A Survey on Hardware Trojan Detection Techniques

Shivam Bhasin, Telecom ParisTech and Temasek Laboratories, NTU, Singapore, shivam.bhasin@telecom-paristech.fr
Francesco Regazzoni, ALaRI - USI, Lugano, Switzerland, regazzoni@alari.ch

Abstract—**Hardware Trojans recently emerged as a serious issue for computer systems, especially for those used in critical applications such as medical or military. Trojan proposed so far can affect the reliability of a device in various ways. Proposed effects range from the leakage of secret information to the complete malfunctioning of the device. A crucial point for securing the overall operation of a device is to guarantee the absence of hardware Trojans. In this paper, we survey several techniques for detecting malicious modification of circuit introduced at different phases of the design flow. We also highlight their capabilities limitations in thwarting hardware Trojans.**

I. Introduction

The semiconductor industry has warmly welcomed globalization and outsourcing as a methodology for lowering cost of Integrated Circuits (ICs). Embedded systems, which thanks to this cost reduction are now pervading our lives, are currently composed of a number of IPs. These IPs come from different vendors and they are designed and assembled using CAD tools realized by third-party. Similarly, manufacturing, bonding and packaging are usually performed by off-shore service providers. This model has open doors to a number of security threats, such as overproduction of devices, use of old and low quality components, counterfeiting, and potential modifications of the electronic circuits. These modification, maliciously and intentionally applied to the circuit, are called hardware Trojans and have recently attracted the attention of the scientific and industrial community

The goals and the effects of these Trojans can be very different. They can reduce the performance of the system, cause Denial of Service, or even spy sensitive information computed internally [31]. To date, no example of hardware Trojan has been discovered in real products. However, several theoretical works and practical proof of concepts demonstrated the real possibility and the potential devastating effects which malicious modifications might have, especially when applied to electronic circuits used in critical systems such as health, finance, aerospace or military.

A typical hardware Trojan is composed of two main parts: a **trigger**: a circuit which activates a Trojan on a specific condition, and a **payload**: a circuit which performs the malicious function. Trojans can be inserted at any point of the design process of integrated circuits and by any of the entity involved. A malicious IP designer can, for instance, modify the RTL code of a component, a malicious CAD tool can change the netlist or the layout of a design, a malicious library provider can release modified technological libraries and a malicious foundry can change the mask layout before the fabrication. Several works showed the leakage of secret information achievable by Trojans inserted at the RTL level [15]. Other works demonstrated the real possibility of inserting Trojans at mask level [7] or during the fabrication step [6].

Owing to their high criticality, hardware Trojans have received significant attention from the research community. Previous works were mainly following two directions: demonstrating the impact of Trojans [15] and try to counteract them by identifying or preventing their insertion. A complete taxonomy was also proposed [30].

Hardware Trojan detection consists mainly in developing methodologies and tools to detect malicious modification of circuits. The nature of Trojan is strictly dependent on the point of insertion, therefore detection techniques can be significantly different. Depending on the type of Trojan which has to be identified, the detection can be performed using logic testing [14], side channel analysis [2] or reverse engineering [7]. Hardware Trojans prevention techniques consist in applying certain modification to the circuit with the goal of making malicious modification of the hardware a difficult task. Known prevention techniques are: state machine obfuscation [9] or logic encoding [26].

In this paper, we survey several Trojan detection techniques, discussing the type of Trojan they can detect, their complexity, and their limitation. The rest of the paper is organized as follows: Section II summarizes current research in hardware Trojan implementation, with the goal of highligthing the threat we has to be defeated. Section III describes in details several techniques currently used to detect hardware Trojans.

II. Hardware Trojan Implementation

The vast majority of hardware Trojan proposed so far in literature are modification to the netlist at RTL level. The modification consists in an addition of a relatively small circuit which is implemented and added to the design using an hardware description language, before the synthesis and the placing and routing of the IC.

A prominent example of this type of Trojan is the one proposed by King et al. [17]. The malicious modification allows an external adversary to obtain the complete control of the system under attack: the adversary, by sending specific malicious UDP packets, captured by the modified CPU, is able to cause arbitrary changes to the software or even obtain unlimited memory access.

