Protection Against Hardware Trojan Attacks: Towards a Comprehensive Solution

Swarup Bhunia
Case Western Reserve University

Miron Abramovici
Independent Consultant

Dakshi Agrawal
IBM Research

Paul Bradley
Tiger's Lair

Michael S. Hsiao
Virginia Tech

Jim Plusquellic
University of New Mexico

Mohammad Tehranipoor
University of Connecticut

Editor's notes:
With the increasing disintegration of the design and manufacturing chain of our microelectronic products, we should not only worry about including unintentional, unwanted hardware features (''bugs''), but also about including intentional malicious hardware features: ''Trojan Horses,'' which act as spies or terrorists. This article provides an overview of hardware Trojans and countermeasures.

—Erik Jan Marinissen, IMEC, Belgium

Hardware Trojan attacks: The problem

The issue of trust is an emerging problem in integrated circuit (IC) security [1], [2]. It has become prominent recently due to widespread outsourcing of the IC manufacturing processes to untrusted foundries in order to reduce cost. An adversary can potentially tamper a design in these fabrication facilities to cause undesired change in IC functionality, integrity, or reliability through addition/deletion/alteration of the circuit structure, popularly referred to as *hardware Trojan attacks*. On the other hand, third-party computer-aided design (CAD) tools and hardware intellectual property (IP) modules increasingly used in a system-on-chip (SoC) design greatly enhance the vulnerability to malicious design modifications [2]. Trojan attacks are intended to affect normal circuit operation, potentially with catastrophic consequences in critical applications in the domains of communications, space, military, and nuclear facilities. They can also aim at leaking secret information from inside a chip through covert channels or affect the reliability of an IC through undesired process changes that cause device/interconnect wear-out and long-term reliability issues [1]. Furthermore, they can be used to assist software attacks by providing hardware back-door. Broadly, two types of Trojans can be inserted in a digital circuit: combinational Trojans, activated by a rare combination of internal node values; and sequential Trojans, activated through a sequence of rare events. Figure 1a shows a circuit with a Trojan instance along with example of combinational and sequential Trojans, which are designed to cause malfunction under a specific condition. Figure 1b shows example Trojan which aims at leaking secret key from inside a processor and can be exploited by

Digital Object Identifier 10.1109/MDT.2012.2196252
Date of publication: 25 April 2012; date of current version: 23 September 2013.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

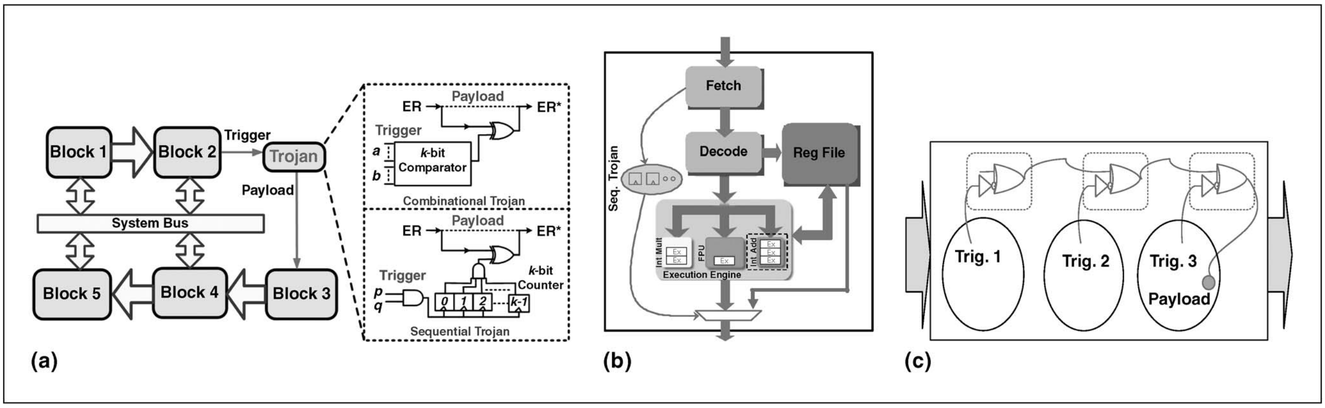

Figure 1. Hardware Trojan attacks of different forms: (a) combinational and sequential Trojans causing malfunction; (b) a Trojan with capability of leaking secret information from inside a processor and can be exploited by software; and (c) a Trojan with distributed trigger condition.

either the software or input data. The Trojan circuit can be localized or distributed in a chip. An example of Trojan with distributed trigger condition is presented in Figure 1c. Note that a trigger condition or the payload of a Trojan can be either digital or analog, e.g., a Trojan can be triggered by analog condition such as temperature.

