Hardware Trojan: Threats and Emerging Solutions

(Invited Paper)

Rajat Subhra Chakraborty, Seetharam Narasimhan and Swarup Bhunia
Dept. of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, Ohio, USA
E-mail: {rsc22, sxn124, skb21}@case.edu

***Abstract*—Malicious modification of hardware during design or fabrication has emerged as a major security concern. Such tampering (also referred to as *Hardware Trojan*) causes an integrated circuit (IC) to have altered functional behavior, potentially with disastrous consequences in safety-critical applications. Conventional design-time verification and post-manufacturing testing cannot be readily extended to detect hardware Trojans due to their stealthy nature, inordinately large number of possible instances and large variety in structure and operating mode. In this paper, we analyze the threat posed by hardware Trojans and the methods of deterring them. We present a Trojan taxonomy, models of Trojan operations and a review of the state-of-the-art Trojan prevention and detection techniques. Next, we discuss the major challenges associated with this security concern and future research needs to address them.**

***Index Terms*—Hardware Trojan; Design for Security;**

I. Introduction

Hardware security to ensure *Trust* in ICs has emerged as an important research topic in recent years. Economic reasons dictate that most of the modern ICs are manufactured in off-shore fabrication facilities [1]. Moreover, modern IC design often involves intellectual property (IP) cores supplied by third-party vendors, outsourced design and test services as well as electronic design automation (EDA) software tools supplied by different vendors. Such a business model has, to a large extent, relinquished the control that IC design houses had over the design and manufacture of ICs making them vulnerable to different security attacks. Fig. 1 illustrates the level of trust at different steps of a typical IC life-cycle [3]. Each party associated with the design and manufacture of an IC can be a potential adversary who inserts malicious modifications, referred as *Hardware Trojans* [2-5]. Concern about this vulnerability of ICs and the resultant compromise of security has been expressed globally [2-4], especially since several unexplained military mishaps are attributed to the presence of malicious hardware Trojans [5, 30-31].

Ideally, any undesired modification made to an IC should be detectable by pre-silicon verification/simulation and post-silicon testing. However, pre-silicon verification or simulation requires a golden model of the entire IC. This might not be always available, especially for IP based designs where IPs can come from third-party vendors. Besides, a large multi-module design is usually not amenable to exhaustive verification [6]. Post-silicon, the design can be verified either through destructive de-packaging and reverse-engineering of the IC [3], or by comparing its functionality or circuit characteristics with a golden version of the IC [20-28]. However, existing state-of-the-art approaches do not allow destructive verification of ICs to be scalable [6]. Moreover, as pointed out in [3], it is possible for the adversary to insert Trojans in only some ICs on a wafer, not the entire population, which limits the usefulness of a destructive approach.

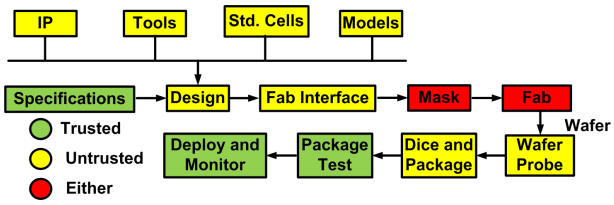


Fig. 1. Vulnerable steps of a modern IC life cycle [3].

Traditional post-manufacturing logic testing is not suitable for detecting hardware Trojans. This is due to the stealthy nature of hardware Trojans and inordinately vast spectrum of possible Trojan instances an adversary can employ. Typically, the adversary would design a Trojan that triggers a malfunction only under rare circuit conditions in order to evade detection. Due to the finite size of the testset, the rare condition for activation of the Trojan might not be realized during the testing period, especially if the Trojan acts as a sequential state machine or "time-bomb" [7]. On the other hand, the techniques for detecting Trojans by comparison of the "side-channel parameters" such as power trace or delay [22-26] are limited by the large process-variation effect in nanoscale IC technologies, reduced detection sensitivity for ultra-small Trojans and measurement noise [22].

In the next section, we give classification, models and examples of different Hardware Trojans. We then present a survey of the emerging techniques of Trojan detection. Finally, we discuss the major challenges in the field of Trojan detection and describe future research directions.