A different way to leak secret information consists in inserting Trojans which create hidden side channels. The adversary has to exploit them to extract internal secrets. It was demonstrated that this can be realized using only a very limited number of gates [21].

Hardware Trojan design is also the topic of an annual challenge: the hardware Trojan challenge yearly organized by

978-1-4799-8391-9/15/$31.00 ©2015 IEEE

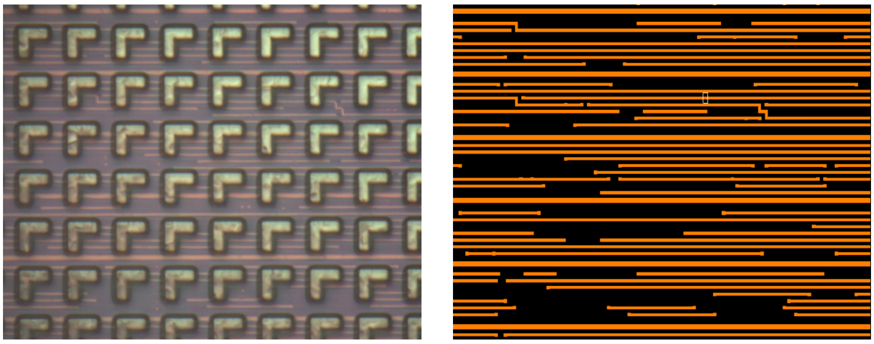

Fig. 1. Optical inspection technique for hardware Trojan detection. The target chip (AES-128 Accelerator), on the right, is compared with its layout (M6 only), on the left. Source: [7]

NYU-Poly. In this challenge, designers and researchers attempt to inject hardware Trojans into reconfigurable hardware aiming at defeating current detection techniques [25].

These Trojans, like the majority of the ones proposed so far, were added to the hardware description language of the device under attack, mimicking the attack scenario where the malicious modifications is performed by an untrusted designer. This type of Trojan can not be easily implemented by malicious foundry. In fact, foundries have usually access only to the final design file storing the the layout information (the GDSII file), where the design is already synthesized and placed & routed. In this scenario, find the space for placing and routing the Trojan gates is very difficult. Furthermore, if applied after the creation of the GDSI, the Trojan gates can be relatively easily identified by optical reverse engineering.

Trojan applied by malicious foundry can be more realistically inserted at the layout level, often manipulating the doping. Shiyanovskii et.al. [29] proposed to change the concentration of doping with the goal of speeding up the aging of the circuit, having as ultimate result the reduction of the lifetime of the device. However, the application of this type of Trojan seems to be limited to a denial of service attack, mainly because it is hard to predict the exact moment in which the device will begin to fail.

The hardware Trojan of Becker et al [6] has also to be inserted into designs at the layout level, after the place & route phase. The Trojan is inserted by modifying the polarity of the doping in the active area. Since only the doping concentration is modified, such Trojans are almost invisible to several optical reverse-engineering commonly used for hardware Trojan detection. For instance, by controlling the doping, an attacker can replace an inverter with a always-on gate. The authors proposed two case studies to show the potential of such a Trojan, a side-channel resistant S-box realized using a protected logic style and an implementation of a secure digital random number derived from ones implemented in the Intel Ivy Bridge processors. Similar Trojan example were also demonstrated by Kumar et al. [18].

III. Hardware Trojan Detection Techniques

Detection of hardware Trojans at an early stage is extremely important because, unlike software, hardware Trojan cannot be removed once inserted. The detection allows isolating Trojan-infected circuits before circulating into the market. This prevents from consequences of the malicious modification of the hardware, if a Trojan is detected before application. Since the nature of Trojans varies widely in terms of activation mechanism, payload and insertion point in the supply chain, it is not possible to develop a universal detection technique.

Detection techniques vary depending on the required inputs and the deployement phase. Some of them require only the integrated circuit to be tested and its netlist (for instance for testing techniques), while others require the layout description of the circuit. Certain techniques also need a so called “golden chip”, which is a copy of the integrated circuit under test known to be free from hardware Trojans. In this last case, the problem of obtaining a completely reliable golden chip is still open.