A recently reported Trojan attack involves U.S. military, who discovered a hardware "back-door" in a microchip used in everything from missiles to transponders. If left undiscovered, the chips could have been hacked and caused disastrous consequences in the event of war (e.g., shutting off a missile) [15]. A compelling aspect of this threat model lies in growing vulnerability of modern microchips toward these attacks, as exposed by researchers as well as hackers. With decreasing control of the IC design and fabrication steps, likelihood of Trojan attacks of various forms is rapidly increasing. Hence, there is a critical need to develop low-cost design and test solutions that provide comprehensive protection against these attacks and thus enable trusted field operation of modern ICs.

Overview of protection approaches

Conventional postmanufacturing test using functional/structural/random patterns cannot reliably detect hardware Trojans. This is because manufacturing test generation and application aim at detecting defects that cause deviation from functional or parametric specifications. They do not aim at detecting additional functionalities or deviation in circuit behavior triggered by rare events. Reliable detection of hardware Trojan using postsilicon validation involves some major challenges. First, an adversary can exploit inordinately large number of Trojan instances of varying forms and sizes [1], [2]. Second, due to their stealthy nature, activating arbitrary Trojan instances and observing their effects can be extremely difficult. Hence, deterministic and exhaustive validation approaches appear infeasible.

Existing research efforts for protection against Trojan attacks have focused on both design and validation techniques. Figure 2a shows broad classification of the protection techniques that apply at different stages of IC lifecycle. The design approaches make hard-to-detect Trojan insertion difficult or facilitate detection during postsilicon validation [3], [6]. Postmanufacturing Trojan detection approaches [1], [2] can be classified into two categories. Destructive testing by depackaging, demetallization, and microphotography-based reverse engineering of a chip is highly expensive and may not work if an attacker selectively tampers only a subset of the manufactured ICs [1]. Logic testing approaches, both functional and structural, aim at developing directed test patterns to activate Trojan instances and propagate their effects to output ports [5]. Although robust under process and measurement noise, these approaches are generally not effective in triggering large Trojans consisting of complex combinational or sequential triggering conditions [5]. An alternative approach is to measure physical side-channel parameters, such as supply current [8], [9] or path delay [4], which can manifest unintended design modifications. However, the effectiveness of side-channel analysis is reduced by

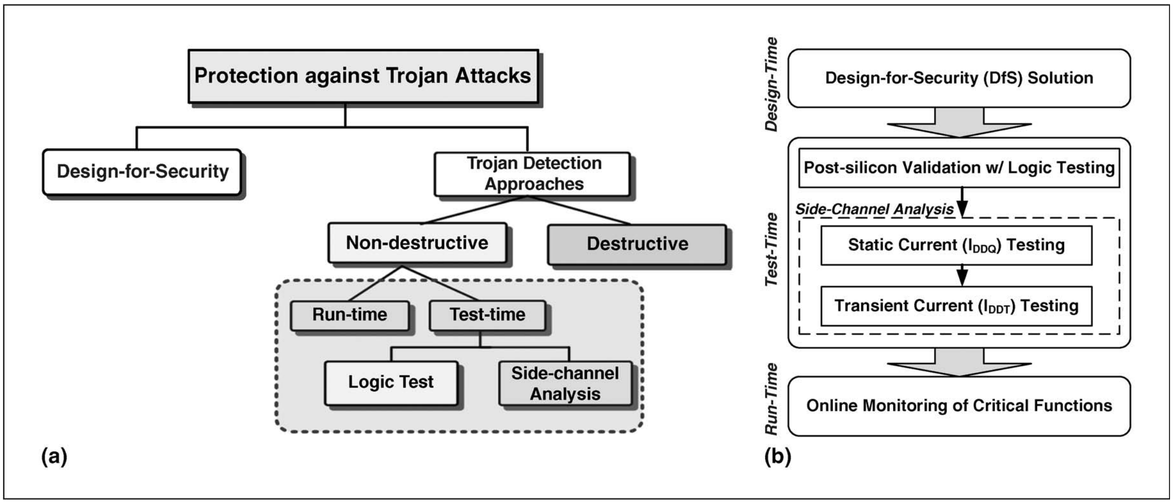
Protection against Trojan Attacks
Design-for-Security
Trojan Detection Approaches
Non-destructive
Destructive
Run-time
Test-time
Logic Test
Side-channel Analysis
(a)
Design-Time
Design-for-Security (DfS) Solution
Post-silicon Validation w/ Logic Testing
Side-Channel Analysis
Test-Time
Static Current (I_{DDQ}) Testing
Transient Current (I_{DDT}) Testing
Run-Time
Online Monitoring of Critical Functions
(b)

