* (Aug 2015), pp. 1–4.

[59] HOU, B., HE, C., WANG, L., EN, Y., AND XIE, S. Hardware trojan detection via current measurement: A method immune to process variation effects. In *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)* (Aug 2014), pp. 1039–1042.

[60] HU, K., NOWROZ, A. N., REDA, S., AND KOUSHANFAR, F. High-sensitivity hardware trojan detection using multimodal characterization. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2013), pp. 1271–1276.

[61] IQBAL, A. Understanding integrated circuit security threats. [Online]. Available: https://sdm.mit.edu/news/news_articles/webinar_021014/iqbal_021014.pdf. Accessed 22 Sept. 2017.

[62] ISMARI, D., PLUSQUELLIC, J., LAMECH, C., BHUNIA, S., AND SAQIB, F. On detecting delay anomalies introduced by hardware trojans. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (Nov 2016), pp. 1–7.

[63] JACOB, N., MERLI, D., HEYSZL, J., AND SIGL, G. Hardware trojans: current challenges and approaches. *IET Computers Digital Techniques 8*, 6 (2014), 264–273.

[64] JACOB, N., MERLI, D., HEYSZL, J., AND SIGL, G. Hardware trojans: current challenges and approaches. *Computers Digital Techniques, IET 8*, 6 (2014), 264–273.

[65] JHA, S., AND JHA, S. K. Randomization based probabilistic approach to detect trojan circuits. In *2008 11th IEEE High Assurance Systems Engineering Symposium* (Dec 2008), pp. 117–124.

[66] JIN, Y., AND MAKRIS, Y. Hardware trojan detection using path delay fingerprint. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (June 2008), pp. 51–57.

[67] KARIMIAN, N., TEHRANIPOOR, F., RAHMAN, M. T., KELLY, S., AND FORTE, D. Genetic algorithm for hardware trojan detection with ring oscillator network (ron). In *2015 IEEE International Symposium on Technologies for Homeland Security (HST)* (April 2015), pp. 1–6.

[68] KARNIK, T., HAZUCHA, P., AND PATEL, J. Characterization of soft errors caused by single event upsets in cmos processes. *IEEE Trans. Dependable and Secure Computing 1*, 2 (2004), 128–143.

[69] KITSOS, P., AND VOYIATZIS, A. G. Towards a hardware trojan detection methodology. In *2014 3rd Mediterranean Conference on Embedded Computing (MECO)* (June 2014), pp. 18–23.

[70] KOUSHANFAR, F., AND MIRHOSEINI, A. A unified framework for multimodal submodular integrated circuits trojan detection. *IEEE Transactions on Information Forensics and Security 6*, 1 (March 2011), 162–174.

[71] KOUTARO INABA, T. Y., AND IMAI, M. Hardware trojan asynchronous noc router. In *Asynchronous Circuits and Systems (ASYNC), 2017 IEEE 24th International Symposium on* (May 2017). to be published.

[72] KUMAR, R., JOVANOVIC, P., BURLESON, W., AND POLIAN, I. Parametric trojans for fault-injection attacks on cryptographic hardware. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on* (Sept 2014), pp. 18–28.

[73] KUMAR, R., JOVANOVIC, P., BURLESON, W., AND POLIAN, I. Parametric trojans for fault-injection attacks on cryptographic hardware. In *FDTC* (2014), pp. 18–28.

[74] LAMECH, C., AARESTAD, J., PLUSQUELLIC, J., RAD, R., AND AGARWAL, K. Rebel and tdc: Two embedded test structures for on-chip measurements of within-die path delay variations. In *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (Nov 2011), pp. 170–177.

[75] LAMECH, C., AND PLUSQUELLIC, J. Trojan detection based on delay variations measured using a high-precision, low-overhead embedded test structure. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust* (June 2012), pp. 75–82.

[76] LECOMTE, M., FOURNIER, J. J. A., AND MAURINE, P. Thoroughly analyzing the use of ring oscillators for on-chip hardware trojan detection. In *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)* (Dec 2015), pp. 1–6.