Detection techniques can be also divided depending on the type of intervention which has to be applied on the device under test. Based on this classification, we can divide them into *destructive* and *non-destructive*. Functional testing, for instance, requires to apply only specific test vectors thus is not destructive. Optical Inspection requires the active removal of layers of the chip, thus it is considered destructive. In the rest of the section we discuss in details of the most commonly used techniques to detect hardware Trojans.

A. *Optical Inspection Based Detection Techniques*

Figure 1 presents the general idea of the optical inspection. The layout of the circuit under test, reported on the left side is compared with pictures of the manufactured circuit under test, obtained by removing the layers one by one and thus destroying the chip under test. Optical inspection (or visual inspection) as the name suggest relies on reverse engineering to detect Trojans. Sophisticated and highly accurate techniques for imaging acquisition and analysis are applied to obtain the die photo of the chip under test. Example of techniques applied in this domain are: Scanning Optical Microscopy (SOM), Scanning Electron Microscopy (SEM), and pico-second imaging circuit analysis (PICA). Images collected are then used to reconstruct the layout of the chip and compare with the layout produced by the designer. Further reverse engineering could allow to obtain eventually also the original netlist. Optical inspection is suitable to detect hardware Trojans applied during the fabrication, thus is a very powerful technique.

The main drawbacks of this technique are two: its cost and the time needed to apply it. For this reason, optical inspection, despite its accuracy, can be less attractive than other less accurate Trojan detection techniques. Furthermore, optical inspection can become impractical when applied on a large number of ICs because the fabrication plant can use original and infected masks for different batches. A recent improvement to the limitation of optical inspection was proposed in by Bhasin [7]. The authors present a methodology to detect Trojans by comparing images of last metal layers and the GDSII produced for fabrication. It was shown that the insertion of Trojans with high core utilization ratio (larger than 80%), would impact the higher metal layers. As a result, these Trojans can be detected by computing cross-correlation between the original GDSII and high-resolution images of the upper layers of the IC [7].

B. Testing Based Detection Techniques

Functional testing is a process which is usually applied to a chip before shipping. Testing can also be used also to detect the presence of hardware Trojans. Of course, as the object to be detected is unknown, the main issue in this case is to define a proper set of test vectors.

Several test patterns can be applied to an IC to detect any abnormal activity, while only few patterns trigger the inserted Trojan. A typical Trojan has usually extremely low activation probability which makes it hard to trigger a Trojan with standard test patterns. For this reason, researchers focused on making testing techniques capable of detecting hardware Trojans. Banga et al. [5] proposed to use the inverted output of flip-flops $\overline{Q}$ in order to raise the control over them and enlarge the space of reachable states. Jha and Jha [14] propose a randomization to compare, in probability, the functionality of the original design and the final circuit. Tehranipoor et al. [28] presented a method to increase the probability of generating a transition in a Trojan and analyze its activation time. In [12], authors suggest to test the rare occurrences on an integrated circuit rather than testing for correctness.

The advantage of testing is that it is not invasive. Also, in principle, testing can identify hardware Trojans applied at different level of the design flow, including malicious modifications applied in the IP itself. However, it is not guaranteed to find the test vectors capable of triggering Trojans and therefore detection using testing is quite unsure.

C. Side Channel Based Detection Techniques

A powerful technique to detect malicious modification in integrated circuit is Side-Channel Analysis (SCA). This technique compares a physical characteristic (like power consumption, EM radiation, time delay) of the device under testing against a reference circuit. Also in this case, a “golden circuit” is almost always needed. The overall principle is depicted in Figure 2. One of the first work in this area was proposed by Agarwal et al. [2]. The authors used Principle Component Analysis (PCA) as side-channel fingerprint of the circuit to compare it with golden model. Other physical characteristics were also used in the past to detect Trojans, example of these are leakage current [23], [24] or dynamic current [13], [27] or internal delays [16], [20].

Banga et al. [4] put forward the “sustained vector technique” to magnify the difference in power consumption between the golden model and infected circuits. In [19], authors present a practical evaluation of previously proposed techniques such as PCA [2] and proposed an improved method based on cumulative probability distribution.