Figure 2. (a) Taxonomy of design and test techniques for protection against Trojan attacks. (b) An integrative protection approach that combines the benefits of design, test, and online monitoring solutions.

large device parameter variations in nanoscale process technologies, which can mask the effect of small Trojans. Table 1 provides a comparison of two major validation paradigms. Nondestructive validation can also be performed at runtime [1], using online monitoring of critical circuit operations. Such approaches provide a last line of defense against Trojan attacks.

Table 2 compares the Trojan detection capability of alternative approaches. Logic testing and side-channel-analysis-based validation provide complementary capabilities in detecting Trojans of different types and sizes. Hence, a postmanufacturing validation approach that combines their benefits can be effective in maximizing the Trojan coverage. For applications which require the highest level of assurance against a Trojan attack, we need to combine postmanufacturing validation with online monitoring. Finally, validation approaches, both postmanufacturing and online, need to be complemented with low-cost design-for-security (DfS) solutions, which harden a design with respect to Trojan insertion or facilitate Trojan detection during validation. Based on these observations, we propose an integrative protection approach, as illustrated in Figure 2(b), which combines the benefits of different protection approaches and provides comprehensive coverage against Trojans of all types and sizes.

Table 1 Comparison of logic testing and side-channel-analysis-based Trojan detection.

	Logic Testing	Side-Channel Analysis
Pros	• Robust under process noise • Effective for ultra-small Trojans	• Effective for large Trojans • Easy to generate test vectors
Cons	• Difficult to generate test vectors • Large Troj. detection challenging	• Vulnerable to process noise • Ultra-small Troj. Det. challenging

Design for Trojan detection

Trojan detection during postmanufacturing test needs to address major challenges due to rare activating nets in the circuit, process variations, and measurement noise [3]. To improve the effectiveness of these detection methods, ICs should be designed with these detection strategies in mind. We outline two DfS approaches that can enhance Trojan detection using logic testing as well as side-channel signal analysis [8].

Removing rare-triggered nets

Trojan circuits are stealthy and are typically triggered by rare conditions. Hence, improving Trojan detection can be tackled in two different ways: 1) generating deterministic test patterns intelligently to detect the impact of Trojans on design characteristics beyond process and environmental variations, which, however, is extremely challenging since the location, type, and size of the Trojans are unknown; and 2) changing the design such that random test patterns can effectively activate Trojans.

Table 2 Capability of Trojan detection schemes to identify different hardware Trojan types and sizes.

Troj. Size		Large		Small	
Troj. Type		Localized	Distributed	Localized	Distributed
Digital	Combinational	Logic Testing	Side-Channel (IDDT) + Online	Logic Testing + Online	Side-Channel (IDDT)
	Sequential	Side-Channel	Side-Channel (IDDQ)	Side-Channel (IDDT)	Side-Channel (IDDT)
Analog		Side-Channel (IDDT, IDDQ, Others) + Online			

The stealthy nature of Trojans suggests that they are activated only under very rare conditions, e.g., a rare circuit state, certain temperature, or noise to avoid accidental detection using structural or functional patterns. As an example, a Trojan can have $q > 1$ trigger inputs which can be 1) nets with very low transition probabilities, or 2) rare combinations of multiple nets. When the transition probability of Net$_i$ is very low, either $P_i(0) \gg P_i(1)$ or $P_i(1) \gg P_i(0)$. With q number of trigger inputs, the probability of generating a specific trigger vector is $P_{\text{trigger}-\text{vector}} = \prod P_i$ ($i = 1$ to q). Here, we assume statistical independence between vectors applied by scan architecture. It is expected that $P_{\text{trigger}-\text{vector}}$ is very low if P_i's are low. By increasing the transition probability of nets with low transition rate, it is possible to eliminate hard-to-activate sites in a design. Note that scan architecture allows access to internal cells of the circuits, thereby improving controllability and observability of internal nodes. To remove hard-to-activate sites, dummy scan flip-flops, depicted in Figure 3, can be inserted to increase transition probability of design nets with transition probability less than a threshold (P_{th}) [6]. Note that such design modifications can be done even if the low-level netlist is not trusted.

