HAL
archives-ouvertes.fr

Identification of Hardware Trojans triggering signals

Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre

▶ **To cite this version:**

Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre. Identification of Hardware Trojans triggering signals. First Workshop on Trustworthy Manufacturing and Utilization of Secure Devices, May 2013, Avignon, France. 2013, <http://trudevice.com/Workshop/>. <lirmm-00991360>

HAL Id: lirmm-00991360
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00991360

Submitted on 15 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

cost
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

Identification of Hardware Trojans triggering signals

Giorgio Dinatale, Sophie Dupuis, Marie-Lise Flottes, Bruno Rouzeyre
LIRMM (Université Montpellier II /CNRS UMR 5506)
Montpellier, France

Abstract — ***Hardware Trojans*** **are malicious alterations to a circuit. These modifications can be inserted either during the design phase or during the fabrication process. Due to the diversity of Hardware Trojans (HTs), detecting and/or locating them are challenging tasks. Numerous approaches have been proposed to address this problem. Methods based on logic testing consist in trying to activate potential Hardware Trojans in order to detect erroneous outputs during simulation. However, traditional ATPG testing may not be sufficient to detect Hardware Trojans. Hardware Trojans are indeed stealthy in nature i.e. mostly inactive unless they are triggered by a rare value. The activation of a Hardware Trojan is therefore a major concern. In this paper, we propose a procedure to identify circuit sites where a possible HT may be easily inserted. The selection of the sites is based on the assumption that the HT is triggered (i) by signals that have potential rare values, (ii) in paths that are not critical, and (iii) combining multiple gates that are close one to the other in the circuit's layout, and close to available space. This identification is then used to automatically generate test patterns able to excite these sites.**

Keywords-Hardware Trojan; Hardware Trojan Detection; Hardware Trojan Activation; Logic testing.

I. INTRODUCTION

With ever-shrinking transistor technologies, the cost of new fabrication facilities is becoming prohibitive and outsourcing the fabrication process to low-cost locations has become a major trend in IC industry in the last decade. This raises the question about untrusted foundries in which circuit descriptions can be manipulated with the possible insertion of malicious circuitry or alterations, referred to as Hardware Trojans (HTs) [1]. Besides, recent issues arose from the possibility of getting HTs from untrusted IP vendors [2].

Due to the diversity of HTs, different classifications have been proposed. The proposed classification in [5] is based on the activation mechanism (referred as the *triggering*) and the introduced effect (referred as the *payload*). The triggering logic monitors a set of inputs to activate the payload at the proper event. A taxonomy is also presented in which HTs are classified based on their trigger and payload mechanisms (digital, analog, combinational, sequential…). The focus of our work is on digital, combinational HTs. The fundamental assumption in that case is that the HT activation should occur under very rare conditions i.e. the trigger is attached on nodes with low controllability. In addition, also for reasons of stealthiness, it is often assumed that the payload is attached on nodes with low observability. This is referred in [5] as *rare values* based HTs. A model of this type of HT is presented in Figure 1.

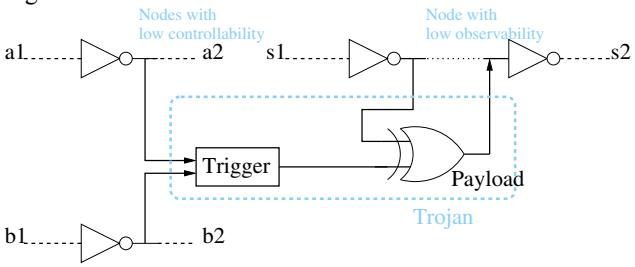

Figure 1. *Rare value* based HT circuit model [5].

HTs detection methods are divided into two categories: methods based on *side-channel analysis* [3, 4], or *logic testing* [5, 6]. In the latter case, if an erroneous behavior of the IC is observed, it can be inferred that a HT has been inserted in the IC. The most important advantage of logic testing is that, as opposed to side channel analysis, it is robust with respect to environment and process variability. It seems therefore more suitable for the detection of small HTs (whose effects can be beyond the threshold of variability). Yet, traditional ATPG test vectors are not sufficient to detect HTs. The assumption is indeed that an attacker will try to hide the HT of ICs' functional behavior i.e. a HT is mostly inactive and is triggered under very rare conditions. The main concern is therefore to be able to activate potential HTs i.e. to find test vectors that can maximize the chances of triggering potential HTs. *Design for hardware trust* methods exist also. These methods consist in incorporating into the ICs some features that should improve the HT detectability [7, 8].