[77] LEITE, F., ET AL. Using bulk built-in current sensors and recomputing techniques to mitigate transient faults in microprocessors. In *Proc. Latin American Test Workshop (LATW'09)* (2009).

[78] LESPERANCE, N., KULKARNI, S., AND CHENG, K.-T. Hardware trojan detection using exhaustive testing of k-bit subspaces. In *The 20th Asia and South Pacific Design Automation Conference* (Jan 2015), pp. 755–760.

[79] LI, H., LIU, Q., ZHANG, J., AND LYU, Y. A survey of hardware trojan detection, diagnosis and prevention. In *2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)* (Aug 2015), pp. 173–180.

[80] LI, J., AND LACH, J. Negative-skewed shadow registers for at-speed delay variation characterization. In *2007 25th International Conference on Computer Design* (2007), pp. 354–359.

[81] LI, J., AND LACH, J. At-speed delay characterization for ic authentication and trojan horse detection. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (June 2008), pp. 8–14.

[82] LISBOA, C., ET AL. Using built-in sensors to cope with long duration transient faults in future technologies. In *ITC* (2007), pp. 1–10.

[83] LIU, Y., HUANG, K., AND MAKRIS, Y. Hardware trojan detection through golden chip-free statistical side-channel fingerprinting. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)* (June 2014), pp. 1–6.

[84] LIU, Y., JIN, Y., NOSRATINIA, A., AND MAKRIS, Y. Silicon demonstration of hardware trojan design and detection in wireless cryptographic ics. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25*, 4 (April 2017), 1506–1519.

[85] LO, J., ET AL. Design of static cmos self-checking circuits using built-in current sensing. In *Proc. IEEE (FTCS'92)* (1992), pp. 104–111.

[86] LODHI, F. K., HASAN, S. R., HASAN, O., AND AWWAD, F. Hardware trojan detection in soft error tolerant macro synchronous micro asynchronous (msma) pipeline. In *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)* (Aug 2014), pp. 659–662.

[87] LODHI, F. K., HASAN, S. R., HASAN, O., AND AWWAD, F. Formal analysis of macro synchronous micro asychronous pipeline for hardware trojan detection. In *2015 Nordic Circuits and Systems Conference (NORCAS): NORCHIP International Symposium on System-on-Chip (SoC)* (Oct 2015), pp. 1–4.

[88] MAITI, A., AND SCHAUMONT, P. Improving the quality of a physical unclonable function using configurable ring oscillators. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on* (Aug 2009), pp. 703–707.

[89] MARTIN, A., AND NYSTROM, M. Asynchronous techniques for system-on-chip design. *Proceedings of the IEEE 94*, 6 (2006), 1089–1120.

[90] MASSEY JR, F. J. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association 46*, 253 (1951), 68–78.

[91] MELO, J. G. M., AND SILL TORRES, F. Exploration of noise impact on integrated bulk current sensors. *Journal of Electronic Testing, Theory and Applications (JETTA) 32*, 2 (2016), 163–173.

[92] MELO, J. G. M., TORRES, F. S., AND BASTOS, R. P. Exploration of noise robustness and sensitivity of bulk current sensors for soft error detection. In *VARI* (2015), pp. 13–18.

[93] MESSENGER, G. C. Collection of charge on junction nodes from ion tracks. *IEEE Transactions on Nuclear Science 29*, 6 (Dec 1982), 2024–2031.

[94] MOEIN, S., SUBRAMNIAN, J., GULLIVER, T. A., GEBALI, F., AND EL-KHARASHI, M. W. Classification of hardware trojan detection techniques. In *2015 Tenth International Conference on Computer Engineering Systems (ICCES)* (Dec 2015), pp. 357–362.

[95] MONNET, Y., RENAUDIN, M., AND LEVEUGLE, R. Designing resistant circuits against malicious faults injection using asynchronous logic. *IEEE Transactions on Computers 55*, 9 (Sept 2006), 1104–1115.