The probabilities of "1" and "0" at the output of scan flip-flop and primary inputs are assumed 1/2 if a random pattern is applied. Thus, by supplying internal nets with equal "1" and "0" probabilities, the transition probabilities on target nets can be increased. In [6], it is proven that by inserting dSFF–AND, shown in Figure 3, when $P_i(0)$ of a net is much lower than its $P_i(1)$, the transition probability of the net can be increased. Dummy scan flip-flop insertion can increase the ratio of Trojan to circuit power consumption by increasing activity in Trojan circuit, which increases the numerator of the ratio.

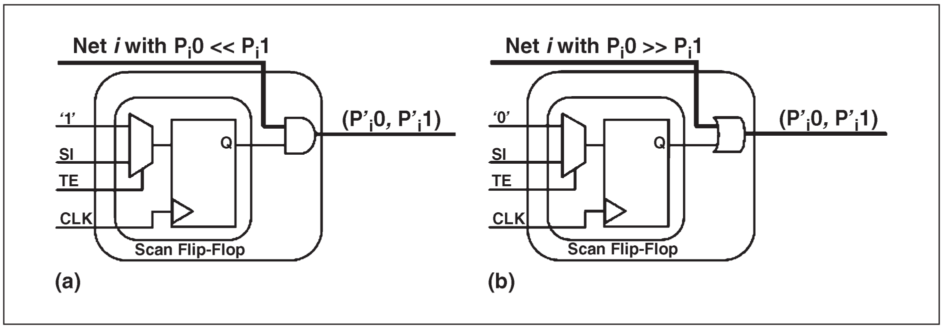

Figure 3. The dummy flip-flop structures when (a) $P_i(1) \gg P_i(0)$ and (b) $P_i(0) \gg P_i(1)$.

The proposed scheme will be resilient against various tampering and removal attacks. For instance, the attacker may use the scan enable signal (i.e., test control) as a trigger for Trojans. This will make the Trojans stay quiet during test mode. However, we can target the Trojans during capture mode of the test process. We have demonstrated that this technique would still be very effective in detecting Trojans if we were to switch between the shift mode and the capture mode [16].

Increasing localized switching

Minimizing normal circuit switching with respect to Trojans increases Trojan-to-circuit-activity (TCA) ratio, defined as the ratio of activity inside a Trojan circuit to circuit activity. This would significantly increase the probability of detecting smaller Trojans whose impact on circuit power is small or negligible. The total power consumption of the circuit under test is highly correlated with the total number of transitions in the scan cells during scan-based pattern application. During scan insertion, scan cells are grouped into a number of scan chains based on different criteria. It is possible to reorder scan cells based on their final physical location in the layout. Layout-aware scan-cell reordering can localize switching activity to one region while limiting it in other regions in a design [7]. It obtains

Table 3 The percentage of switching activity in each region of s38417 and s35932 benchmark circuits after running four simulations.

Benchmark Region #	s38417	s35932
1	15.7%, 15.8%, 15.2%, 15%	11.6%, 11.4%, 11.9%, 11%
2	7.1%, 7.2%, 7.2%, 7%	8.3%, 7.5%, 8.2%, 6.8%
3	9.7%, 9.3%, 9.6%, 9.5%	10.3%, 9.9%, 10.4%, 9%
4	67.5%, 67.7%, 69%, 68.3%	69.7%, 71%, 69.3%, 73%

placement information of scan cells and restitches the scan chains based on the physical information and the number of regions (N). Finally, the netlist is updated with restitched cells for routing.

Table 3 shows the effectiveness of scan-cell reordering in limiting switching activity in any target region for *s*38417 and *s*35932 benchmarks. Scan cells in both benchmarks are grouped into $N = 4$ scan chains using layout-aware scan-cell reordering. The simulation is run four times, and a total of 132 random patterns are applied to the circuits. Patterns apply random "0" and "1" to the scan chain covering the target region (region 4) while "0" is applied to all other scan chains. The percentage of activity in each region is reported in the table as (Run1, Run2, Run3, Run4). The results clearly indicate that, in all four runs, switching activity is mostly limited to region 4, while the other regions are kept fairly inactive in both benchmarks. This will significantly increase the TCA, thus improving the detection probability. Our results show that this technique is very effective for small and large Trojans as well as distributed and localized Trojans [15], since the reduction in circuit switching is substantially larger than the reduction in Trojan's switching even if the attacker distributes the Trojan gates among different regions in the circuit.