In this paper, we propose a procedure to identify circuit sites where a possible HT may be easily inserted. The selection of the sites is based on the assumption that the HT is triggered (i) by gates that have potential rare values as proposed in [5, 6], (ii) in paths that are not critical, and (iii) combining multiple gates that are close one to the other in the circuit's layout, and close to available space.

This paper is organized as follows. In Section II, we recall the different proposed logic testing HTs detection methods. In Section III, we present our technique. Finally, Section IV concludes the paper.

II. PRIOR WORK

In order to be able to detect a potential HT by logic testing, the main concern is to be able to activate the HT. The assumption is that a HT has a stealthy nature and is activated under very rare conditions. Based on this

assumption, the HT detection methods aim at optimizing pattern generation techniques in order to maximize the probability of inserted HTs getting activated and therefore detected by logic testing.

The first logic based detection approach is presented by Wolff et al. in [5]. The goal is to find so-called *HT test vectors* i.e. vectors that can detect HTs triggered by rare values. It is assumed that HTs triggered by *non-rare values* should be detected by traditional manufacturing testing. Firstly, a logic simulator is used to find most likely target sites to attach a HT trigger, i.e. low controllability nodes. This results in a set of Q targets for q-input triggers HTs (note that an assumption is be done on the number of trigger inputs). Secondly, a fault simulator is used to identify low observability nodes, which results in a set of targets for payload. From these two sets, the trigger values and frequencies of each possible HT are computed, as well as the input trigger vectors associated with these trigger values. Then, an ATPG tool is used to check whether each vector from the set can be propagated to the circuit output. Simulations were performed on a small set of ISCAS'85 benchmarks. The results show that, assuming a 2-input HT, ATPG vectors are not sufficient for trigger coverage.

In [6], Chakraborty et al. propose also to generate a set of test patterns that is compact (in order to minimize test time) while maximizing the chance to detect a HT. The proposed methodology is Called MERO for Multiple Excitation of Rare Occurrence. The assumption is that the number of times a HT trigger condition is satisfied increases with the number of times the trigger nodes have been individually excited to their rare value. This results then in increasing the probability to trigger the HT. From the circuit netlist, a set of random patterns, a list of rare nodes and the number of times to activate each node to its rare value, the set of patterns is modified so that each node satisfies its rare value the desired number of times. To validate the method, a comparison with random and ATPG patterns has been done for a set of ISCAS'85 and ISCAS'89 benchmarks. The signal probabilities were estimated with a simulation tool (and 100 000 random vectors). 100 000 random instances of HTs have been considered, with 2 or 4 triggers and 1 payload. The conclusions are that the MERO set produces a better HT coverage with a reduced test set.

The identification of nodes with low controllability is also referred in [7]. To avoid simulations, Salmani et al. propose to compute the probabilities of each node to be '0' or '1'. This consists in propagating the probabilities of each node to be '0' or '1' from the inputs to the outputs, probabilities of ½ and ½ being put to the inputs of the circuits. Therefore, when the probabilities become unbalanced for a node, this node has a low controllability.

It must be mentioned firstly that, in all these techniques, all signals are considered individually, while the trigger can be driven by several signals. In other words, rare values signals are not synonymous of rare logic conditions. Thus, the goal of our approach is to identify *sets* of signals that conjunctively may trigger a HT. The number of such subsets being prohibitively large, our technique aims at reducing the search space. Furthermore, the position on the triggering signals on the layout is not taken into account, while it is unrealistic that a HT can be inserted anywhere in the circuit. This is an important parameter of our approach.

III. OUR HT DETECTION APPROACH

Our procedure is divided into two steps. First of all, we select a set of nodes that may be targets for the attacker to attach a HT trigger. Second of all, we generate test vectors that aim at activating a set of q triggers. We extend previous approaches by making the nodes' selection according to three criteria:

- First, the nodes' controllability: nodes with a low controllability are difficult to set to a required logic value and are therefore good candidate for rare value triggering,
- Second, the nodes' slack time i.e. nodes with a positive slack and for which the insertion of a HT does not generate a degradation of delay. The assumption is that the attacker will want to hide the HT from a delay point of view so that it is stealthy. HT detection methods based on delay analysis exist (e.g. [4]), but are limited to HT impacting only critical paths. The idea behind this criterion is then that by inserting the HT into the available slack, these techniques would fail.
- Third, the position of these nodes in the layout i.e. the nodes for which there is enough free space around to insert an extra door without compromising the placement.