[96] NARASIMHAN, S., DU, D., CHAKRABORTY, R. S., PAUL, S., WOLFF, F. G., PAPACHRISTOU, C. A., ROY, K., AND BHUNIA, S. Hardware trojan detection by multiple-parameter side-channel analysis. *IEEE Transactions on Computers 62*, 11 (Nov 2013), 2183–2195.

[97] NARSALE, A., AND M. C. HUANG, M. C. Variation-tolerant hierarchical voltage monitoring circuit for soft error detection. In *Proc. IEEE (ISQED'09)* (2009), pp. 799–805.

[98] NDAI, P., ET AL. A soft error monitor using switching current detection. In *Proc. IEEE (ICCD'05)* (2005), pp. 185–190.

[99] NENNI, D., AND MCLELLAN, P. *Fabless: The Transformation of the Semiconductor Industry*. Createspace Independent Pub, 2014.

[100] NETO, E. H., ET AL. Evaluating fault coverage of bulk built-in current sensor for soft errors in combinational and sequential logic. In *Proc. Symposium on Integrated Circuits and Systems Design (SBCCI'12)* (2005), pp. 62–67.

[101] NETO, E. H., ET AL. Using bulk built-in current sensors to detect soft errors. *IEEE Micro 26*, 5 (2006), 10–18.

[102] NETO, E. H., KASTENSMIDT, F. L., AND WIRTH, G. Tbulk-bics: A built-in current sensor robust to process and temperature variations for soft error detection. *IEEE Trans. Nuclear Science 55*, 4 (2008), 2281–2288.

[103] NGO, X. T., EXURVILLE, I., BHASIN, S., DANGER, J. L., GUILLEY, S., NAJM, Z., RIGAUD, J. B., AND ROBISSON, B. Hardware trojan detection by delay and electromagnetic measurements. In *DATE* (2015), pp. 782–787.

[104] NOWICK, S., AND SINGH, M. Asynchronous design #x2014;part 1: Overview and recent advances. *Design Test, IEEE 32*, 3 (2015), 5–18.

[105] NOWICK, S., AND SINGH, M. Asynchronous design #x2014;part 2: Systems and methodologies. *Design Test, IEEE 32*, 3 (2015), 19–28.

[106] ON HIGH PERFORMANCE MICROCHIP SUPPLY, U. S. D. S. B. T. F., UNITED STATES. OFFICE OF THE UNDER SECRETARY OF DEFENSE FOR ACQUISITION, T., AND LOGISTICS. *Defense Science Board Task Force on High Performance Microchip Supply*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, 2005.

[107] OYA, M., SHI, Y., YANAGISAWA, M., AND TOGAWA, N. A score-based classification method for identifying hardware-trojans at gate-level netlists. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)* (March 2015), pp. 465–470.

[108] POSSAMAI BASTOS, R., ET AL. A new bulk built-in current sensor-based strategy for dealing with long-duration transient faults in deep-submicron technologies. In *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT'11)* (2011), pp. 302–308.

[109] POSSAMAI BASTOS, R., ET AL. Novel transient-fault detection circuit featuring enhanced bulk built-in current sensor with low-power sleep-mode. *Microelectronics Reliability 52*, 9-10 (2012), 1781–1786.

[110] POSSAMAI BASTOS, R., ET AL. A bulk built-in sensor for detection of fault attacks. In *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'13)* (2013), pp. 51–54.

[111] POSSAMAI BASTOS, R., ET AL. A new recovery scheme against short-to-long duration transient faults in combinational logic. *Journal of Electronic Testing, Theory and Applications (JETTA) 29*, 3 (2013), 331–340.

[112] POSSAMAI BASTOS, R., ET AL. A single built-in sensor to check pull-up and pull-down cmos networks against transient faults. In *Proc. International Workshop on Power and Timing Modeling, Optimization, and Simulation (PATMOS'13)* (2013), pp. 157–163.