The main limitations of side channel based Trojan detection are two. Firstly, the physical characteristic can be modified also by other factors and not only by the hardware Trojan. For instance, power side channel fingerprint of a circuit under test might be different from the one of golden circuit due to process variations. Secondly, the selected physical characteristic might be hard to measure precisely. It can be hard, for instance, to extract the exact timing required by a specific path in the circuit.

D. Run Time Detection Techniques

Among the non destructive techniques, there were some proposed to detect hardware Trojans at run-time. Techniques designed for this purpose are designed together with a physical countermeasures. Once a Trojan is detected in the operation phase, an attempt is made to bypass the detected Trojan and operate the circuit safely. The work of Bloom et al. [8] is in this direction. The proposed technique uses both hardware and software to detect Trojans. Along the same line is the detection technique called DEsign-For-ENabling-SEcurity (DEFENSE [1]). In this case, reconfigurable logic is added to the functional design in order to support a security monitoring at run-time. These techniques can be expensive in terms of circuit area and are not able to detect certain types of Trojans [22].

E. Invasive Detection Techniques

Invasive methods consist to modify IC structures in design phase (RTL, netlist levels) with the goal of avoiding the insertion of hardware Trojans. However, similar techniques can be used to assist and to increase the effectiveness of detection techniques applied after the design is manufactured. In [3], authors propose a tool for IP protection. In this technique, designers modifies voluntarily the IPs and use the voluntary change as a fingerprint to identify the IP after fabrication. This fingerprint is designed to be low-overhead and to bypass standard detection techniques. In details, it is an extended FSM which can be triggered by specific inputs from the designer. In [9] [10] [11], the authors show different hardware obfuscation methods to prevent hardware Trojan insertion and to assist others detection methods. These invasive methods can be difficult to achieve for a complex IC and they can add increase even significantly the cost of the ICs.

IV. Conclusion

With the goal of providing an updated overview of the current possibilities of defeating malicious and deliberate modification to integrated circuits, in this paper, we reviewed several techniques to detect hardware Trojans after their insertion. For each of the considered technique, we have highlighted their operating principle, discussed their application field (which type of Trojan they are capable of detecting) and summarized their limitations.

Acknowledgments

Authors thanks *Xuan-Thuy Ngo* from *Telecom Paristech, France* for sharing his results and his interesting comments.

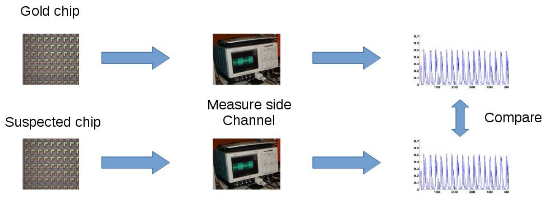

Fig. 2. Graphical representation of side channel technique for hardware Trojan detection. The physical characteristics of the golden chip, on the left, are compared with the ones of the chip under test, on the right. This technique is not destructive.

REFERENCES

[1] M. Abramovici and P. Bradley. Integrated circuit security: new threats and solutions. In F. T. Sheldon, G. Peterson, A. W. Krings, R. K. Abercrombie, and A. Mili, editors, *CSIIRW*, page 55. ACM, 2009.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan Detection using IC Fingerprinting. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 296–310, Washington, DC, USA, 2007. IEEE Computer Society.

[3] Y. Alkabani and F. Koushanfar. Designer's hardware trojan horse. In *HOST*, pages 82–83, 2008.

[4] M. Banga and M. S. Hsiao. A Novel Sustained Vector Technique for the Detection of Hardware Trojans. In *Proceedings of the 2009 22nd International Conference on VLSI Design*, VLSID '09, pages 327–332, Washington, DC, USA, 2009. IEEE Computer Society.

[5] M. Banga and M. S. Hsiao. ODETTE : A Non-Scan Design-for-Test Methodology for Trojan Detection in ICs. In *International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE*, pages 18–23, 2011.

[6] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson. Stealthy dopant-level hardware trojans. In *Proceedings of the 15th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'13, pages 197–214, Berlin, Heidelberg, 2013. Springer-Verlag.

[7] S. Bhasin, J.-L. Danger, S. Guilley, T. Ngo, and L. Sauvage. Hardware Trojan Horses in Cryptographic IP Cores. In *FDTC*, pages 15–29, August 20 2013. Santa Barbara, CA, USA.

