Extraction of intrinsic structure for Hardware Trojan detection

Maxime Lecomte, Jacques Jean-Alain Fournier, Philippe Maurine

Extraction of intrinsic structure for Hardware Trojan detection

Maxime Lecomte
lecomte.maxime@cea.fr

Jacques J.A. Fournier
jacques.fournier@cea.fr

Philippe Maurine
philippe.maurine@lirmm.fr

September 18, 2015

Abstract

In this paper we present a novel solution to address the problem of potential malicious circuitry on FPGA. This method is based on an a technique of structure extraction which consider the infection of an all lot. This structure is related to the design (place and route, power grid...) of the integrated circuits which composes the lot. In case of additional circuitry this design will be modify and the extracted structure will be affected. After developing the extraction techniques we present a methodology to insert detection of hardware trojan and counterfeit in different IC manufacturing steps. At last an application example using 30 FPGA boards validate our extraction method. Finally, statistical tools are then applied on the experimental results to distinguish a genuine lot from an infected one and confirm the potential of detection the extracted structure.

1 Introduction

Due to the recent trend of outsourcing integrated circuit (IC) manufacturing and design, structural integrity verification of ICs has become a hot topic. From the specification step to that of packaging, and especially during the design step, a circuit can be corrupted by a malicious adversary. This malicious alteration of the IC structure is called a Hardware Trojan (HT) insertion and its effects can range from performance degradation (e.g. denial of service) up to more sophisticated functionalities (memory dumping etc) [21]. Similarly with the multiplication of foundries and IC vendors, counterfeits are spreading rapidly, ranging from simple copies to a complete replacement of an given IC by one of lower quality.

A HT is composed of two parts: the trigger and the payload. The trigger is the mechanism that scans some signals within the IC until a specific condition

is met. When this condition is met, the payload is activated. The trigger can either be generated externally (external signals or physicals condition) or internally (a special internal state, data, etc). Moreover the trigger can either be combinational (result of a logical operation) or sequential (related to a succession of states). The payload is the 'malicious' effect of the HT. It can either be explicit where of signals are directly added, removed or deactivated. It can also be implicit where the effect cannot be directly observed like, for example, adding side-channel information in the power consumption. The detection of a HT before its activation is a difficult task and it still remains a challenging problem even after its activation when the payload is implicit.

The probability of triggering a HT during functional tests is low and testing is an expensive approach to that end. Moreover, inspecting the circuit through reverse engineering is an expensive process in terms of cost and time and can be destructive. This solution can therefore be applied to only few devices, even though latest imaging based methods have proven to offer a simpler and faster alternatives [9].

Several non destructive methods for HT detection have been recently proposed. The first proposed approaches analyze, using statistical techniques, the overall consumption of an IC to detect the impact of the HT. For example, in [2], a detection technique based on the Karhunen-Loève theorem is proposed in order to detect the power consumption of the HT within process variations and noise. However, this paper only reports validations obtained by simulations, omitting things like the measurement noise. Moreover, the technique may not scale to complex Systems on Chip (SoC).

Hence, in order to enhance the detection capabilities, techniques have been proposed in [14] to analyze locally the propagation delays of logical paths with embedded monitors. However, once again, only simulation results are provided. Another method has been proposed in [1] to integrate a hardware system allowing to monitor important nets of ICs. However, little information is given in this paper about the efficiency of the proposed technique. In parallel to this approach, a test solution was also proposed in [8] in order to trigger easily a HT or at least to increase their electrical activity.

In [16], a first attempt to suppress process variations from the HT detection problem has been proposed based on the strong correlation between the maximum operating frequency of ICs, F_{max}, and their dynamic power consumption. This approach however faces the difficult problem of measuring F_{max} on SoCs [5, 17].

Then in 2011, the use of Ring Oscillators (RO) has been proposed to detect HTs. For example, in [13] the authors propose an analysis of RO sensitivity to the presence of HT but conclude that it seems difficult to detect really small HT. In parallel [24] proposes the use of an array of RO, used in conjunction with a Principal Component Analysis (PCA) [10], to distinguish infected ICs from genuine ones. This proposal has been experimentally validated on an FPGA using a Digital Sampling Oscilloscope (DSO) (thus off-chip) to measure the oscillating frequency of ROs. This idea has then be applied to design an ASIC in 2012 [11] where the initial results were only partly validated. This is may

due to the use of an embedded 8-bit counter as RO, which reduces the accuracy of the measured frequencies.

Later in 2013, the authors in [6] propose to cluster, during the design step, the power grid in several voltage islands embedding each a dedicated sensor to enhance the detection capability. However, no experimental results are given in this paper, neither on the improvements obtained nor on the cost in terms of Silicon area. In 2014, [19] describes a method based on the use of near field electromagnetic cartography. Yet, the authors conclude that it seems difficult to detect all the HT. However, in [3], a more efficient technique is used to interpret the EM traces. As a result, authors conclude it is possible to detect really small HTs but with a special care to control the temperature during measurements. Finally, in [23] the authors analyze EM emanations from FPGAs and succeeds to differentiate a genuine population from an infected one.

Based on those results, the on-chip monitoring solutions seem relevant in term of efficiency since the obtained detections rates are higher than for off-chip methods. Furthermore, these solutions seem industrially viable since the cost of the equipments, dedicated to the data acquisition, is reduced as the tests can easily be done in parallel. For those reasons, the work presented here is based on on-chip methods.