II. Trojan Taxonomy and Examples

Different methods of classifying hardware Trojans based on various characteristics have been proposed. In [27], the authors propose a simple classification of Trojans – *combinational* (whose activation depends on the occurrence of a particular condition at certain internal nodes of the circuit) and *sequential*

978-1-4244-4823-4/09/$25.00 ©2009 IEEE

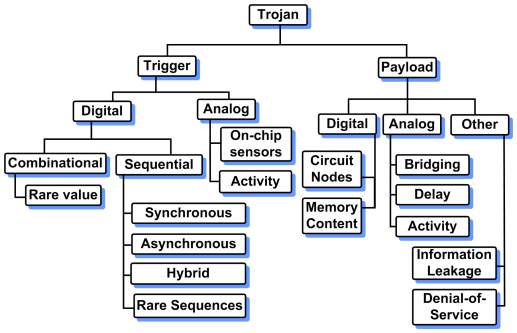


Fig. 2. Trojan taxonomy based on trigger and payload mechanisms.

(whose activation depends on the occurrence of a specific sequence of rare logic values at internal nodes). In [8], the authors classify Trojans based on three attributes: *physical*, *activation* and *action*. Some classifications (e.g. [7, 24]) are based on the activation mechanisms (referred as *Trojan trigger*) and the part of the circuit or the functionality affected by the activation of the Trojan (referred as *Trojan payload*). In this paper, we follow and expand the Trojan taxonomy proposed in [7], where the Trojans are classified based on their trigger and payload mechanisms, as shown in Fig. 2. The trigger mechanisms can be of two types: *digital* and *analog*. *Digitally triggered* Trojans can again be classified into *combinational* and *sequential* types. Fig. 3(a) shows an example of a *combinationally triggered* Trojan where the occurrence of the condition $A=0$, $B=0$ at the trigger nodes A and B causes a payload node C to have an incorrect value at $C_{modified}$. Typically, an adversary would choose an extremely rare activation condition so that it is very unlikely for the Trojan to trigger during conventional manufacturing test.

Sequentially triggered Trojans (the so-called time bombs), on the other hand, are activated by the occurrence of a sequence, or a period of continuous operation. The simplest sequential Trojans are synchronous stand-alone counters, which trigger a malfunction on reaching a particular count. Fig. 3(b) shows a synchronous k-bit counter which activates when the count reaches 2^k–1, by modifying the node ER to an incorrect value at node $ER^{\star}$. An asynchronous version is shown in Fig. 3(c), where the count is increased not by the clock, but by a rising transition at the output of an AND gate with inputs p and q. The trigger mechanism can also be *hybrid*, where the counts of both a synchronous and an asynchronous counter simultaneously determine the Trojan trigger condition, as shown in Fig. 3(d). Note that more complex state machines of different types and sizes can be used to generate the trigger condition based on a sequence of rare events. In general, it is more challenging to detect sequential Trojans using conventional test generation and application, because it requires satisfying a sequence of rare conditions at internal circuit nodes to activate them. The number of such sequential trigger conditions for arbitrary Trojan instances can be unmanageably large for a deterministic logic testing approach.