Trojan detection using logic testing

The design approaches described in Section III facilitate Trojan detection through both functional testing and side-channel approaches. A functional or logic testing approach can exploit the improved internal node characteristics of the modified design to optimize the test length or Trojan coverage. As mentioned earlier, deterministic test generation for logic testing is infeasible since the Trojan space can be inordinately large due to its combinatorial dependence on the circuit nodes. As an example, for four trigger and single payload nodes, a small ISCAS-85 circuit *c*880 with 451 gates can have $\sim 10^{11}$ distinct Trojan instances. This indicates that instead of an exact approach, a statistical approach can be computationally more tractable.

We propose to use a statistical logic testing approach [5] with the objective to derive a set of test patterns that is compact (minimizing test time and cost), while maximizing the Trojan detection coverage (estimated as the percentage of random Trojan instances detected by a vector set [5]). The basic concept is to detect low probability conditions in the design at the internal nodes and then derive an optimal set of vectors that can trigger each of these nodes individually to their rare logic values multiple times (e.g., at least N times, where N is a user-defined parameter). By increasing the toggling of nodes that are random-pattern resistant, it improves the probability of activating an unknown Trojan compared to purely random patterns. It does not require a trusted design environment, i.e., the test generation can be performed on a tampered design. Since the proposed detection is based on functional validation using logic values, it is robust with respect to parameter variations and can reliably detect very small Trojans, both combinational and sequential.

Side-channel analysis approaches

In this section, we propose two side-channel approaches for Trojan detection using supply current. The approaches aim at identifying the Trojan effect in either transient (I_{DDT}) or static current (I_{DDQ}) considering the effect of process noise.

Transient current analysis

The goal of transient current analysis is to detect switching activity inside a Trojan circuit. As the changes in transient current could be very small—indeed that would be the goal of an adversary—any detection mechanism needs to carefully consider sources of natural variations in current flow and discount the influence of such sources. Figure 4a illustrates the device parameter variations, both die-to-die and within-die, in a nanometer process and corresponding impact in two side-channel parameters: I_{DDT} and $F_{\max}$ (maximum operating