In this paper, we consider that the infection of a single device is not realistic based on the current life cycle of ICs (mainly linked to production and distribution constraints). That why the HT detection methodology proposed in this paper does not aim at establishing if an IC is infected but aims at checking the integrity of a whole production lot. Moreover, this approach also allows to determine if a given IC is a counterfeit (apart from the case of a really high quality copy). The principle of the proposed methodology is to detect, thanks to a embedded sensor network, an alteration in the design's structure induced by the presence of a HT, or by a modification in its place and route or in its floor-plan. These alterations modify the IC power distribution and in particular the static voltage drops [15] in the glue logic and hence that in the sensor array. This method is based on a novel variation model of the performance of CMOS structures in real designs (not in test chips dedicated to the fine measurement of the intra and inter die variations), model which is introduced and validated in this paper.

The paper is organized as follows. First, section 2 describes the threats to IC integrity and specifies which ones are covered in this paper. Then section 3 introduces the proposed variation model and defines the basic principle of our detection methods. Section 4 details the HT and counterfeited detection methodology and section 5 describes the experimental results which validate the proposed approach and the proposed variation model on a set of 30 FPGAs. Finally, the wide range of perspectives generated by our approach is discussed, as a conclusion, in section 6.

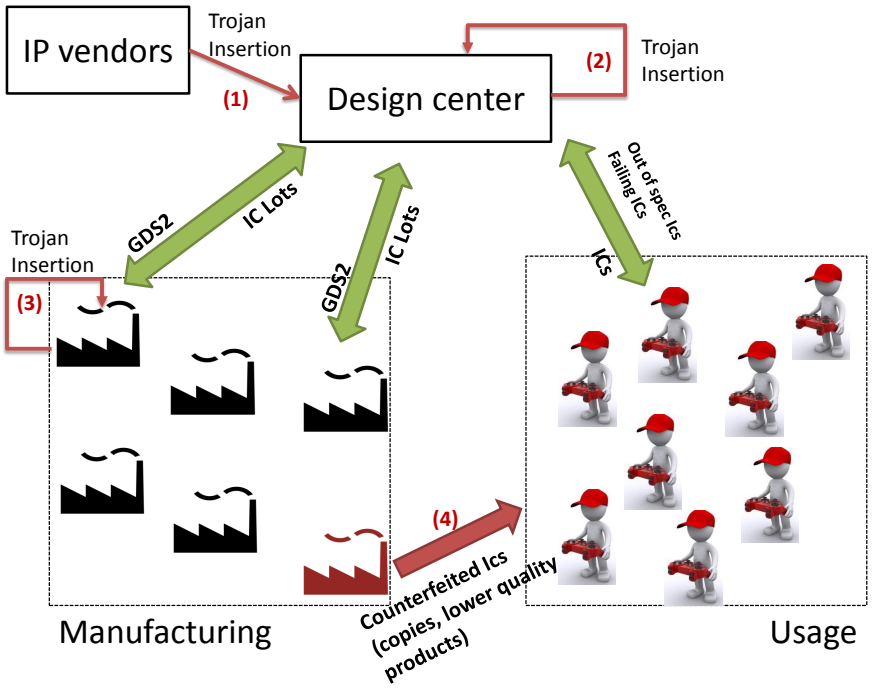

Figure 1: Mains threats to IC integrity

2 Threats and characteristics of infected devices and counterfeits

First, this section recalls briefly the main threats to IC integrity. Second, the HT and counterfeit detection method is described specifying which threats it covers.

2.1 Threats to IC integrity

Fig. 1 summarizes the different steps, from design to exploitation, for manufacturing an IC and the associated threats. The first vulnerabilities are at the design stage. A corrupted piece of hardware IP can be introduced into the product (Threat (1) in Fig. 1) or a compromised employee from the design center can introduce a HT into the HDL description (Threat (2) in Fig. 1). It is difficult to protect against such threats, but some solutions based on ad-hoc design and verification methods have been proposed [7, 18, 22].

The second vulnerable stage is the manufacturing (Threat (3) in Fig. 1). For example, filler cells can be substituted by logic gates, inducing a denial of service or more complex functionalities, or a fuse can be disabled etc. A last threat is that of counterfeits. It consists in selling second hand products, lower quality devices or functional copies directly onto the market causing financial

losses (Threat (4) in Fig. 1). Some can even be almost perfect copies which are difficult to detect.

2.2 Features of infected circuits or counterfeits

Many methods have been proposed to detect HTs. Among them, a large majority aims at detecting the parasitic electrical activity (additional power consumption) generated by the HT's trigger. However, this parasitic electrical activity is not the only measurable trace left by the HTs. Another one is the alteration of the inner structure of the IC. For example, the HT insertion modifies the local and global capacitance and resistance of the power and ground nets. This modification induces a different current flow in the IC, and thus a different static or dynamic voltage distribution (static or dynamic voltages drops).

In the same way, counterfeits are characterized by a more (functional copies) or less (almost exact hardware copy) different physical structures, and therefore by a different repartition of the voltage across the IC's surface.