[8] G. Bloom, B. Narahari, and R. Simha. OS Support for Detecting Trojan Circuit Attacks. In *HOST*, pages 100–103, 2009.

[9] R. S. Chakraborty and S. Bhunia. Hardware protection and authentication through netlist level obfuscation. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '08, pages 674–677, Piscataway, NJ, USA, 2008. IEEE Press.

[10] R. S. Chakraborty and S. Bhunia. HARPOON: an obfuscation-based soc design methodology for hardware protection. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 28(10):1493–1502, 2009.

[11] R. S. Chakraborty and S. Bhunia. Security through obscurity: An approach for protecting register transfer level hardware IP. In *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009, San Francisco, CA, USA, July 27, 2009. Proceedings*, pages 96–99, 2009.

[12] R. S. Chakraborty, F. G. Wolff, S. Paul, C. A. Papachristou, and S. Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In C. Clavier and K. Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 396–410. Springer, 2009.

[13] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia. Self-referencing: A scalable side-channel approach for hardware trojan detection. In S. Mangard and F.-X. Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 173–187. Springer, 2010.

[14] S. Jha and S. K. Jha. Randomization Based Probabilistic Approach to Detect Trojan Circuits. In *Proceedings of the 2008 11th IEEE High Assurance Systems Engineering Symposium*, HASE '08, pages 117–124, Washington, DC, USA, 2008. IEEE Computer Society.

[15] Y. Jin, N. Kupp, and Y. Makris. Experiences in hardware trojan design and implementation. In *Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, HST '09, pages 50–57, Washington, DC, USA, 2009. IEEE Computer Society.

[16] Y. Jin and Y. Makris. Hardware trojan detection using path delay fingerprint. In *HOST*, pages 51–57, 2008.

[17] S. T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, and Y. Zhou. Designing and implementing malicious hardware. In *Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET 08)*, pages 1–8, 2008.

[18] R. Kumar, P. Jovanovic, W. P. Burleson, and I. Polian. Parametric trojans for fault-injection attacks on cryptographic hardware. In A. Tria and D. Choi, editors, *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014*, pages 18–28. IEEE, 2014.

[19] S. Kutzner, A. Y. Poschmann, and M. Stöttinger. Hardware trojan design and detection: A practical evaluation. In *Proceedings of the Workshop on Embedded Systems Security*, WESS '13, pages 1:1–1:9, New York, NY, USA, 2013. ACM.

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

[21] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, LNCS, pages 382–395. Springer, 2009.

[22] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 382–395. Springer, September 6–9 2009. Lausanne, Switzerland.

[23] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware trojan horse detection using gate-level characterization. In *DAC*, pages 688–693. ACM, 2009.

[24] R. Rad, J. Plusquellic, and M. Tehranipoor. Sensitivity analysis to hardware Trojans using power supply transient signals. In *Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, HST '08, pages 3–7, Washington, DC, USA, 2008. IEEE Computer Society.

[25] J. Rajendran, V. Jyothi, and R. Karri. Blue team red team approach to hardware trust assessment. In *IEEE 29th International Conference on Computer Design (ICCD 2011)*, pages 285–288, oct. 2011.

[26] J. A. Roy, F. Koushanfar, and I. L. Markov. EPIC: Ending Piracy of Integrated Circuits. In *DATE*, pages 1069–1074. IEEE, 2008.

[27] H. Salmani and M. Tehranipoor. Layout-aware switching activity localization to enhance hardware trojan detection. *IEEE Transactions on Information Forensics and Security*, 7(1):76–87, 2012.

[28] H. Salmani, M. Tehranipoor, and J. Plusquellic. A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Trans. VLSI Syst.*, 20(1):112–125, 2012.

[29] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay. Process reliability based trojans through NBTI and HCI effects. In *NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2010)*, pages 215–222, 2010.

[30] M. Tehranipoor and F. Koushanfar. A Survey of Hardware Trojan Taxonomy and Detection. *IEEE Des. Test*, 27(1):10–25, Jan. 2010.

[31] U.S. Department Of Defense. Defense science board task force on high performance microchip supply. http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf.