Hardware Trojan Threat and Its Countermeasures

Xuan-Thuy Ngo*, Van-Phuc Hoang†, Han Le Duc†
* Secure-IC S.A.S., 15 Rue Claude Chappe, Bât. B ZAC des Champs Blancs, 35510 Cesson-Sévigné, France.
† Le Quy Don Technical University, 236 Hoang Quoc Viet Str., Hanoi, Vietnam.

***Abstract*—Hardware Trojan (HT) is a new threat in the embedded system domain. It is malicious modification of a design during its conception flow. Once inserted, it can create many critical attacks on a system. Therefore, HT inserted in integrated circuits have received special attention of researchers. In this paper, we will first introduce the concept of HT. Then, we will give a summary of HT detection methods including optical methods, testing methods, Side-Channel methods. Finally, we will focus on the Side-Channel methods and our perspectives for the future work.**

I. Introduction

The semiconductor industry has developed quickly so that it involves different companies and countries in the time of globalization. As a results, different design phases for an Integrated Circuit (IC) may be implemented at geographically dispersed locations. In the semiconductors industry, outsourcing for IC design and fabrication has become a common trend as well. However, this trend also results in new security threats. One of these threats which was raised few years ago is the HT insertion. In general, an HT is a malicious hardware module inserted in an IC during the design or fabrication stage. Once inserted, the HT can perform many dangerous tasks such as Denial of Service (DoS), deteriorate circuit performance [17], leakage of sensible data via circuit outputs, etc [25].

Because of its malicious and dangerous natures, HT can create serious problems in many critical applications as military system, financial infrastructures, heath applications, IoTs, etc. This threat is mentioned in some recent military reports: US Defense [17] and IEEE Spectrum [2]. Therefore, In 2007, DARPA (Defense Advanced Research Project Agency) has initiated its TRUST in Integrated Circuits [16] program to develop technologies that ensure the trust of ICs used in military systems, but designed and fabricated under untrusted conditions. In 2008, The Australian Department of Defense raised an awareness of HT and proposed classes of HT and their countermeasures [30]. In 2012, European Union's Seventh Framework Programme has launched the HINT program to prevent the counterfeit circuit and HT. Recently, the French funded R&D project HOMERE (Hardware TrOjans : Menaces et robustEsse des ciRcuits intEgrés) started in 2012 with the objective to search for and develop HT detection methods.

In this paper, we will present in more detail this threat and its countermeasures. The rest of this paper is organized as follows: Section II will describe the HT structure and possible HT insertion steps. Section III give a summary of HT countermeasures including prevention and detection methods.

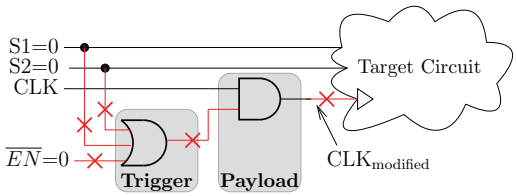

Figure 1. Minimalist HT example

In section IV, we will discuss about HT detection method using Side-Channel Analysis and our perspectives. Finally, we will provide a short conclusion in section V.

II. HT Attack

A. HT Structure

As described before, An HT is a malicious module inserted in an Integrated Circuit during the design or fabrication stage. It can be implemented in ASIC circuits, microprocessor, microcontrollers, GPU, DSP and also in FPGA bitstream. By definition, an HT consists of two basic components. The first component is the **Trigger** which *reads* the state of the target circuit (the condition to trigger its malicious function). The second one is the **Payload** which *writes* on the target circuit state (to executes its malicious function).

Fig. 1 gives an example of one simplistic HT. In this HT example, the trigger is a simple OR gate: it tests if inputs S0 or S1 equals to 0; the payload is an AND gate: it will disable the system clock once the HT is activated hence creating a Denial of Service.

B. HT Insertion Phases

To understand the threat of HT insertion, it is necessary to consider the IC development process. The Fig. 3 (a) presents this development process which start with the idea. Then the precise specifications will be defined. After, a model written with Hardware Description Language (HDL) will be used to develop these specifications. Typically, the circuit will not be designed from scratch: the designer frequently uses embedded components coming from third parts (IP-cores). These IP-cores can be in different forms (VHDL or Verilog code, netlist). Then the circuit will be transformed (synthesized) to logical gates of the target technology. After, these elements will be placed and routed on the chip, hence creating the layout of the circuit. Finally, the layout will be implemented on FPGA or be sent to the foundry for the fabrication of an ASIC. We can see that there are many different and complex steps for

978-1-5386-7983-8/18/$31.00 ©2018 IEEE

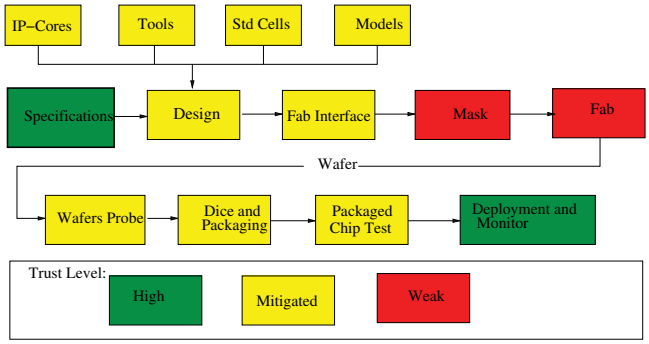

Figure 2. Trust level of IC development flow [16]

an IC development. The final production stage for an ASIC is particularly costly and is carried out by semi-conductors fabs. That is why fabless companies outsource their designs or some steps of development. Fig. 3 (b) presents the different HT attack scenarios regarding the different IP development steps.

So the HT can be inserted in different steps of IC development process, from specification to packaging and test step. Fig. 2 from [16] indicate the trust level of each step of the design flow. The steps with an high trust level (those were great controlled) are shown in green, those with mixed trust levels (where the control depends on the context) in yellow, and those with low trust level in red. In the next section, we will talk about the HTs classification and designs in the state of the art.

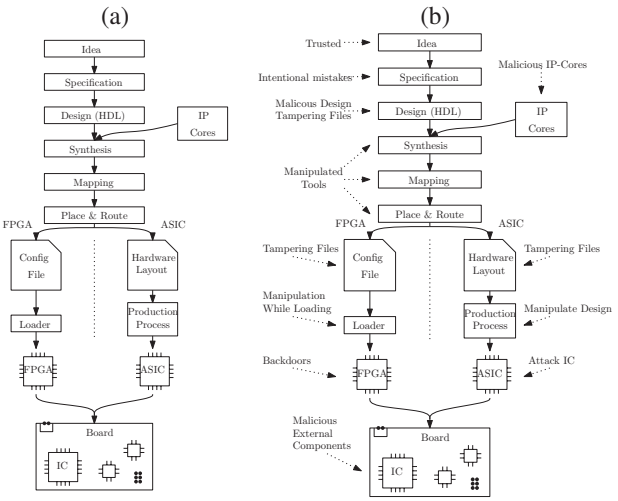

Figure 3. (a): IC development process. (b): HT scenario attacks on IC development process [19]

C. HT Implementations

In the state of the art, many examples of HT is shown/discovered in accademic and industry. For industrial application, Skorobogatov and al. discover an undocumented backdoor inserted into the Actel/Microsemi ProASIC3 chips (military grade chip) for accessing FPGA configuration [37] in 2012. By using this HT, an attacker can extract the configuration data from the chip, reprogram the crypto core and access keys. Moreover, it can modify low-level silicon features, access unencrypted configuration bitstream or permanently damage the device. ProASIC3 devices are used in many critical products such as weapons, guidance, flight control, networking and communications therefore this HT can create fatal accidents. But we do not know if this backdoor is inserted intentionally by Actel or by a malicious attacker. In 2014, the discover of specific US-made components designed to intercept the satellites' communications in France-UAE satellite was reported in the news of www.rt.com. But we do not have much information about these components. Recently, in 2014, documents leaked by NSA whistleblower Edward Snowden intimate that the NSA planted back-doors in Cisco products as routers hence gaining access to entire networks and all their users. Routers, switches, and servers made by Cisco are booby-trapped with surveillance equipment that intercepts traffic handled by those devices and copies it to the NSA's network, the book states.

For academics, many researchers have focused on the topic of HT insertion and reported some small and intelligent HTs. In [27], King et al. presented a malicious HT on a open-source Leon3 processor that maliciously grants root-level privileges to an attacker, then creates a permanent backdoor into a system and steals passwords from other users of the system. This HT is composed of 2 modifications on CPU: modification of memory mechanism which allows attackers to access the protected memory areas and the second modification is the shadow mode creation allowing attackers to execute the hidden firmware. The overhead of this HT is around 1341 gates. In 2008, Polytechnic Institute of NYU launched the Cyber Security Awareness Week (CSAW) Embedded System Challenge whereby the goal is to insert HTs to compromise an FPGA cryptographic device Alpha. The winner in 2009 [25] and in 2011 [7] proposed several HT mechanisms to leak the secret keys from circuit IO and serial communication channel (UART). These HTs are principally inserted in the design phase. Lin et al. proposed in [29] HT structures, named HT Side-Channels (TSC), that are less than 50 gates to leak the secret information via power side-channels. It is implemented in an AES cryptographic co-processor and allows leaking multi-bit information below the IC noise level in order to not be detected during test time. Nowadays, more and more cryptographic co-processors are implemented in embedded systems to accelerate the encryption/decryption of sensitive data or to secure the communication channels. Therefore HTs can be inserted in these co-processors in order to leak the secret key (used for encryption/decryption) hence bypassing the system security mechanism or to assist another physical attacks as SCA, FIA, etc. In 2007, Agrawal et al. demonstrated two simple HTs embedded in RSA encryption [3]. In [20], David et al. also proposed a HT structure which leak the encryption keys of an AES via RS232 communication channel without perturbing the system. This HT encodes the '0' data

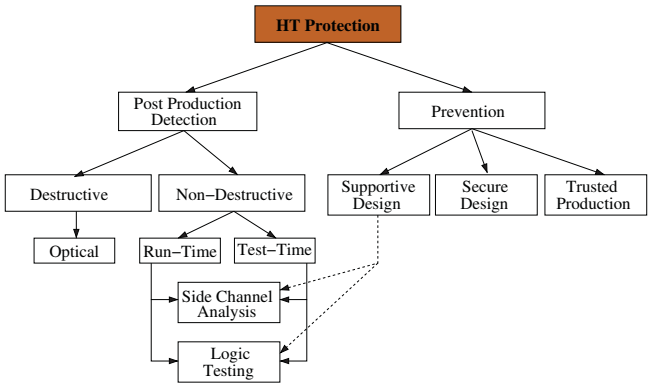

Figure 4. HT Detection Methods Overview [19]

by a null signal and the '1' data by a 10 ns pulse. And the key will be retrieved by monitoring the RS232 line at the speed of 1200 bits per second. In [8] show an example of HT inserted in AES co-processor at the layout level. Jin et al. experimented an example of HT which leak the key of Data Encryption Standard (DES) algorithm in [25]. Michael Muehlberghuber et al. also demonstrate an example of HT insertion on AES at Layout Level in [31]. So HT is a real threat that is received many attention in the last decade. Therefore, HT countermeasures development is crucial. In the next section, we will summary different HT countermeasures.

III. HT Countermeasures

In the state of the art, there are several dedicated detection methods of HTs. Fig. 4 presents the overview of these detection methods. These methods can be classified into two categories: *Post Production Detection* and *Prevention*.

A. Prevention Methods

Prevention methods consist in modifying/obfuscating directly the original design during the conception phase in order to make a secure design, to assist another detection technique or to create a trusted production chain. Chakraborty et al. have presented a prevention method against HT at ICCAD 2009 [11]. It is inspired by obfuscation methods [4], [12] initially intended to protect against IC counterfeiting. In this paper, the original design is obfuscated by increasing the total number of reachable states of the original circuit. These states are then partitioned into two parts including an original state space and an isolation state space. The original state space will be reached using a specific input pattern (as a secret key). Moreover, with any wrong input pattern, the IC will fall in the isolation state space. This space is constructed such that, once entered, it cannot be exited and outputs will never be correct.

Another technique, nicknamed ODETTE [6], aims at changing the polarity of the flip-flops (also known as DFFs). This option can be achieved at low cost, since DFFs of standard libraries (provided by the founders) feature two complementary outputs (called $\overline{Q}$ and Q). This coding is akin to Vernam cipher, where each bit of the state is masked with one bit of secret. Authors claim that it is able to obfuscate partially the circuit. In [18], authors present a logic gates encryption technique using an external key to prevent HT insertion. The drawback of these prevention methods in the state of the art is that they obfuscate only the state machine of the IC. This means that only the control part is protected, while the combinational part is unprotected. Moreover in papers presented in [11], [18], when the IC is well configured to reach the original state using static configuration keys, it operates normally and cannot resist others physical attacks. The prevention method, ODETTE [6], is more intended to raise the HT activity for a better detectability than a proactive prevention. Furthermore, each bit of the state is masked with one bit of secret.

In [33], the concept of "encoded circuits" was proposed and a provable randomization method using the Linear Complementary Pair (LCP) codes C and D was used to prevent HT insertions. Encoded circuits are realized by encoding all internal registers (sequential part) of the target design with a *binary code* C and followed by addition (XOR) of random masks in its *supplementary code* D. Once the sequential part is encoded, the combinational part can be easily obfuscated by exploiting the "flatten" option of the netlist synthesis tool. After encoding, the design complexity increases so that the real functionality of the IC is obfuscated. Using this coding method, they manage; to some extent, to protect both control and data parts. Moreover, they can not only protect against HT insertion attack but also against others physical attacks because of the use of random masks. They also propose two security parameters $d_{Trigger}$ and $d_{Payload}$.This method is at the same time prevention and run-time detection method. $d_{Trigger}$ and $d_{Payload}$ parameters can be chosen independently in order to increase the prevention capacity or detection capacity.

B. Post Production Detection

Post production detection methods consist in detecting the HT insertion after IC fabrication. It can be done using destructive or non-destructive approaches. The destructive approach is Destructive Reverse Engineering method which consists in reconstructing the netlist and layout of test circuit and comparing it with those of reference circuit. They also can be detected using non-destructive methods at run-time or test-time. Run-time techniques consist monitoring the circuit operation to detect the anomalies created by HT insertion. And Test-time methods consist in detecting HT basing on testing techniques.

1) Destructive Reverse Engineering: By studying the HT structure and implementation, we pay attention to the following definition: HT is a **malicious modification** on the target circuit. It means that an HT, inserted at the layout level (GDSII files), will create the physical modification in the original IC layout. Therefore, there will be the differences between the genuine layout (the original layout) and the infected layout. And this difference can be observed optically. For this reason, Destructive Reverse Engineering (DRE) is the first approach which could be used for HT detection.

Generally, destructive reverse engineering is often used in the semiconductor industry to extract technical or patent related information from a competitor's chip [22]. The chip can be reverse engineered at different levels like: product components, system-level, process-level and circuit-level [39]. In our case we are interested with the circuit-level reverse engineering. Modern devices are fabricated in technology like 45 nm or lower which can have up to 12 layers of metal and several millions of transistors. The details of the circuit are extracted in the following sequential steps including package removal, delayering and imaging.

In the state of the art, "destructive reverse-engineering" can be used to detect HT for our scenario. Reverse-engineering is generally performed by Chemical Metal Polishing (CMP) followed by Scanning Electron Microscope (SEM) image reconstruction and analysis [38]. It helps to reconstruct exactly all via, metal and silicon layers. After, the determination of the "correctness" of a chip is performed through visual comparison with a known good example or "golden reference". But this technique is very expensive since it takes a lot of time (hence costs a lot) to realize properly: the mere error in a picture (because the delayering left pieces of material on the chip surface or because the recognition software [36] failed) makes the reverse engineering fail. In addition, modern devices are extremely small and densely packed, which makes the cost (in money and time) of reverse engineering even higher.

In [15], authors present the "SEMBA" method, a fast invasive technique for white team Hardware Trojan detection, used to differentiate between a maliciously infected integrated circuit and a genuine one. This methodology is based on the observation of the component's hardware structure and includes the use of wet etching, Scanning Electron Microscopy and Multiple Image Alignment. Once the Integrated Circuits' image have been fully reconstructed, image processing allows to detect the presence of the Hardware Trojan (HT). They clame that "SEMBA" is a fully automated approach with a 100% success rate, detecting any 'transistor-size' HTs and requiring 'affordable' resources and time.

In [8], authors present a HT detection method at the layout level of integrated circuits by using a low-cost *direct* visual detection. They also show the feasibility and impacts of HTs insertion to the circuit layout (especially for AES circuit). Additionally, they have demonstrated the possibility to detect HTs with a low cost visual technique; this technique can be automated thanks to a cross-correlation. In particular, the observation of the sole top-level metal layer suffices (for large enough HTs); thus the method avoids mechanical or chemical preparation, known to produce dust of material that would cause a lot of false positive detections. As a corollary, if no HT is detected in the observed circuit, then this circuit can be trusted and used safely in an application, even if other circuits from a different batch happen to be infected by a HT. The results show that the insertion complexity and the visibility of HT increases with the Core Utilization Rate (CUR) of circuits. With a high CUR, HT can be detected by comparing layout images and the GDSII file.

2) Runtime Detection Methods: HTs detection at run-time can be seen as the last line of defense when other techniques (prevention, detection at test time) fail. This method can combine with other detection methods to improve the HT detection probability.

Regarding the run-time method, Huffmire et al. specify legal memory access policies for FPGA-based embedded systems [21]. The policies are synthesized into a reconfigurable hardware module that decides the legality of every memory access request generated from a datapath module. This work was further developed into a method of generating hardware-based security checkers to detect processor malicious inclusions at run-time [9]. In [1], authors proposed to add reconfigurable Design-For-Enabling-Security (DEFENSE) logic to the functional design to implement real-time security monitors. It can be reconfigured dynamically in order to implement different security checks. In [26], Kim et al. present a Trojan-resistant SoC bus architecture which detect and report all malicious bus behavior to the system CPU. In [10], authors proposed a secure execution environments named SHADE architecture by combining two layers of hardware encryption with a heartbeat of off-chip. In [34], authors create a synthesizable assertions module Hardware Property Checker (HPC) which verifies the permitted and prohibited behaviors of IC at run-time. The advantage of this method is the reduction of development time comparing to those in the state of the art by using the hardware assertion and particularly by using the Property Specification Language (PSL) to validate the properties to be checked in simulation stage. Moreover, HTs with a combination of Hardware and Software (HW/SW) vulnerability can be detected with this detection method. They also propose to detect HPC using different approaches such as 3-D circuit and configurable HPC using FPGAs.

3) Test Time Methods: In the state of the art, the first HT detection approach at test-time is using logic testing. It involves applying test patterns at the input and try to detect abnormal behaviors of ICs [6]. For logic testing, Jha and Jha in 2008 present a randomization-based technique that checks randomly the functionality of the design of the circuit with the test circuit [23]. It allows increasing the HT detection probability. Chakraborty et al. suggest to test rare occurrences on an IC rather than testing for correctness [14]. These rare occurrences will be potentially used by attacker for HT activation because of its malicious nature. Therefore, using the test patterns which create these rare occurrences, the HT activation probability during the test time will be increased. A tool is used to determine rare states and also the corresponding test patterns. In 2009 [13], Chakraborty et al. present a statistic study which shows that a HT with a maximum of four trigger nodes and one payload node can have a 10^9 possible triggers and 10^{11} possible payload in a target circuit c880 (an 8-bit ALU) with 451 gates. It shows that HT detection, using logic testing, is very difficult (even impossible) because of the complexity of test patterns. And there can be also HTs which do not modify the states of target circuit hence bypassing all logic testing techniques. For this

reason, many researchers focus on Side-Channel Analysis to detect HT at test-time.

In the state of the art, Side-Channel Analysis (SCA) has long been used as a tool to attack cryptographic algorithms. However, it can also be deployed to detect HTs for the following reason: HTs are malicious modifications in the target circuit. Therefore the layout of infected circuit and the genuine circuit will not be physically the same. It will produce the difference of physical characteristics between the infected circuit and genuine one. The HT detection based on SCA observes and compares these physical traits (example power consumption, time delay, etc.) of a test IC against a trusted IC named "golden circuit". In 2007, Agrawal et al. [3] show an example of this approach. Some golden (genuine) ICs is obtained to generate SCA traces that will be used as the IC fingerprint. Then test chip traces will be generated to compare with the fingerprint. And the HT will be detected basing on the difference of this comparison. In 2008, Plusquellic et al. [35] used the power supply transient signal analysis as side channel to detect HTs. They believe to be capable to detect HTs which have only three additional gates by using the simulation results. Jin et al. [24] propose the path delay as the side channel to detect HTs. Authors claim to be capable to detect 100% explicit HTs (HTs which affect directly the circuits) and 30% implicit HTs (HTs which do not make changes to the circuitry of target circuit). Wang et al [40] use current charge integration from multiple measurement points to detect HT. In 2009, Banga et al. [5] propose the "Sustained Vertor Technique" to magnify power difference between infected circuits and genuine circuits. This technique consists in repeating several time certain inputs hence allowing circuit time to reach a stable state. It also allows isolating the infected regions. In [28], authors present a practical evaluation of HT detection using SCA on FPGA. But the experiments were performed on a single FPGA, so the process variations were not taken into account. The placement and routing of original circuit in golden model and infected models are not the same in the experiments, which makes it hard to quantify the effect of HTs alone. In [32], authors present their SCA detection methods based on delay and electromagnetic measurements. These methods can detect inserted HTs by using the direct comparison of the delay/EM measurement between the genuine and infected designs. HTs with different sizes are tested to estimate the detection probability as a function of its size taking into account the inter-die process variations. 8 different FPGA of the same reference (Xilinx *LX*30) are used to study the impact of inter-die process variations on this detection probability. The implementation results show that, by using this metric, there is a probability greater than 95% with a false negative rate of 5% to detect a HT larger than 1.7% of the original circuit area.

IV. Discussion and Perspectives for HT detection method using Side-Channel Analysis

In the state of the art, there are many different countermeasures against HT insertion. For our point of view, each countermeasures have their own drawbacks. For optical methods (reverse engineering method), we need to destroy the design for testing. Moreover, we need other skills on chemical, reverse engineering, image processing, etc. So these method could be expensive in term of time and money. For prevention methods, we need to modify the circuit to add extra logic. Some prevention methods can increase substantially the overhead of the circuit. These methods also require the participation on the SoC and IP design flow. For runtime detection and logic testing methods, it can detect HTs only if it is activated during testing time or during the operation of the circuit. This scenario is nearly impossible because of the furtive nature of the HT. A smart attacker will insert a HT that bypass all logic testing procedures. And during the operation of the system, the detection could be too late. For now, we think that the Side-Channel analysis method is a best way for detecting the HT. These methods do not need to destroy the test chip. Moreover, they can detect the HT even if it is not activated. It is because that the trigger part of HT alway monitors the system state for searching the activation condition. The activity of this monitoring will contribute on the Side-Channel signals. One drawback of this method is how can we obtain the "reference" circuit. In [32], authors present a methodology to obtain the "reference" circuit using a low-cost optical method. So it opens the possibility to apply the HT detection methods using SCA for different scenario. However, all research works on this fields is limited on prototype designs. Moreover, there is not a work that compare the efficient of theses methods using different SCA information such as Power, Electromagnetic, Delay. For these reasons, we want to investigate on an advanced SCA method against HT insertion. Our perspectives are the followings:

- Analysis the efficient of different SCA methods;
- Application of the SCA methods on real circuits (Processors, SoC, etc.);
- Use the Virtual traces as "reference traces" for SCA detection methods.

V. Conclusion

Nowadays, the Hardware Trojan is a real threat of embedded system field. Many researchers and governments are working on this problem. In this paper, we introduced the concept of HT, its structure and its attack scenarios. We also summaried different countermeasures against HT such as reserve engineering, prevention, testing, Side-Channel and runtime methods. We believe that, for instant, SCA method is the best one for detecting the HT. However, an advanced study on this method is required in order to become an industrial method.

Acknowledgment

This research is funded by the project of Fostering Innovation through Research, Science and Technology (FIRST) in "Hardware Cybersecurity: Methods, Technologies and Applications" under grant number 28/FIRST/1a/LQDTU.

References

[1] M. Abramovici and P. Bradley. Integrated circuit security - new threats and solutions. In *Cyber Security and Information Intelligence Research Workshop (CSIIRW), ACM*, 2009.

[2] Sally Adee. The Hunt For The Kill Switch. *IEEE Spectr.*, 45(5):34–39, May 2008.

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

[4] Yousra M. Alkabani and Farinaz Koushanfar. Active hardware metering for intellectual property protection and security. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 20:1–20:16, Berkeley, CA, USA, 2007. USENIX Association.

[5] M. Banga and M. S. Hsiao. A novel sustained vector technique for the detection of hardware trojans. In *International Conference on VLSI Design, IEEE*, pages 327–332, 2009.

[6] Mainak Banga and Michael S Hsiao. Odette: A non-scan design-for-test methodology for trojan detection in ics. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, 2011.

[7] Alex Baumgarten, Michael Steffen, Matthew Clausman, and Joseph Zambreno. A case study in hardware trojan design and implementation. *International Journal of Information Security*, 10(1):1–14, 2011.

[8] S. Bhasin, J. Danger, S. Guilley, X. T. Ngo, and L. Sauvage. Hardware trojan horses in cryptographic ip cores. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 15–29, Aug 2013.

[9] Michael Bilzor, Ted Huffmire, Cynthia E. Irvine, and Timothy E. Levin. Evaluating security requirements in a general-purpose processor by combining assertion checkers with code coverage. In *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pages 49–54, June 2012.

[10] Gedare Bloom, Bhagirath Narahari, Rahul Simha, and Joseph Zambreno. Providing secure execution environments with a last line of defense against trojan circuit attacks. *Comput. Secur.*, 28(7):660–669, October 2009.

[11] R. S. Chakraborty and S. Bhunia. Security against hardware trojan through a novel application of design obfuscation. In *International Conference on Computer-Aided Design Digest of Technical Papers (ICCAD), IEEE*, pages 113–116, 2009.

[12] Rajat Subhra Chakraborty and Swarup 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.

[13] Rajat Subhra Chakraborty, Seetharam Narasimhan, and Swarup Bhunia. Hardware trojan: Threats and emerging solutions. In *IEEE International High Level Design Validation and Test Workshop, HLDVT 2009, San Francisco, CA, USA, 4-6 November 2009*, pages 166–171, 2009.

[14] Rajat Subhra 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, *Cryptographic Hardware and Embedded Systems - CHES 2009*, pages 396–410, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[15] F. Courbon, P. Loubet-Moundi, J. J. A. Fournier, and A. Tria. Semba: A sem based acquisition technique for fast invasive hardware trojan detection. In *2015 European Conference on Circuit Theory and Design (ECCTD)*, pages 1–4, Aug 2015.

[16] DARPA. Trust in integrated circuits, 2007.

[17] U.S. Department Of Defense. Defense science board task force on high performance microchip supply.

[18] S. Dupuis, P.-S. Ba, G. Di Natale, M.-L. Flottes, and B. Rouzeyre. A novel hardware logic encryption technique for thwarting illegal overproduction and hardware trojans. In *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International*, pages 49–54, July 2014.

[19] J. Francq and F. Frick. Introduction to hardware trojan detection methods. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 770–775, March 2015.

[20] D. Hely, M. Augagneur, Y. Clauzel, and J. Dubeuf. Malicious key emission via hardware trojan against encryption system. In *Computer Design (ICCD), 2012 IEEE 30th International Conference on*, pages 127–130, Sept 2012.

[21] Ted Huffmire, Timothy E. Levin, Thuy D. Nguyen, Cynthia E. Irvine, Brett Brotherton, Gang Wang, Timothy Sherwood, and Ryan Kastner. Security primitives for reconfigurable hardware-based systems. *TRETS*, 3(2):10, 2010.

[22] Silicon Investigations. Company website: http://www.siliconinvestigations.com/.

[23] Susmit Jha. Randomization based probabilistic approach to detect trojan circuits. In *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, 2008.

[24] Y. Jin and Y. Makris. Hardware trojan detection using path delay fingerprint. In *International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE*, pages 51–57, 2008.

[25] Yier Jin, Nathan Kupp, and Yiorgos Makris. Experiences in hardware trojan design and implementation. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, 2009.

[26] Lok-Won Kim, John D. Villasenor, and Çetin K. Koç. A trojan-resistant system-on-chip bus architecture. In *Proceedings of the 28th IEEE Conference on Military Communications*, MILCOM'09, pages 2452–2457, Piscataway, NJ, USA, 2009. IEEE Press.

[27] Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, pages 5:1–5:8, Berkeley, CA, USA, 2008. USENIX Association.

[28] Sebastian Kutzner, Axel Y Poschmann, and Marc Stöttinger. Hardware trojan design and detection: a practical evaluation. In *Proceedings of the Workshop on Embedded Systems Security*, 2013.

[29] Lang Lin, Markus Kasper, Tim Güneysu, Christof Paar, and Wayne Burleson. Trojan side-channels: Lightweight hardware trojans through side-channel engineering. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, 2009.

[30] Bradley Hopkins Mark Beaumont and Tristan Newby. Hardware Trojans - Prevention, Detection, Countermeasures. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA547668.

[31] Michael Muehlberghuber, Frank K. Gürkaynak, Thomas Korak, Philipp Dunst, and Michael Hutter. Red team vs. blue team hardware trojan analysis: Detection of a hardware trojan on an actual asic. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 1:1–1:8, New York, NY, USA, 2013. ACM.

[32] X. Ngo, I. Exurville, S. Bhasin, J. Danger, S. Guilley, Z. Najm, J. Rigaud, and B. Robisson. Hardware trojan detection by delay and electromagnetic measurements. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 782–787, March 2015.

[33] X. T. Ngo, S. Bhasin, J. Danger, S. Guilley, and Z. Najm. Linear complementary dual code improvement to strengthen encoded circuit against hardware trojan horses. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 82–87, May 2015.

[34] X. T. Ngo, J. Danger, S. Guilley, Z. Najm, and O. Emery. Hardware property checker for run-time hardware trojan detection. In *2015 European Conference on Circuit Theory and Design (ECCTD)*, pages 1–4, Aug 2015.

[35] Reza Rad, Jim Plusquellic, and Mohammad Tehranipoor. Sensitivity analysis to hardware trojans using power supply transient signals. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, 2008.

[36] Martin Schobert. GNU software DEGATE. Webpage: http://www.degate.org/.

[37] Sergei Skorobogatov and Christopher Woods. Breakthrough silicon scanning discovers backdoor in military chip. In *Proceedings of the 14th international conference on Cryptographic Hardware and Embedded Systems*, CHES'12, pages 23–40, Berlin, Heidelberg, 2012. Springer-Verlag.

[38] Randy Torrance and Dick James. The State-of-the-Art in IC Reverse Engineering. In *CHES*, volume 5747 of *LNCS*, pages 363–381. Springer, September 6-9 2009. Lausanne, Switzerland.

[39] Randy Torrance and Dick James. The state-of-the-art in semiconductor reverse engineering. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 333–338, 2011.

[40] Xiaoxiao Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic. Hardware trojan detection and isolation using current integration and localized current analysis. In *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS '08. IEEE International Symposium on*, pages 87–95, Oct 2008.