The trigger-mechanism can also be *analog* in nature, where

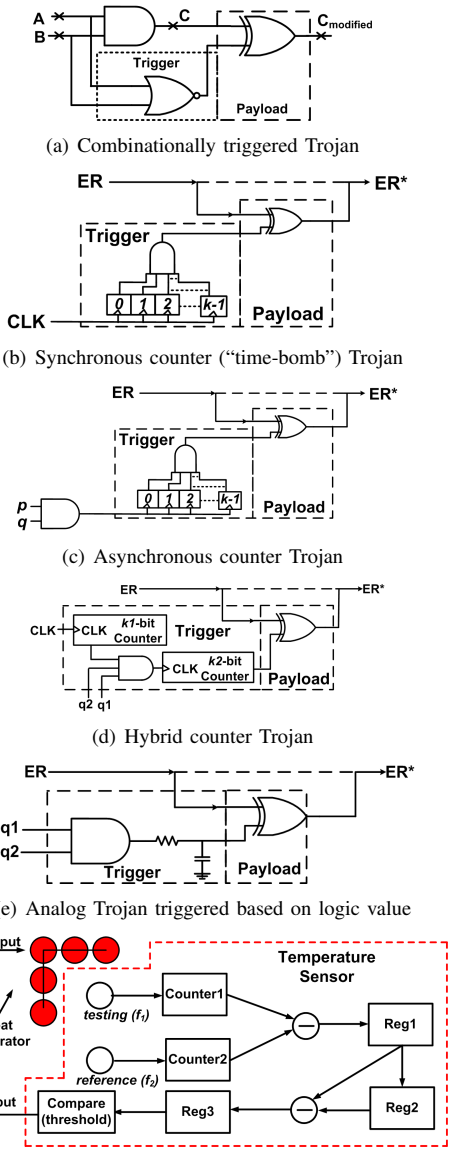


Fig. 3. Examples of Trojans with various trigger mechanisms.

on-chip sensors are used to trigger a malfunction. Fig. 3(e) shows an example of an analog trigger mechanism where the inserted capacitance is charged through the resistor if the condition $q_1=1$, $q_2=1$ is satisfied, and discharged otherwise, causing the logic threshold to be crossed after a large number of cycles. A different analog Trojan trigger mechanism (see Fig. 3(f)) was proposed in [9], where higher circuit activity and the resultant rise of temperature was used to trigger the malfunction, through a pair of ring oscillators and a counter.

Trojans can also be classified based on their *payload* mechanisms into two main classes - *digital* and *analog*. Digital Trojans can either affect the logic values at chosen internal payload nodes, or can modify the contents of memory locations. Analog payload Trojans, on the other hand, affect

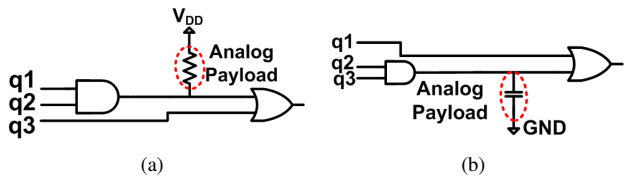


Fig. 4. Examples of analog payload Trojans.

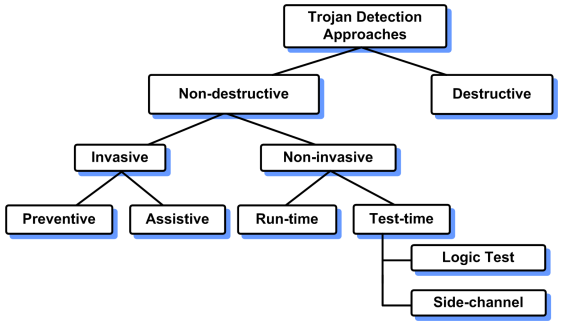


Fig. 5. Trojan detection techniques.

circuit parameters such as performance, power and noise margin. Fig. 4(a) shows an example where a bridging fault is introduced using an inserted resistor, while Fig. 4(b) shows an example where the delay of the path is affected by increasing the capacitive load. Another form of analog payload would be generation of excess activity in a circuit (similar to that shown in Fig. 3(f)), to accelerate the aging process of an IC and shorten its life-span, without affecting its functionality.

Apart from triggering logic errors in the IC, the Trojan can also be designed to assist in software-based attacks like privilege escalation, login backdoor and password theft [31]. Two different Trojan payload mechanisms were explored in the works described in [9-11]. The first is the "information leakage" attack, where secret information is leaked by a Trojan via a transmitted radio signal or serial data port interface such as the RS-232-C port. It could also involve side-channel attack where the information is leaked through the power trace [32] or through thermal radiation or through optical modulation of an output LED [33]. Another type of Trojan payload proposed is that implementing a "Denial of Service" (DoS) attack, which causes a system functionality to be unavailable.

III. Trojan Detection Methods

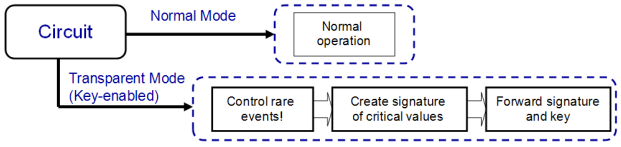
In this section, we describe the state-of-the-art of Trojan detection techniques. Fig. 5 shows a classification of existing Trojan detection techniques. Note that there is no single "silver bullet" technique available yet that can be applied to detect all classes of Trojans. Majority of existing techniques address Trojan detection in manufactured ICs and assume the availability of gate-level golden netlist. Very few investigations have addressed Trojan detection at higher level design descriptions e.g. register transfer level IP. In [38], a structural checking approach is suggested to verify integrity of third party IP, but the technique is not easily scalable to large designs [6].

The Trojan detection approaches can be classified under two main types: *destructive* and *non-destructive*. The *destructive* techniques [36-37] use a sample of the manufactured ICs which are subject to de-metallization using Chemical Mechanical Polishing (CMP) followed by Scanning Electron Microscope (SEM) image re-construction and analysis [3]. However, such approaches are extremely expensive and time-consuming (destructive analysis of a single chip taking several months) and do not scale well with increase in transistor integration density. Moreover, the results of analyzing a sample cannot be extrapolated to the entire manufactured lot [3]. Since an adversary might affect only a small population of the manufactured ICs, destructive reverse engineering approaches cannot be effective for trust validation in ICs. The proposed *non-destructive* approaches can again be classified under two main heads: *non-invasive* and *invasive*. The *non-invasive* techniques leave the original design unaltered while the *invasive* techniques modify the design to embed features targeted towards Trojan detection.

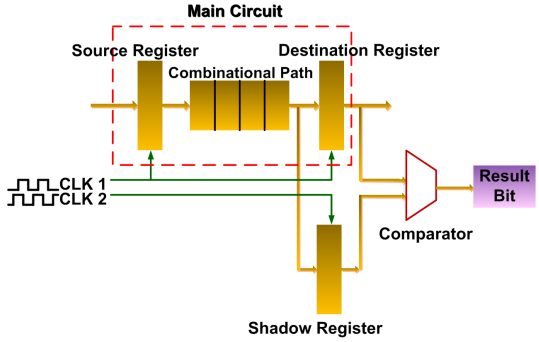
A. Invasive Trojan Detection Techniques

The invasive Trojan detection techniques can again be classified under two different headings - those which are directed towards *preventing* the insertion of Trojans during design or fabrication of an IC, and those which facilitate *detection* of inserted Trojans using post-manufacturing test. In [8], it was noted that Trojan insertion depends on the availability of free *dead space* within an IC layout, since the total area of the die cannot be altered by the adversary. However, if the adversary is capable of extracting the gate-level netlist from the layout, through logic optimization and better place and route techniques, it might be possible to free-up space to accommodate the Trojan. A design technique and associated design automation flow are proposed in [12] to prevent effective insertion of Trojans. Here, the original design is *obfuscated* through expansion of the reachable state space to make it difficult for an adversary to reverse-engineer the functionality of the circuit and to find the true rare events. Trojans inserted into a design without considering its true functional behavior become either invalid (i.e. not triggered in normal mode of operation) or easily detectable. The approach results in 24% improvement in Trojan detection coverage at less than 10% design overhead.

Design techniques can also be used to *assist* in Trojan detection by logic testing or side-channel measurements. Since an adversary is likely to exploit rare internal node conditions to construct a Trojan, a design approach that aims at increasing their controllability and observability can help in increasing Trojan detection coverage. In [13], each module in a design is modified such that a specific sequence of inputs activates an embedded FSM in the module which takes it to a special mode called the *transparent mode* (see Fig. 6(a)). In this mode, the controllability and observability of probable Trojan trigger and payload nodes is enhanced and a compacted signature is presented at the primary outputs, which indicates the presence



(a) On-demand Transparency Scheme [13]



(b) Shadow-latch Based Delay Characterization Technique [16]

Fig. 6. Examples of invasive Trojan detection techniques.

or absence of a Trojan. In [15], a design technique termed *VITAMIN* based on the inversion of the supply voltage of alternate logic levels in an IC is proposed. The logic behavior of a gate operating with inverted supply voltage is inverted during test mode. As a result, the activity of a rarely activated Trojan circuit is enhanced and it can be detected by comparing the power profiles of different ICs.

In [16], the authors propose a low-overhead "at-speed" delay characterization technique which is capable of detecting modifications to the circuit, both at run-time and at test-time. The path delay characterization is based on the insertion of "shadow latches" (see Fig. 6(b)) in the design to capture and compare with the data latched by registers in the original circuit paths. The test-time measurements are compared to the design-time projections and any substantial statistical difference indicates malicious design alteration. In [25], this method was shown to be capable of detecting Trojans in an 8×8 array multiplier circuit under $\pm 20\%$ process variations. A dummy flip-flop insertion technique to increase the trigger probability of Trojans was presented in [29], to aid in the detection of Trojans through *side-channel* techniques. It can also help in Trojan detection with logic testing by making the malicious effect of a Trojan observable at the primary output.

Another novel technique proposed in [14] is to use 3-D IC technology to integrate the security mechanisms in a separate plane (called the *control plane*) above an existing plane of circuitry in an IC (called the *computation plane*). The paper describes several security mechanisms to be performed by the control plane; however, it does not discuss the technical challenges and the design overhead.

B. Non-invasive Trojan Detection Techniques

In the *non-invasive* Trojan detection techniques, a Trojan is detected by comparing the behavior of the test IC with the golden IC instance or a golden functional model. They can be further classified into two main types: *run-time* and *test-time* techniques. The *run-time* techniques employ an online monitoring system that tries to detect suspicious activity during in-field operation, while the test-time techniques are aimed at detecting Trojan-infected chips before deployment.

1) Run-time, non-invasive Trojan detection approaches: In [6], the authors propose the addition of reconfigurable *Design for Enabling Security* (DEFENSE) logic in a given SoC to enable real-time functionality monitoring. The checks can be performed concurrently with the normal circuit operation and trigger appropriate countermeasures when a deviation from normal functionality is detected. However, the effectiveness and the hardware overhead associated with this scheme is not mentioned in the work. In [17], the authors propose a novel SoC bus architecture that can detect malicious bus behaviors associated with Trojan hardware, protect the system and system bus from them and report the malicious behaviors to the system CPU, without loss of bus performance. The authors report an additional gate-count of about 800 logic gates in a four million gate SoC, and negligible delay overhead.

In [18] the authors propose a scheme whereby functionally equivalent software instances are executed on multiple CPU cores, assisted by dynamic distributed software scheduling. The sub-task outputs from different cores are compared to dynamically evaluate their individual trust-levels, with the distributed scheduler undergoing a *trust learning* procedure for multiple runs. The authors show that the scheme is capable of successfully completing jobs in a Trojan infested environment, with improvement in throughput over successive runs.

A combined hardware-software approach to perform run-time execution monitoring has been proposed in [19, 35]. Here, a simple verifiable "hardware guard" module external to the CPU is considered. The work targets primarily DoS and *privilege escalation* attacks, using periodic checks by the operating system (OS) which is enhanced with *live check* functionality. The authors report 2.2% average performance overhead using SPECint 2006 benchmark programs, but do not report the hardware design overhead.

2) Test-time, non-invasive Trojan detection approaches: There are two main classes of testing based approaches for Trojan detection: (a) those based on logic testing, and (b) those based on the measurement of side-channel parameters such as power, delay, etc. The main advantage of the test-time techniques over the run-time techniques is that the test-time techniques incur no hardware overhead, while the main disadvantage is the requirement of a "golden" (i.e. Trojan-free) manufactured IC or functional model. Run-time methods typically involve considerable performance and power overhead, however, they provide the last line of defense and are capable of providing 100% confidence in computed results.

a) Logic testing-based approaches: The main challenge in a logic testing based approach is the enormously large *Trojan space*, which makes the generation of an exhaustive set of test vectors to detect all possible Trojans computationally infeasible. As an example, even with the constraint of maximum four trigger nodes and a single payload node, a

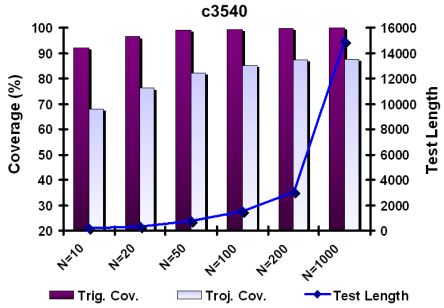


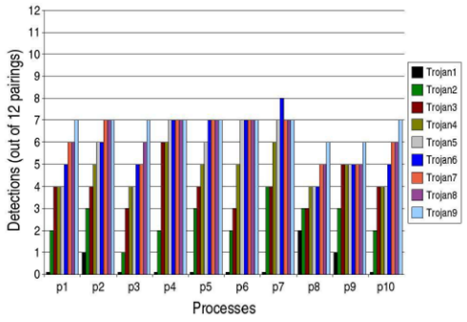
Fig. 7. Logic testing based technique: Impact of N (number of times a rare point satisfies its rare value) on the trigger/Trojan coverage and test length for an ISCAS-85 benchmark circuit [20].

small ISCAS-85 benchmark circuit $c880$ with 451 gates can have $\sim 10^9$ triggers and $\sim 10^{11}$ possible Trojan instances, respectively. Hence, a statistical approach seems intuitively more suitable for test vector generation in case of logic testing based Trojan detection.

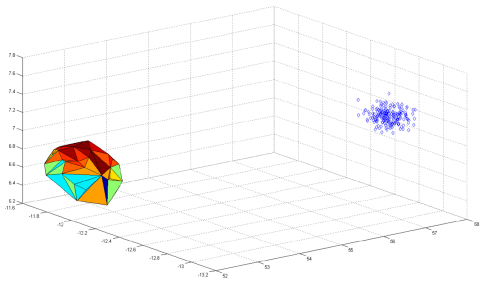
A randomization-based technique to probabilistically compare the functionality of the implemented circuit with the design of the circuit is described in [34]. It can also report a fingerprint input pattern to differentiate between the golden and the Trojan-infested circuit. In [20], a statistical vector generation approach for Trojan detection has been proposed that targets generation of an optimal set of test vectors that can trigger each rare node in a circuit to its rare value multiple times (N times, where N is a user-specified parameter), similar to the concept of *N-Detect* test [21]. By increasing the toggling of nodes that are random-pattern resistant, it improves the probability of activating a Trojan compared to purely random patterns. The coverage increases with N, with increasing test length (see Fig. 7). The procedure achieves an 85% reduction in test length compared to a weighted random pattern set, for similar Trojan detection coverage.

b) Side-channel analysis-based approaches: The side-channel analysis based Trojan detection approaches are based on observing the effect of an inserted Trojan on a physical parameter such as circuit current transient, power consumption or path delay. The advantage of these approaches lies in the fact that even if the Trojan circuit does not cause observable malfunction in the circuit during test, the presence of the extra circuitry can be reflected in some side-channel parameter. However, the main challenges associated with side-channel analysis are large process variation in modern nanometer technologies and measurement noise, which can mask the effect of the Trojan circuit, especially for small Trojans.

In [22], the authors introduced the concept of *IC fingerprinting*, where each IC instance is associated with a signature, called a "fingerprint" obtained by measurement of one or more side-channel parameters. From the analysis of power traces, used as the IC fingerprint in this work, the authors were able to identify Trojan instances with an equivalent area as small as 0.01% of the total size of the circuit, in the presence of $\pm 7.5\%$ random parameter variations. Another approach based



(a) Power-Supply Transient Based Trojan Detection [23]



(b) Path Delay Fingerprint Technique [24]

Fig. 8. Examples of side-channel based Trojan detection techniques.

on measurement of power-supply transient signal is described in [23], where signals from multiple power ports for several IC instances are obtained by a calibration process and subjected to statistical characterization. The technique was capable of detecting around 50% of activated Trojans and 30% of inactive Trojans in an ISCAS-85 benchmark circuit (see Fig. 8(a)).

In [27], the authors propose a test vector generation approach to maximize the activity in individual partitions of a circuit, while minimizing the activity of other partitions. Simulation results assuming a process variation of up to 7.5% were able to successfully detect most of the inserted Trojans. In [28], a *sustained vector technique* for Trojan detection is proposed, where each input test vector is repeated multiple times to ensure the reduction of extraneous toggles within the genuine circuit. This technique is shown to magnify the power profile difference between the original and the infected circuit by up to thirty times compared to previous approaches.

Path delays of output ports were used as the fingerprint in [24], with extensive characterization for process variations. The procedure could detect implicit payload Trojans occupying only 0.13% of the total area under 7.5% process variations (see Fig. 8(b)), while for explicit payload Trojans occupying 0.36% of the total area, the detection rate was 36%. Both path delay and leakage current were considered in a *gate-level characterization* technique proposed in [26], where the detection problem was formulated as a Linear Programming Problem (LPP). The authors show that the technique is capable of detecting Trojans in ISCAS-85 benchmark circuits with a high level of confidence; however, the scalability of the technique to large sequential designs is not addressed. Note that major-

TABLE I
ADVANTAGES AND DISADVANTAGES OF LOGIC TESTING AND SIDE CHANNEL TROJAN DETECTION APPROACHES

	Logic Testing Approach	Side–channel Approach
Pros	(a) Effective for small Trojans (b) Robust under process noise	(a) Effective for large Trojans (b) Test generation is easy
Cons	(a) Test generation is complex (b) Large Trojan detection challenging	(a) Vulnerable to process noise (b) Small Trojan detection challenging

ity of published side-channel approaches provide simulation verification results. Process variations, design marginalities and measurement noise depend on many parameters and are hard to model accurately. Hence, it is important to perform hardware validation of a side-channel approach to accurately analyze its detection sensitivity.

Table I summarizes the relative advantages and disadvantages of logic testing and side-channel approaches for Trojan detection. From this table, it is clear that the two approaches have complementary scope in terms of Trojan detection capability. Hence, approaches that combine the best of both worlds can be the most promising in terms of generic Trojan detection capability.

IV. SUMMARY

The issue of hardware Trojans and effective countermeasures against them have drawn considerable interest in recent times. In this paper, we have presented a comprehensive study of different Trojan types and discussed emerging methods of detecting them. Considering the varied nature and size of hardware Trojans, it is likely that a combination of techniques, both during design and test, would be required to provide acceptable level of security. Design-time approaches would span various levels of design descriptions. On the other hand, post-silicon validation would require a combination of logic and side-channel test approaches to cover Trojans of different types and sizes under large parameter variations. Major future challenges in this area would include developing detection mechanisms for analog Trojans which can implement numerous types of activation and observation conditions; an integrated metric to quantify the level of trust that combines both design and test time approaches; and an evaluation platform that analyzes a design to identify vulnerable regions and the impact of a design change on the level of achievable trust.

REFERENCES

[1] Semiconductor Industry Association (SIA), "Global billings report history (3-month moving average) 1976-March 2009", 2008. [Online]. Available: http://www.sia-online.org/galleries/Statistics/GSR1976-March09.xls.

[2] Defense Science Board, "Task force on high performance microchip supply", 2005. [Online]. Available: http://www.acq.osd.mil/dsb/reports/200502HPMSReportFinal.pdf.

[3] DARPA, "TRUST in Integrated Circuits (TIC) - Proposer Information Pamphlet", 2007. [Online]. Available: http://www.darpa.mil/MTO/solicitations/baa07-24/index.html.

[4] Australian Government DoD-DSTO, "Towards Countering the Rise of the Silicon Trojan", 2008. [Online]. Available: http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/9736/1/DSTO-TR-2220\%20PR.pdf.

[5] S. Adee, "The Hunt for the Kill Switch", *IEEE Spectrum*, May 2008.

[6] M. Abramovici and P. Bradley, "Integrated Circuit Security - New Threats and Solutions", *CSIIR Workshop*, 2009.

[7] F. Wolff *et al*, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", *DATE*, 2008.

[8] X. Wang, M. Tehranipoor and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions", *HOST*, 2008.

[9] Z. Chen *et al*, "Hardware Trojan Designs on BASYS FPGA Board (Virginia Tech)", *CSAW Embedded System Challenge*, 2008. [Online]. Available: http://isis.poly.edu/~vikram/vt.pdf.

[10] A. Baumgarten *et al*, "Embedded Systems Challenge (Iowa State University)", *CSAW Embedded System Challenge*, 2008. [Online]. Available: http://isis.poly.edu/~vikram/iowa_state.pdf.

[11] Y. Jin and N. Kupp, "CSAW 2008 Team Report (Yale University)", *CSAW Embedded System Challenge*, 2008. [Online]. Available: http://isis.poly.edu/~vikram/yale.pdf.

[12] R.S. Chakraborty and S. Bhunia, "Security against Hardware Trojan through a Novel Application of Design Obfuscation", *ICCAD*, 2009.

[13] R.S. Chakraborty, S. Paul and S. Bhunia, "On-Demand Transparency for Improving Hardware Trojan Detectability", *HOST*, 2008.

[14] T. Huffmire *et al*, "Trustworthy System Security through 3-D Integrated Hardware", *HOST*, 2008.

[15] M. Banga and M.S. Hsiao, "VITAMIN: Voltage Inversion Technique to Ascertain Malicious Insertions in ICs", *HOST*, 2009.

[16] J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", *HOST*, 2008.

[17] L.W. Kim, J.D. Villasenor and C.K. Koc, "A Trojan-resistant System-on-chip Bus Architecture", *Intl. Conf. on Military Communication*, 2009.

[18] D. McIntyre *et al*, "Dynamic Evaluation of Hardware Trust", *HOST*, 2009.

[19] G. Bloom, B. Narahari and R. Simha, "OS Support for Detecting Trojan Circuit Attacks", *HOST*, 2009.

[20] R.S. Chakraborty *et al*, "MERO: A Statistical Approach for Hardware Trojan Detection", *CHES Workshop*, 2009.

[21] I. Pomeranz and S.M. Reddy, "A Measure of Quality for n–Detection Test Sets", *IEEE. Trans. on Computers*, Nov. 2004.

[22] D. Agrawal *et al*, "Trojan detection using IC fingerprinting", *IEEE Symp. on Security and Privacy*, 2007.

[23] R.M. Rad, J. Plusquellic and M. Tehranipoor, "Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals", *HOST*, 2008.

[24] Y. Jin and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprint", *HOST*, 2008.

[25] D. Rai and J. Lach, "Performance of Delay-Based Trojan Detection Techniques under Parameter Variations", *HOST*, 2009.

[26] M. Potkonjak *et al*, "Hardware Trojan Horse Detection Using Gate-Level Characterization", *DAC*, 2009.

[27] M. Banga and M.S. Hsiao, "A Region Based Approach for the Identification of Hardware Trojans", *HOST*, 2008.

[28] M. Banga and M.S. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans", *VLSI Design*, 2009.

[29] H. Salmani, M. Tehranipoor and J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time", *HOST*, 2009.

[30] Y. Alkabani and F. Koushanfar, "Designer's Hardware Trojan Horse", *HOST*, 2008.

[31] S. King *et al*, "Designing and Implementing Malicious Hardware", *LEET*, 2008.

[32] L. Lin *et al*, "Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering", *CHES Workshop*, 2009.

[33] F. Kiamilev and R. Hoover, "Demonstration of Hardware Trojans", *DEFCON 16*, 2008.

[34] S. Jha and S.K. Jha, "Randomization Based Probabilistic Approach to Detect Trojan Circuits", *11th IEEE High Assurance Systems Engineering Symposium*, 2008.

[35] G. Bloom *et al*, "Providing Secure Execution Environments with a Last Line of Defense against Trojan Circuit Attacks", *Computers and Security*, 2009.

[36] Chipworks, Inc., "Semiconductor Manufacturing - Reverse Engineering of Semiconductor components, parts and process". [Online]. Available: http://www.chipworks.com

[37] J.A. Kash, J.C. Tsang and D.R. Knebel, "Method and Apparatus for Reverse Engineering Integrated Circuits by Monitoring Optical Emission", *United States Patent Number 6,496,022 B1*, 2002.

[38] S. Smith and J. Di, "Detecting Malicious Logic Through Structural Checking", *IEEE Region 5 Technical Conference*, April 2007.