3 Principle of HT and counterfeit detection

Our detection method is based upon a simple principle: a fingerprinting of the supply voltage on the surface of IC at rest (i.e. just powered on with the clock active). In order to do this, a network of sensors is uniformly spread over the whole IC surface to get a cartography of the inner supply voltage. Any sensor sensitive to the supply voltage, Vdd, can be used. In the experiments reported in section 5, Ring Oscillators (RO) are used. Given that the frequency f of a RO is sensitive to the local Vdd value, the distribution of measured values for f above the IC surface, in the absence of any process variation (P), is a direct picture of the Vdd distribution. Hence in our approach, we have to get rid of the effect of intra-die and inter-die process variations. With this approach, we shall be able to mitigate risks linked the introduction of HTs at the manufacturing stage.

3.1 Process variation model and performance variation model of CMOS structures

Given p, an inherent parameter of the IC fabrication technology, the impact of the manufacturing process variations, the so called process variations, are generally described as follows:

$$p = \bar{p} + \Delta p_{inter} + \Delta p_{intra} \tag{1}$$

with $\bar{p}$ being the mean (or typical) value of the parameter on a whole lot of a production, $\Delta p_{inter} \sim N(0, \sigma^2_{inter})$ the effect of the inter-die variations assumed normal and $\Delta p_{intra} \sim N(0, \sigma^2_{intra})$ the impact of variations within a die, i.e. the intra-die process variations also assumed normal.

This process variation model is well known and widely adopted to simulate the effect of process variations on the parameter p of an IC (a transistor, a resistance, a pn junction, etc). However, the extraction of the standard deviation values σ_{intra} and σ_{intra} is generally performed on dedicated ICs (regular arrays of MOS transistors [12] or SRAM cells [4]) which are quite uniform relative to their physical structures and under controlled voltage and temperature. This process variation model does not take into account the impact of the IC's physical structure (power supply routing, local transistor density, etc) on the CMOS gate performance or on that of an embedded sensor which, of course, depends on all process variations through equation (1). Hence for our case, we shall use the following variation model for the output value $T(x_i, y_i)$ of a sensor i located at (x_i, y_i) on the IC's surface:

$$T(x_i, y_i) = \bar{T} + \Delta T_{inter} + \Delta T_{intra} + \Delta T(x_i, y_i) \tag{2}$$

where $\Delta T(x, y)$ is a deterministic value which depends of the sensor's position over the IC, and which models the impact of the IC's structure on the performance of the sensor. To ease the reading, $T(x_i, y_i)$ and $\Delta T(x_i, y_i)$ shall be noted T_i and ΔT_i respectively, showing that the variation model considered in this paper is a spatial model.

3.2 Fingerprinting the IC's structure

Considering the model given by equation (2), fingerprinting the structure of a design featuring a network of q sensors regularly spread on its surface is relatively simple for a same manufacturing lot of ICs. The q values of ΔT_i are calculated by averaging the impact of the process variations on m_{lot} devices of the same lot:

$$\widehat{\Delta T_i} = \frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} T_i^j - \bar{T} = \frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} \Delta T_i^j \tag{3}$$

$$\sigma_{\Delta T_i} = \sqrt{\frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} (\Delta T_i^j - \widehat{\Delta T_i})^2} \tag{4}$$

where:

$$\bar{T} = \frac{1}{m_{lot} \cdot q} \cdot \sum_{j=1}^{m_{lot}} \sum_{i=1}^{q} T_i^j \tag{5}$$

T_i^j is the measurement of the output of the sensor i of the device $j \in \{1, \ldots, m_{lot}\}$ of the considered lot.

With these notations, the vector $S^{Design} = [s_1^{Design}, \ldots, s_{2q}^{Design}]$ can be defined as follows:

$$S^{Design} = [\widehat{\Delta T_1}, \ldots, \widehat{\Delta T_q}, \sigma_{\Delta T_1}, \ldots, \sigma_{\Delta T_q}] \tag{6}$$

S^{Design} represents the fingerprint of the physical structure of an IC called 'Design' and is by construction independent of the process variations. This fingerprint is the base of the HT and counterfeit detection methods proposed in section 4.

4 Detection Methodology

The starting point of our methodology is the addition of a network of sensors sensitive to the supply voltage. Those sensors are placed so as to cover most of the IC's surface. The granularity, i.e. the distance between two sensors, is chosen by the IC designer depending on the trade-off between detection capability and cost.

When the first run or the test run (which are less likely to be infected) is received, the integrity of some devices is verified to qualify the whole lot. This could be done by applying reverse engineering methods or by using optical based methods [20]. Once the first production lot is qualified, the signature (see eq. (6)) of the design is calculated using equations (3) and (4). This fingerprint constitutes the reference fingerprint for the considered design.

The designer will then usually order other runs ("production runs") from the same foundry or from another one which offers the same technology node. Once those new production lots are received, their corresponding fingerprints are calculated and are 'compared' with the reference one in order to verify that the newly received lots have not been corrupted. Ageing techniques could be applied to this reference lot to derive fingerprints of the design at different ages.

In the same way, at some (later) point in time, the designer can have 'field returns' which could contain counterfeits. With the reference fingerprint, the origin of these devices can be verified without application of expensive, complex and destructive methods. In order to do that, the designer extracts the fingerprint of the suspected device and compares it with the reference fingerprint to finally get a probability that the device is a genuine one. If the probability is too low, complementary analyses (like reverse engineering) can be applied.

The above procedures require the comparison of the reference fingerprint with that of a new production lot in order to detect the eventual presence of an HT (case 1). The procedures also require the comparison of the reference fingerprint $\mathrm{S}^{\mathrm{REF}}$ with the fingerprint of a single device in order to detect counterfeits (case 2).

4.1 Case 1:HT detection

When the integrity of a new lot of devices has to be checked, the first step is to calculate its fingerprint $\mathrm{S}^{\mathrm{NewRun}}$. Since this signature shall be, in practice, calculated using a high number of devices (> 100), the estimate of means can be considered as reliable. It is therefore possible to apply distinguishers or a statistical tools working on the means, like the Difference of Means (DoM) or the T-test (and more precisely the Welch's test). This implies using the $\widehat{\Delta T_i}$s

of the signatures (eq. (6)) and to analyze the values

$$\widehat{\Delta S_i} = S_i^{Ref} - S_i^{NewRun} \tag{7}$$

for $i = \{1, \ldots, q\}$, or that of

$$T\_Stat_i = \frac{S_i^{Ref} - S_i^{NewRun}}{\sqrt{\frac{(s_{q+i}^{Ref})^2}{m_{lot}} - \frac{(s_{q+i}^{NewRun})^2}{m_{lot}}}}. \tag{8}$$

It is also possible to use other statical tools such as the Kolmogrow-Smirnov test (KS) to determine if the distributions for all $j \in \{1, \ldots, m_{lot}\}$ of $(\mathrm{T_i^j} - \bar{\mathrm{T}})$ for the reference lot and for the new one are drawn from the same distribution law, i.e. if the devices embed the same design or not.

With the same idea of comparing centered populations of $(\mathrm{T_i^j} - \bar{\mathrm{T}})$, clustering solutions can be used such as the k-means (with k = 2) or using simply the median. To do this, the populations $(\mathrm{T_i^j} - \bar{\mathrm{T}})$ of the reference lot and of the new lot are merged within the same set. Then, the k-means is applied on the resulting population (or the median is calculated) so that to split it into two clusters. Finally, the number of devices in each cluster coming from the reference lot and from the new one are counted. If the tested ICs are genuine, the two resulting clusters have the same number of elements from the reference lot and that of the new lot. More precisely, if the cardinals of the reference lot and that of the new lot are equal, 50% of the devices from the reference lot and 50% of the devices from the new lot have to constitute each cluster. If this is not the case, the new lot can be considered as different from the reference one.

4.2 Case 2: counterfeit detection

The case of the suspected 'field return' is more difficult to treat as the fingerprint as described so far for HT detection cannot be calculated on one single device: we only have the $\mathrm{T_i^{Suspected}}$ of the considered device. In this case, we first "recenter" all the values using the value $\bar{\mathrm{T}}^{\mathrm{Suspected}}$ of the suspected IC (i.e. calculate $\mathrm{T_i^{Suspected}} - \bar{\mathrm{T}}^{\mathrm{Suspected}}$) and then calculate the probabilities that each $\mathrm{T_i^{Suspected}}$ values comes from the normal distribution:

$$N(0, (s_{q+i}^{Suspected})^2) = N(0, \sigma^2_{\widehat{\Delta T_i}}) \tag{9}$$

$\sigma_{\widehat{\Delta T_i}}$ is indeed the standard deviation of the sensor i, value estimated with the ICs from the reference lot. The probabilities for all sensors are then combined (and more precisely a multi-normal distribution is defined with all $\sigma_{\widehat{\Delta T_i}}$) to obtain the probability that the considered device is a genuine one.

5 Experimental results

The HT detection methodology has been experimentally tested on a set of 30 FPGA boards featuring a Xilinx Spartan3E-1600. 15 boards have been used

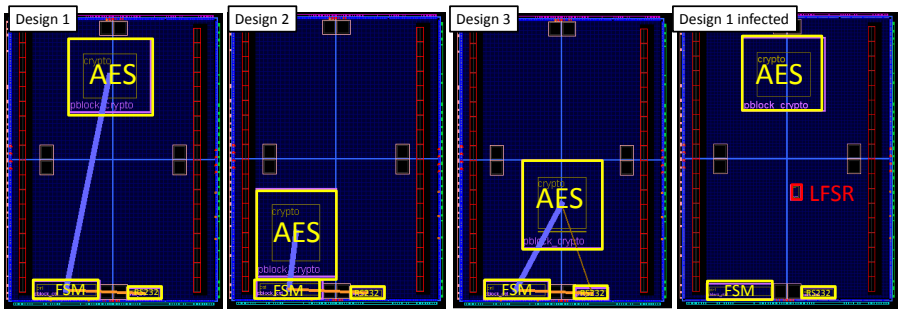

Figure 2: From left to right: 3 implementations of the same HDL-code without any HT. The rightmost picture is an infected (LFSR) version of Design 1.

as genuine ICs and 15 have been used in order to emulate an infected lot or counterfeited devices. Validation of our methodology using FPGAs is a first step before entering the long and costly process of designing an ASIC.

5.1 Experimental protocol

On each Spartan-3E-1600 FPGA, a 128-bit-key AES, an RS232 communication block and a Finite State Machine (FSM) have been placed and routed. An array of 60 ROs has been added to the design. Each RO is coupled with a clock divider by two so as to be able to observe and measure precisely the 60 frequencies on an IO pad through a multiplexer. The area overhead incurred by the addition of our on-chip detection hardware is about 3.2% of the FPGA resources. The frequency measurements are performed with an oscilloscope from Lecroy featuring a 4GHz bandwidth and a 40GS/s sampling rate.

In order to obtain accurate measurements (accuracy of $\pm 0.025ps$), each frequency estimation is done by measuring the duration equivalent to 100 periods and by repeating this experiment 100 times to obtain a mean value of the period of each RO: T_i^j. During these measurements, the IC is kept inactive, i.e. just powered on and with the clock running. The time spent to measure the 60 values T_i^j on a board is lower than $30min$ which is short enough to consider the temperature as constant in a laboratory environment. In order to guarantee a good stability of the supply voltage, the FPGA is powered by a stabilized dc supply source with an accuracy of 0.05%.

To emulate the effect of a HT, a 64-bit LFSR is used. It occupies an area of 48 slices which represents 0.32% of the FPGA's surface (see the rightmost picture of Fig.2). Note that the AES alone is mapped onto 1778 slices. The LFSR is clocked at $50MHz$ by taking the clock input of a D Flip-Flop of the AES. This HT can therefore be considered as a sequential HT.

To emulate counterfeits, several constrained place and route steps of the design are performed. Fig. 2 gives 3 different floorplans of the same HDL code (three leftmost pictures). One of them (Design 1) is considered as the original / genuine design, the two others as counterfeits.

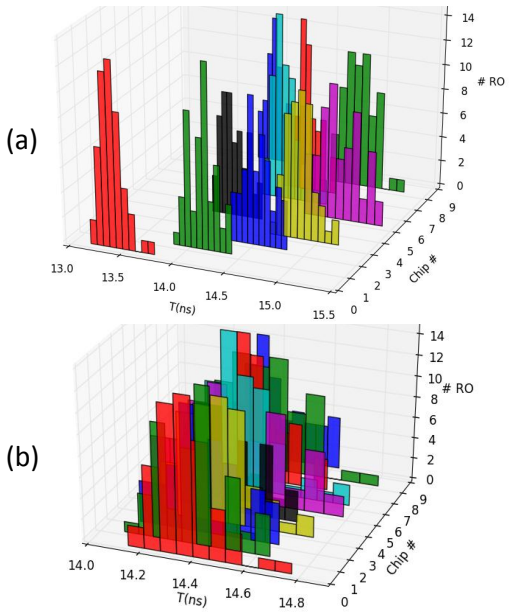

Figure 3: Impact of inter-die (a) and intra-die (b) variations

5.2 Estimations of σ^2_{inter} and σ^2_{intra}

Before evaluating the relevance of the proposed methodology and thus the relevance of the proposed variation model of CMOS structures, the impact of the inter and intra-die variations have been estimated on the 30 FPGA boards. Fig. 3(a) gives the histogram of the periods (T_i^j) for the 60 ROs on ten boards, i.e. ten different devices. One can observe that the mean period for each device varies from $13.5ns$ to $14.5ns$. The impact of inter-die variations is therefore of the order of several hundreds ps. Hence, using data from 30 boards, we estimate that the inter-die impact, can be modelled by the following normal distribution $N(0, \sigma^2_{inter})$ with $\sigma^2_{inter} = 460ps$.

Fig. 3(b) shows the histograms for the measured periods (centred for each board) for the 60 ROs values and this for the same ten boards. On can observe that the intra-die variations have an impact of the order of a hundred ps. Based on those assumptions, the intra-die variations can be considered to follow a zero-mean normal distribution and can be modeled by $N(0, \sigma^2_{intra})$ with $\sigma_{intra} = 130ps$.

5.3 Validation of our variation model and Counterfeit detection

In section 3, we introduced a variation model for the performance of CMOS structures and for that of sensors. This model is novel as it introduces a deter-

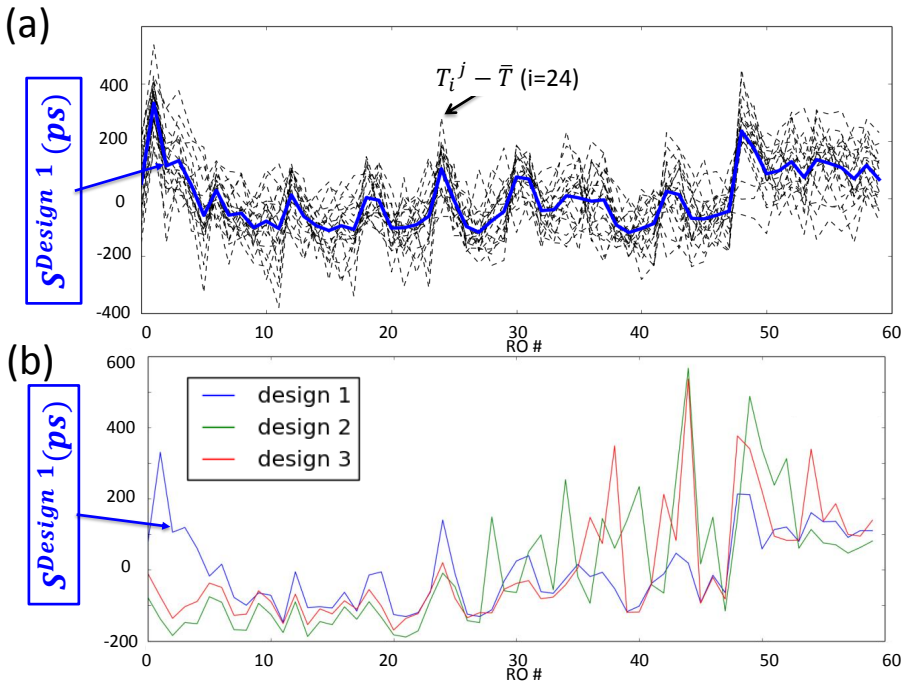

Figure 4: (a) $S^{Design1}$ in blue and the 15 specific fingerprints associated to the the 15 boards used to compute $S^{Design1}$ (b) Fingerprints of the three designs reported Fig. 2

ministic term which expresses the impact of the design structure on the sensor and particularly the impact of the power distribution. This novel model being the base of the proposed detection method, we start by evaluating its relevance.

In order to do that, the frequency of the 60 ROs from the 3 leftmost designs shown in Fig. 2 have been measured on 15 boards. Then, the first half of their fingerprints S^{Design} (i.e the $\widehat{\Delta T_i}$s) have been compared. Fig. 4(a) shows $S^{Design1}$ (dark curve) and the unique fingerprints (dotted curves) for each of the 15 devices used to compute $S^{Design1}$. Fig. 4(b) shows the three fingerprints obtained from the three designs. One can observe they are significantly different despite the use of the same 15 boards. One can also observe that the designs 2 and 3 have significantly different fingerprints (which are in turn different from that of design 1) even though the two floorplans are relatively similar. Note that in Fig. 6(a) (dotted curve), we see that the signatures corresponding to the same design obtained from two different lots of 15 FPGA are the same.

For these designs 2 and 3, the sensors 30 to 45, located in the neighborhood or in the AES (see Fig. 2), are characterized by high $s_i = \Delta T_i$ values. For the design 1, the AES is around sensors 1 to 10 (see Fig. 4(b)). This reinforces the hypothesis that the floorplan influences the sensors by modifying locally the

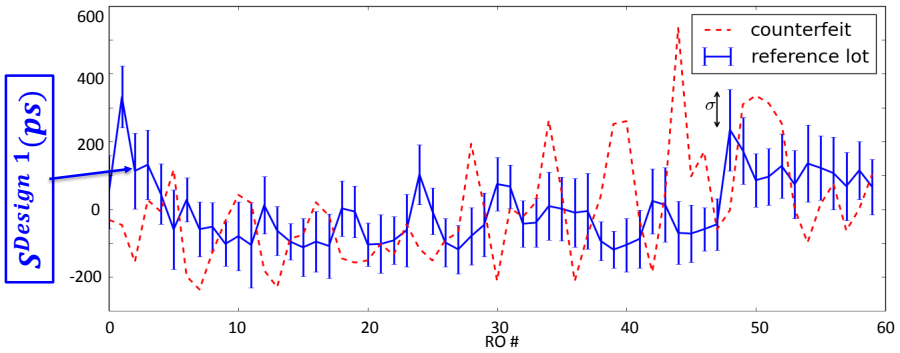

Figure 5: $S^{Design1}$ and signature of a counterfeit

distribution of supply voltage even when the IC does not compute anything.

At this point, we see that our variation model of CMOS structures is valid and allows to distinguish different floorplans of a same HDL code. To show that our method can determine whether a suspected IC is a counterfeit (compared to a reference lot), we shall refer to Fig. 5 which shows the complete fingerprint (calculated from 15 samples) of the design 1 (dark curve), i.e. the values of $\widehat{\Delta T_i} \pm \sigma_{\widehat{\Delta T_i}}$, and the fingerprint of a suspected device (dotted line). In this case, there are visually no doubt that the considered device is a counterfeit. For example, the ROs 30, 39 and 40 are out of the $\pm 3 \cdot \sigma_{\widehat{\Delta T_i}}$ measured on the reference lot. It is the same for the ROs 1 and 44.

5.4 HT detection

The detection method of an infected lot is similar to that of a counterfeit lot, although the alteration of the physical structure is expected to be significantly smaller and localized. Fig. 6 shows the results obtained by applying the DoM (upper picture) and the T-test (lower picture) in order to verify the integrity of the 15 infected and 15 genuine ICs with the 15 reference ICs. 30 boards are used. To emulate the infection (the presence of a sequential HT), a 64-bit LFSR (48 slices) has been added to the design 1. Both the DoM and the T-test allow to detect an anomaly located around RO 33 which is effectively close to the LFSR. Moreover, the DoM stays low between the reference and the genuine lots. In this case, the absolute T-values ($\mid T\_Stat_i \mid$) does not exceed, 1.22 for $i \in \{1, \ldots, 60\}$. Genuine lots are therefore recognized as uninfected lots. Similar results have been obtained with the k-means (Fig. 8) and the median (Fig. 7). These results validate the proposed detection methodology and above all the proposed variation model of the performance of a CMOS structure in a real design which strongly depends on the power distribution in advanced CMOS technologies.

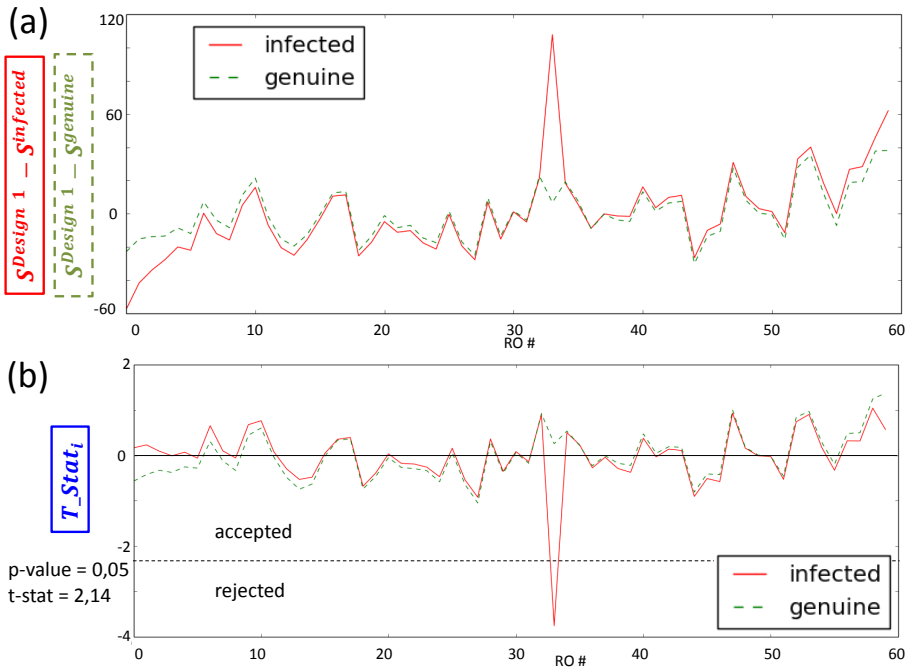

Figure 6: Difference between the fingerprints of 15 infected and 15 genuine devices with the signature of the 15 reference devices, (a) DoM, (b) T-test

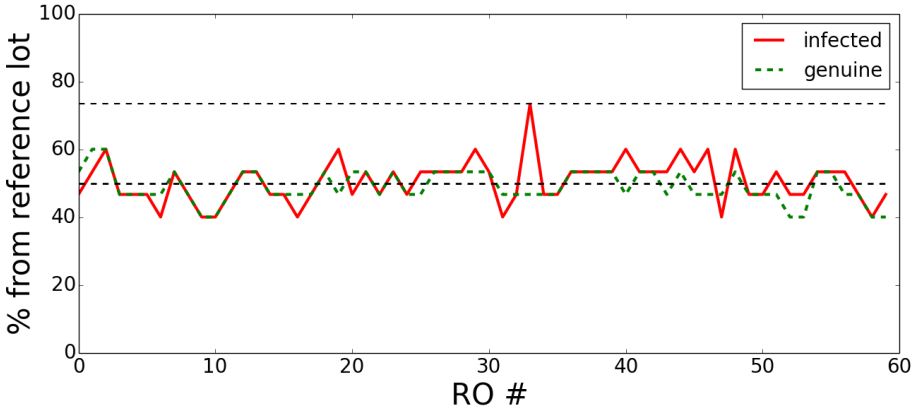

Figure 7: Percentage of a cluster (made by median) that come form the reference lot

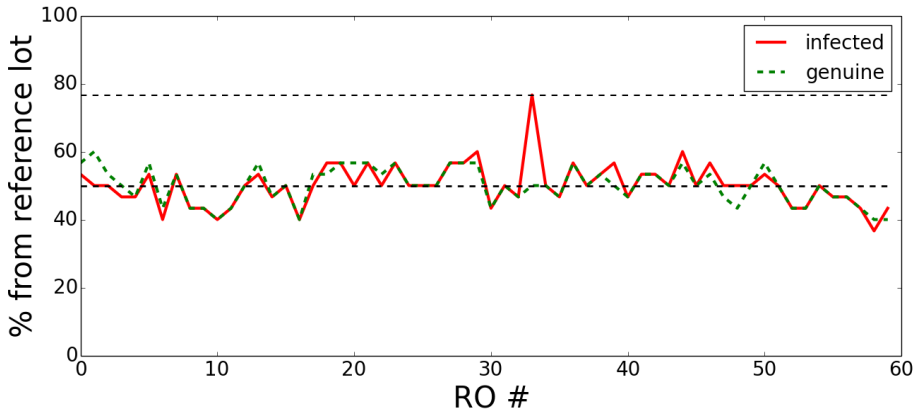

Figure 8: Percentage of a cluster (made by kmeans) that come form the reference lot

6 Conclusion and perspectives

This paper introduces an efficient and practical approach for detecting Hardware Trojans in ICs and counterfeits based on a new variation model for the performance of CMOS structures. This model gets rid of the process variation issues usually met when doing HT detection in practice. The model is actually built from measurements made on 'real circuits' and not designs usually dedicated to the monitoring of process variations, the latter being much more uniform and regular in terms of physical structure and content. This approach is based on the assumption that IC infection is more likely to be done on a lot level than on individual isolated ICs.

A reference signature is first derived from a trusted lot (which could correspond in practice to a first run usually done for characterisation purposes). Then the same signatures will be calculated on subsequent (suspected) lots and compared with the reference one, using the different statistical methods described in this paper. For counterfeit detection, we propose a way of calculating this IC fingerprint and comparing it with the reference signature. The model and the methods have been successfully experimented on a set of 30 FPGA boards as an initial validation strategy.

The next steps will be to implement the on-chip sensor structures on an ASIC design and validate the method on large set of ICs. Moreover, other factors like the size of the HT, the density of sensors to used or the positioning of those sensors shall be investigated.

References

[1] Miron Abramovici and Paul Bradley. Integrated circuit security: New threats and solutions. In *Proceedings of CSIIRW 2009*, pages 55:1–55:3.

ACM.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using ic fingerprinting. In *Security and Privacy, 2007. SP '07. IEEE Symposium*, pages 296–310, May 2007.

[3] J Balasch, B Gierlichs, and I Verbauwhede. Electromagnetic circuit fingerprints for hardware trojan detection. *EMC 2015, IEEE*, 2015.

[4] A. Bhavnagarwala, S. Kosonocky, C. Radens, K. Stawiasz, R. Mann, Qiuyi Ye, and Ken Chin. Fluctuation limits amp; scaling opportunities for cmos sram cells. In *IEDM 2005*, pages 659–662, Dec 2005.

[5] K. Bowman, C. Tokunaga, J. Tschanz, A. Raychowdhury, M. Khellah, B. Geuskens, Shih-Lien Lu, P. Aseron, T. Karnik, and V. De. Dynamic variation monitor for measuring the impact of voltage droops on microprocessor clock frequency. In *CICC, 2010 IEEE*, pages 1–4, Sept 2010.

[6] Yuan Cao, Chip-Hong Chang, and Shoushun Chen. Cluster-based distributed active current timer for hardware trojan detection. In *ISCAS, 2013 IEEE International Symposium*, pages 1010–1013, May 2013.

[7] RajatSubhra Chakraborty and Swarup Bhunia. Security against hardware trojan attacks using key-based design obfuscation. *Journal of Electronic Testing*, 27(6):767–785, 2011.

[8] RajatSubhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. Mero: A statistical approach for hardware trojan detection. In Christophe Clavier and Kris Gaj, editors, *CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 396–410. Springer Berlin Heidelberg, 2009.

[9] Franck Courbon, Philippe Loubet-Moundi, Jacques J.A. Fournier, and Assia Tria. A high efficiency hardware trojan detection technique based on fast sem imaging. In *In the proceedings of DATE'15*, March 2015.

[10] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics (4th Edition)*. Pearson, 2011.

[11] Andrew Ferraiuolo, Xuehui Zhang, and Mohammad Tehranipoor. Experimental analysis of a ring oscillator network for hardware trojan detection in a 90nm ASIC. *Proceedings of the ICCAD '12*, page 37, 2012.

[12] Ali Keshavarzi, Gerhard Schrom, Stephen Tang, Sean Ma, Keith Bowman, Sunit Tyagi, Kevin Zhang, Tom Linton, Nagib Hakim, Steven Duvall, John Brews, and Vivek De. Measurements and modeling of intrinsic fluctuations in mosfet threshold voltage. In *Proceedings of ISLPED 2005*, pages 26–29, New York, NY, USA, 2005. ACM.

[13] C. Lamech, R.M. Rad, M. Tehranipoor, and J. Plusquellic. An experimental analysis of power and delay signal-to-noise requirements for detecting trojans and methods for achieving the required detection sensitivities. *Information Forensics and Security, IEEE Transactions on*, 6(3):1170–1179, Sept 2011.

[14] Jie Li and J. Lach. At-speed delay characterization for ic authentication and trojan horse detection. In *HOST 2008*, pages 8–14, June 2008.

[15] Chen-Wei Liu and Yao-Wen Chang. Floorplan and power/ground network co-synthesis for fast design convergence. In *Proceedings of ISPD 2006*, pages 86–93, New York, NY, USA, 2006. ACM.

[16] S. Narasimhan, Dongdong Du, R.S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia. Multiple-parameter side-channel analysis: A non-invasive hardware trojan detection approach. In *HOST 2010*, pages 13–18, June 2010.

[17] Bettina Rebaud, Marc Belleville, Edith Beigné, Christian Bernard, Michel Robert, Philippe Maurine, and Nadine Azémard. Timing slack monitoring under process and environmental variations: Application to a DSP performance optimization. *Microelectronics Journal*, 42(5):718–732, 2011.

[18] Seyed Mohammad Hossein Shekarian and Morteza Saheb Zamani. A trust-driven placement approach: A new perspective on design for hardware trust. *Journal of Circuits, Systems and Computers*, 0(0):1550115, 0.

[19] O. Soll, T. Korak, M. Muehlberghuber, and M. Hutter. Em-based detection of hardware trojans on fpgas. In *HOST 2014*, pages 84–87.

[20] F. Stellari, Peilin Song, and H.A. Ainspan. Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements. In *VTS, 2014 IEEE 32nd*, pages 1–6, 2014.

[21] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. *Design Test of Computers, IEEE*, 27(1):10–25, Jan 2010.

[22] Kan Xiao and M. Tehranipoor. Bisa: Built-in self-authentication for preventing hardware trojan insertion. In *HOST 2013*, pages 45–50, June 2013.

[23] NGO Xuan Thuy, Najm Zakaria, Shivam Bhasin, Guilley Sylvain, and Danger Jean-luc. Method taking into account process dispersions to detect hardware trojan horse by side-channel. June 2015.

[24] Xuehui Zhang and M. Tehranipoor. Ron: An on-chip ring oscillator network for hardware trojan detection. In *DATE 2011*, pages 1–6, March 2011.