Once the sets of nodes are identified, the test vector generation can be done aiming at activating a subset of q nodes in this set. The subset generation is done according once again to the nodes placement in the layout: our assumption is that the different trigger inputs of a HT must be placed close to each other.

The criteria taken in our approach are intended to better reflect the choices that may be done by an attacker in an untrusted foundry i.e. an attacker who has access to the layout and wants to insert a HT as discreet as possible (both from the functional point of view and from the layout point view).

Our first criterion is the controllability of the nodes, as in previous approaches. The nodes that have a very low controllability i.e. that are difficult to set to a required logic value and are therefore good candidate for rare value triggering are selected. Our second criterion is based on the assumption that an attacker wants the HT to be not visible also from a delay point of view, otherwise it would have been detected by delay-testing. Therefore a selection is made of the nodes with a positive slack. Our third criterion is the "free space" around the nodes. In this case, the assumption is that, for an attacker to insert triggering gates connected to a node, there must be free space close to this node so that the gate(s) can be easily inserted. In this way, the placement of gates around is not compromised, which, if necessary, would be much more visible.

A set of nodes that satisfy these three criteria is then obtained. From this set, an assumption has to be done on the number of triggers (e.g. *t*) of potential HTs, and test vectors have to be generated in order to activate each possible subset of *t* triggers. Depending on the number of selected nodes, the set of vectors to be generated can still be very large. Once again, so as to limit the number of vectors and hence, the simulation time, we consider the layout to generate the test vectors: the criterion is the proximity of the nodes to consider. Our assumption is that an attacker will not select nodes that are distant in the layout but nodes close to each other so that the connection of the HT is feasible.

A. Nodes with a low controllability

The first step is to be able to find rarely activated nodes in a circuit. To do that, simulation results can be used. However, an exhaustive simulation is a very time consuming task, and a simulation with only a small portion of input vectors can lead to imprecise results. Using probabilities seems like a good alternative such as presented in [7]. This consists in propagating the probabilities of each node to be '0' or '1' from the inputs to the outputs, probabilities of ½ and ½ being put to the inputs of the circuits (cf. Fig. 2). However, this probability-based method gives correct results as long as the input signals of a gate are independent. In order to manage realistic circuits, this method has to take into account reconvergence, i.e. correlations among input signals of a gate. To do that, the *support-set* of each node is needed i.e. the set of nodes representing a portion of the circuit that has only independent signals as inputs. An example of support-set is presented in Figure 3. In this figure, all gates except gate F have independent inputs. The support-sets if these gates consist in their two inputs. As for gate F, the support-set is {A, 3, 4, 5, 6} (A representing the output of gate A). Then, for each support-set showing reconvergence, the probabilities are computed given all inputs in the support-set. An exhaustive simulation can be performed up to a certain support-set size (e.g. 16 inputs). If the size of the support set is beyond this size, a simulation is performed with random input patterns (e.g. 2^{16} patterns). For sequential circuits, flip-flops are considered as primary inputs / outputs i.e. with $p_0=p_1=½$. The p_0 and p_1 of the gates in the combinational part are computed accordingly.

Once the transition probability is known for each node, only nodes with a probability under a certain threshold are good candidates.

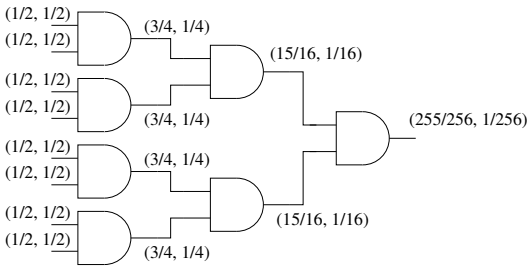

Figure 2. Nodes' probability.

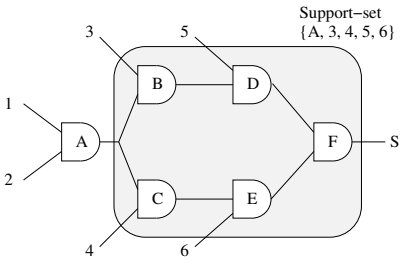

Figure 3. Support-set.

B. Slack time

The second step is to know the slack time of each node. HT detection techniques indeed exist that are based on the analysis of the delays of the circuits. The assumption is therefore that the attacker will hide the HT from a delay point of view so that it is stealthy.

Based on the gates delay model and the interconnect, a timing analysis is done to compute the circuit nodes' slack time such as presented in Figure 4:

- From the set of arrival times asserted on starting points, the analysis propagates arrival times forward (As Soon As Possible),
- From the set of required arrival times asserted on end points, the analysis propagates required arrival time backward (As Late As Possible),
- Then, the slack time at any timing node is the difference of its required arrival time minus its arrival time.

The more accurate the estimation of the delay is, the better. This computation is therefore done after place and route in order to take into account not only the gates' delay, but also the interconnect's delay. Furthermore, this is what reflects the best the information that an attacker can obtain (from the GDSII sent to the foundry).

Once this information is known for each node, only nodes with a slack time large enough to accommodate the insertion of a HT are good candidates.

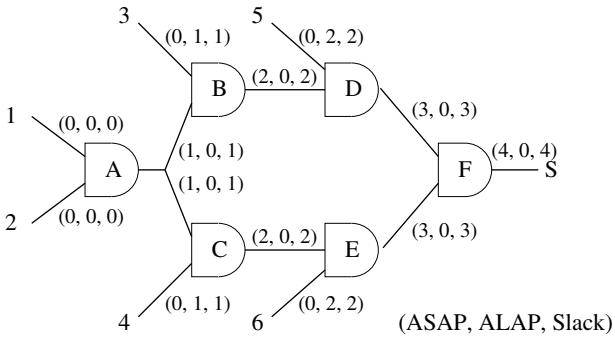

Figure 4. Slack time computation.

C. Layout

The third criterion is the available free space around the gates for an attacker to insert triggering gates. Only nodes for which there is enough free space around to insert an

extra door without compromising the placement are good candidates. This step is still under development.

D. Subsets creation

The last step is to create subsets among the nodes that correspond to the three previous criteria. The number of such subsets being still large, a last criterion is used to reduce it., Signals are partitioned into subsets according to the layout of the circuit: by combining multiple gates that are close one to the other in the circuit's layout Each subset is then a potential triggering support set. This step is also still under development.

IV. CONCLUSION

In this paper, we have presented a method for identifying potential support sets of multiple inputs triggering conditions. Three criteria are taken into account: low controllability, sufficient slack time and available space silicon area. Once these sites are identified, ATPG can be driven to produce input sequences intended to trigger the potential HTs.

REFERENCES

[1] M. Tehranipoor and F. Koushanfar. A Survey of Hardware Trojan Taxonomy and Detection. *IEEE Design & Test of Computer,* 27:10–25, 2010.

[2] Y. Jin and Y. Makris. Proof Carrying-Based Information Flow Tracking for Data Secrecy Protection and Hardware Trust. In *IEEE VLSI Test Symposium (VTS'12)*, pages 252-257, 2012.

[3] D.Agrawal, S.Baktir, D.Karakoyunlu, P.Rohatgi, and B.Sunar. Trojan Detection using IC Fingerprinting. In *IEEE Symposium on Security and Privacy (SP'07),* pages 296–310, 2007.

[4] Y. Jin and Y. Makris. Hardware Trojan Detection Using Path Delay Fingerprint. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08),* pages 51–57, 2008.

[5] F. Wolf, C. Papachristou, S. Bhunia and R. S. Chakraborty. Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme. In *Design, Automation and Test in Europe (DATE'08)*, pages 1362–1365, 2008.

[6] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In *International Conference on Cryptographic Hardware and Embedded Systems (CHES'09),* pages 396–410, 2009.

[7] H. Salmani, M. Tehranipoor, and J. Plusquellic. A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* 20(1):112–125, 2012.

[8] R. S. Chakraborty, S. Paul, and S. Bhunia. On-Demand Transparency For Improving Hardware Trojan Detectability. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08),* pages 48–50, 2008.