[113] POTKONJAK, M., NAHAPETIAN, A., NELSON, M., AND MASSEY, T. Hardware trojan horse detection using gate-level characterization. In *2009 46th ACM/IEEE Design Automation Conference* (July 2009), pp. 688–693.

[114] RAD, R., PLUSQUELLIC, J., AND TEHRANIPOOR, M. Sensitivity analysis to hardware trojans using power supply transient signals. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (June 2008), pp. 3–7.

[115] RAD, R. M., WANG, X., TEHRANIPOOR, M., AND PLUSQUELLIC, J. Power supply signal calibration techniques for improving detection resolution to hardware trojans. In *2008 IEEE/ACM International Conference on Computer-Aided Design* (Nov 2008), pp. 632–639.

[116] RAJALAKSHMI, T. R., AND SUDHAKAR, R. A novel carbon nanotubefet based bulk built-in current sensor for single event upset detection. *Sadhana 41*, 5 (2016), 489–495.

[117] REECE, T., AND ROBINSON, W. H. Detection of hardware trojans in third-party intellectual property using untrusted modules. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35*, 3 (March 2016), 357–366.

[118] SALMANI, H., TEHRANIPOOR, M., AND KARRI, R. On design vulnerability analysis and trust benchmark development. In *ICCD* (2013).

[119] SALMANI, H., TEHRANIPOOR, M., AND PLUSQUELLIC, J. A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 20*, 1 (Jan 2012), 112–125.

[120] SHAKYA, B., HE, T., SALMANI, H., FORTE, D., BHUNIA, S., AND TEHRANIPOOR, M. Benchmarking of hardware trojans and maliciously affected circuits. *Journal of Hardware and Systems Security 1*, 1 (Mar 2017), 85–102.

[121] SILL TORRES, F., AND POSSAMAI BASTOS, R. Robust modular bulk built-in current sensors for detection of transient faults. In *Proc. Symposium on Integrated Circuits and Systems Design (SBCCI'12)* (2012), pp. 1–6.

[122] SILL TORRES, F., AND POSSAMAI BASTOS, R. Detection of transient faults in nanometer technologies by using modular built-in current sensors. *Journal of Integrated Circuits and Systems (JICS) 8*, 2 (2013), 89–97.

[123] SIMIONOVSKI, A., AND WIRTH, G. Simulation evaluation of an implemented set of complementary bulk built-in current sensors with dynamic storage cell. *IEEE Trans. Device and Materials Reliability 14*, 1 (2013), 255–261.

[124] SIMIONOVSKI, A., AND WIRTH, G. Adding a self-reset feature to the bulk-bics with dynamic storage cell. *Microelectronics Reliability* (2015).

[125] SIMIONOVSKI, A., AND WIRTH, G. I. A bulk built-in current sensor for set detection with dynamic memory cell. In *2012 IEEE 3rd Latin American Symposium on Circuits and Systems (LASCAS)* (Feb 2012), pp. 1–4.

[126] SIMIONOVSKI, A., AND WIRTH, G. I. A bulk built-in current sensor for set detection with dynamic memory cell. In *LASCAS* (2012), pp. 1–4.

[127] SPARSØ, J., AND FURBER, S. *Principles of Asynchronous Circuit Design: A Systems Perspective*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[128] SUNAR, B., MARTIN, W., AND STINSON, D. A provably secure true random number generator with built-in tolerance to active attacks. *Computers, IEEE Transactions on 56*, 1 (Jan 2007), 109–119.

[129] SÖLL, O., KORAK, T., MUEHLBERGHUBER, M., AND HUTTER, M. Em-based detection of hardware trojans on fpgas. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (May 2014), pp. 84–87.

[130] TEHRANIPOOR, M., AND KOUSHANFAR, F. A survey of hardware trojan taxonomy and detection. *IEEE Design Test of Computers 27*, 1 (Jan 2010), 10–25.

[131] TEHRANIPOOR, M., SALMANI, H., ZHANG, X., WANG, M., KARRI, R., RAJENDRAN, J., AND ROSENFELD, K. Trustworthy hardware: Trojan detection and design-for-trust challenges. *Computer 44*, 7 (July 2011), 66–74.

[132] TEHRANIPOOR, M., AND WANG, C. *Introduction to Hardware Security and Trust*. Springer Publishing Company, Incorporated, 2011.

[133] VARGAS, F., AND NICOLAIDIS, M. Seu-tolerant sram design based on current monitoring. In *Proc. IEEE (FTCS'94)* (1994), pp. 106–115.

[134] VIERA, R. A. C., ET AL. Evaluation of bulk built-in current sensors detecting multiple transient faults. In *Proc. IEEE Asian Test Symposium (ATS'15)* (2013), pp. 157–163.

[135] WANG, H. B., ET AL. A novel built-in current sensor for n-well set detection. *Journal of Electronic Testing, Theory and Applications (JETTA) 31*, 4 (2015), 395–401.

[136] WANG, X., SALMANI, H., TEHRANIPOOR, M., AND PLUSQUELLIC, J. Hardware trojan detection and isolation using current integration and localized current analysis. In *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems* (Oct 2008), pp. 87–95.

[137] WANG, X., TEHRANIPOOR, M., AND PLUSQUELLIC, J. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (June 2008), pp. 15–19.

[138] WEI, S., MEGUERDICHIAN, S., AND POTKONJAK, M. Gate-level characterization: Foundations and hardware security applications. In *Design Automation Conference* (June 2010), pp. 222–227.

[139] WEI, S., AND POTKONJAK, M. Self-consistency and consistency-based detection and diagnosis of malicious circuitry. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 22*, 9 (Sept 2014), 1845–1853.

[140] WILCOX, I., SAQIB, F., AND PLUSQUELLIC, J. Gds-ii trojan detection using multiple supply pad vdd and gnd iddqs in asic functional units. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (May 2015), pp. 144–150.

[141] WIRTH, G. Bulk built in current sensors for single event transient detection in deep-submicron technologies. *Microelectronics Reliability 48*, 5 (2008), 710–715.

[142] WIRTH, G., AND FAYOMI, C. The bulk built in current sensor approach for single event transient detection. In *Proc. International Symposium on System-on-Chip (ISSOC'07)* (2007), pp. 1–4.

[143] WOLD, K., AND TAN, C. H. Analysis and enhancement of random number generator in fpga based on oscillator rings. In *2008 International Conference on Reconfigurable Computing and FPGAs* (Dec 2008), pp. 385–390.

[144] XIAO, K., FORTE, D., AND TEHRANIPOOR, M. A novel built-in self-authentication technique to prevent inserting hardware trojans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 33*, 12 (Dec 2014), 1778–1791.

[145] XIE, H., LI, H., AND XU, G. Hardware trojan prevention based on fully homomorphic encryption. In *2015 IEEE International Conference on Information and Automation* (Aug 2015), pp. 1104–1109.

[146] ZARRINCHIAN, G., AND ZAMANI, M. S. Latch-based structure: A high resolution and self-reference technique for hardware trojan detection. *IEEE Transactions on Computers 66*, 1 (Jan 2017), 100–113.

[147] ZHANG, J., YU, H., AND XU, Q. Htoutlier: Hardware trojan detection with side-channel signature outlier identification. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust* (June 2012), pp. 55–58.

[148] ZHANG, L., AND CHANG, C. H. Hardware trojan detection with linear regression based gate-level characterization. In *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (Nov 2014), pp. 256–259.

[149] ZHANG, X., AND TEHRANIPOOR, M. Ron: An on-chip ring oscillator network for hardware trojan detection. In *2011 Design, Automation Test in Europe* (March 2011), pp. 1–6.

[150] ZHANG, Z., ET AL. A new bulk built-in current sensing circuit for single-event transient detection. In *Proc. Canadian Conference on Electrical and Computer Engineering (CCECE'10)* (2010), pp. 1–4.

[151] ZHANG, Z., ET AL. A bulk built-in voltage sensor to detect physical location of single-event transients. *Journal of Electronic Testing, Theory and Applications (JETTA) 29*, 2 (2013), 249–253.

[152] ZHOU, B., ADATO, R., ZANGENEH, M., YANG, T., UYAR, A., GOLDBERG, B., UNLU, S., AND JOSHI, A. Detecting hardware trojans using backside optical imaging of embedded watermarks. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)* (June 2015), pp. 1–6.

[153] ZIAD, M. T. I., ALANWAR, A., ALKABANI, Y., EL-KHARASHI, M. W., AND BEDOUR, H. Homomorphic data isolation for hardware trojan protection. In *2015 IEEE Computer Society Annual Symposium on VLSI* (July 2015), pp. 131–136.

Techniques de Test Pour la Détection de Chevaux de Troie Matériels en Circuits Intégrés de Systèmes Sécurisés

Résumé – La mondialisation et la déverticalisation des métiers du semi-conducteur a mené cette industrie à sous-traiter certaines étapes de conception et souvent la totalité de la fabrication. Au cours de ces étapes, les circuits intégrés (CIs) sont vulnérables à des altérations malignes : les chevaux de Troie matériels (HTs). Dans les applications sécuritaires, il est important de garantir que les circuits intégrés utilisés ne soient pas altérés par de tels dispositifs. Afin d'offrir un niveau de confiance élevé dans ces circuits, il est nécessaire de développer de nouvelles techniques de test pour détecter les HTs, aussi légers et furtifs soient-ils. Cette thèse étudie les menaces et propose deux approches originales de test post-fabrication pour détecter des HTs implantés après synthèse. La première technique exploite des capteurs de courant incorporés au substrat (BBICS), originalement conçus pour identifier les défauts transitoires dans les CIs. Dans notre cas, ils fournissent une signature numérique obtenue par analyse statistique permettant de détecter tout éventuel HT, même au niveau dopant. La deuxième proposition est une méthode non intrusive pour détecter les HTs dans les circuits asynchrones. Cette technique utilise la plateforme de test du circuit et ne requiert aucun matériel supplémentaire. Elle permet la détection de HTs dont la surface est inférieure à 1% de celle du circuit. Les méthodes et les techniques mises au point dans cette thèse contribuent donc à réduire la vulnérabilité des CIs aux HTs soit par adjonction d'un capteur (BBICS), soit en exploitant les mécanismes de test s'il s'agit de circuits asynchrones.

Mots clés – Chevaux de Troie matériels, conception pour le test & la sécurité, analyse par canaux-cachés, défauts transitoires

Testing Techniques for Detection of Hardware Trojans in Integrated Circuits of Trusted Systems

Abstract – The world globalization has led the semiconductor industry to outsource design and fabrication phases, making integrated circuits (ICs) potentially more vulnerable to malicious modifications at design or fabrication time: the hardware Trojans (HTs). New efficient testing techniques are thus required to disclose potential slight and stealth HTs, and to ensure trusted devices. This thesis studies possible threats and proposes two new post-silicon testing techniques able to detect HTs implanted after the generation of the IC netlist. The first proposed technique exploits bulk built-in current sensors (BBICS) – which are originally designed to identify transient faults in ICs – by using them as testing mechanisms that provide statistically-comparable digital signatures of the devices under test. With only 16 IC samples, the testing technique can detect dopant-level Trojans of zero-area overhead.The second proposition is a non-intrusive technique for detection of gate-level HTs in asynchronous circuits. With this technique, neither additional hardware nor alterations on the original test set-up are required to detect Trojans smaller than 1% of the original circuit. The studies and techniques devised in this thesis contribute to reduce the IC vulnerability to HT, reusing testing mechanisms and keeping security features of original devices.

Keywords – Hardware Trojans, design for test & security, side-channel analysis, transient faults.

Thèse préparée au laboratoire TIMA (Techniques de l'Informatique et de la Microélectronique pour l'Architecture des ordinateurs), 46 Avenue Félix Viallet, 38031, Grenoble Cedex, France

ISBN: 978-2-11-129237-6