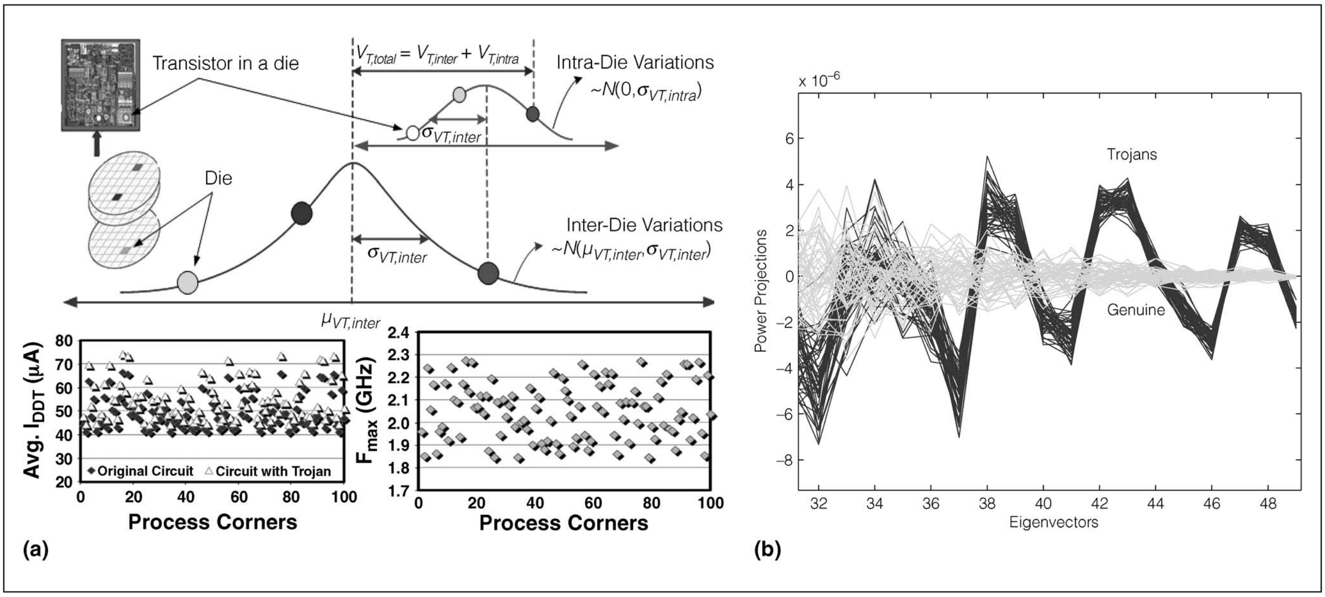

Figure 4. (a) Die-to-die and within-die variation in a device parameter (threshold voltage or V_T) and corresponding impact in side-channel parameters: transient supply current (I_{DDT}) and maximum frequency (F_{max}). (b) Side-channel transient current signal from a Trojan can be separated and identified using Karhunen–Loève expansion.

frequency). Clearly, intrinsic variations can mask the effect of a Trojan in side-channel parameters. Hence, the variability in silicon manufacturing process as well as measurement process needs to be discounted to isolate the effect of small Trojans.

The variability in the measurement process can be minimized by standardizing the measurement setup and specifying tight tolerances on the ambient environment, test harness, probes, and other equipment. However, regardless of the tightness of tolerances, inherent thermal noise present in the circuits still introduces variability that may mask contribution of a Trojan in the transient current. Fortunately, the thermal process is well understood, and it can be modeled as a zero-mean random noise process, and therefore, can be eliminated by taking average of a large number of measurements from the IC under test.

The variability in the manufacturing process is much more difficult to discount. As the process variations are intrinsic and constant for a given IC, they cannot be eliminated by averaging multiple measurements. Thus, the observed current has variability due to two possible sources: 1) process variation; and 2) Trojan circuit, if any. Once the variability has been isolated (by subtracting the average of a large number of measurements done on different genuine circuits), the detection of a Trojan circuit becomes a standard hypothesis testing problem [10]:

$$\begin{aligned} H_G &: n_p(t; I; C) \\ H_T &: n_p(t; I; C) + \tau(t; I; C) \end{aligned}$$

where H_G (genuine circuit) and H_T (circuit with Trojan) are the two hypotheses under test, n_p is the side-channel signal due to process variability, and τ is the signal contributed by the Trojan. Note that the side-channel signal is a waveform (a function of time t), and both possible components of it depend on the intrinsic chip characteristics I and the test calculation being performed C.

The key to Trojan detection is to realize that a statistical distribution of the process noise n_p can be obtained by profiling side channels from several genuine ICs. Thus, as long as the statistical distribution of the Trojan side-channel signal τ has components in a subspace that is not spanned by the process noise distribution n_p, Trojan presence can immediately be detected by side-channel components that are orthogonal to the process noise signal. The algebra of computing these signal subspaces and components can be derived from the standard techniques, including the Karhunen–Loève expansion [10]. Figure 4(b) illustrates how the side-channel signal from a Trojan can be separated and identified using Karhunen–Loève expansion. The *x*-axis is discrete with each integer representing a signal subspace, and *y*-axis is continuous denoting the strength of a signal in a particular subspace. It shows signal strengths for genuine circuits and circuits with Trojan using green (light) and blue (dark) curves, respectively. Clearly, in signal subspace number 43, 46, and 48, Trojan contributions far outstrip any genuine process noise and process noise variability, thus revealing existence of the Trojan. We note that the proposed analysis can identify a 3-b it comparator Trojan claiming 0.01% of total circuit area under 7.5% parameter variations. For a more detailed setup of the synthesized RSA circuits and experimental methodology used to produce these curves, refer to [8].

Static current analysis

An important requirement for Trojan detection through transient current analysis is to induce activity inside a Trojan circuit. While efficient test generation, as described in Section VI, target amplifying activity for arbitrary Trojans, they cannot encompass Trojans of all forms and sizes. To deal with this challenge, we propose a side-channel approach that isolates the effect of a Trojan in static current (I_{DDQ}) and hence does not require Trojan activation. Besides, it makes use of simple on-chip hardware primitive to mitigate the adverse effects of process variations and leverages the multiple supply ports (MSPs) on a chip to improve signal-to-noise ratio (SNR) [9]. MSPs are incorporated because the metal defining the power grid has a finite resistance. One benefit of this parasitic resistance is that it creates regional current behavior, i.e., the current behavior through each of the supply ports is unique and is largely influenced by transistors that are topologically close to the supply port. Therefore, regional observability is possible by measuring the I_{DDQ}'s from each of these supply ports separately.

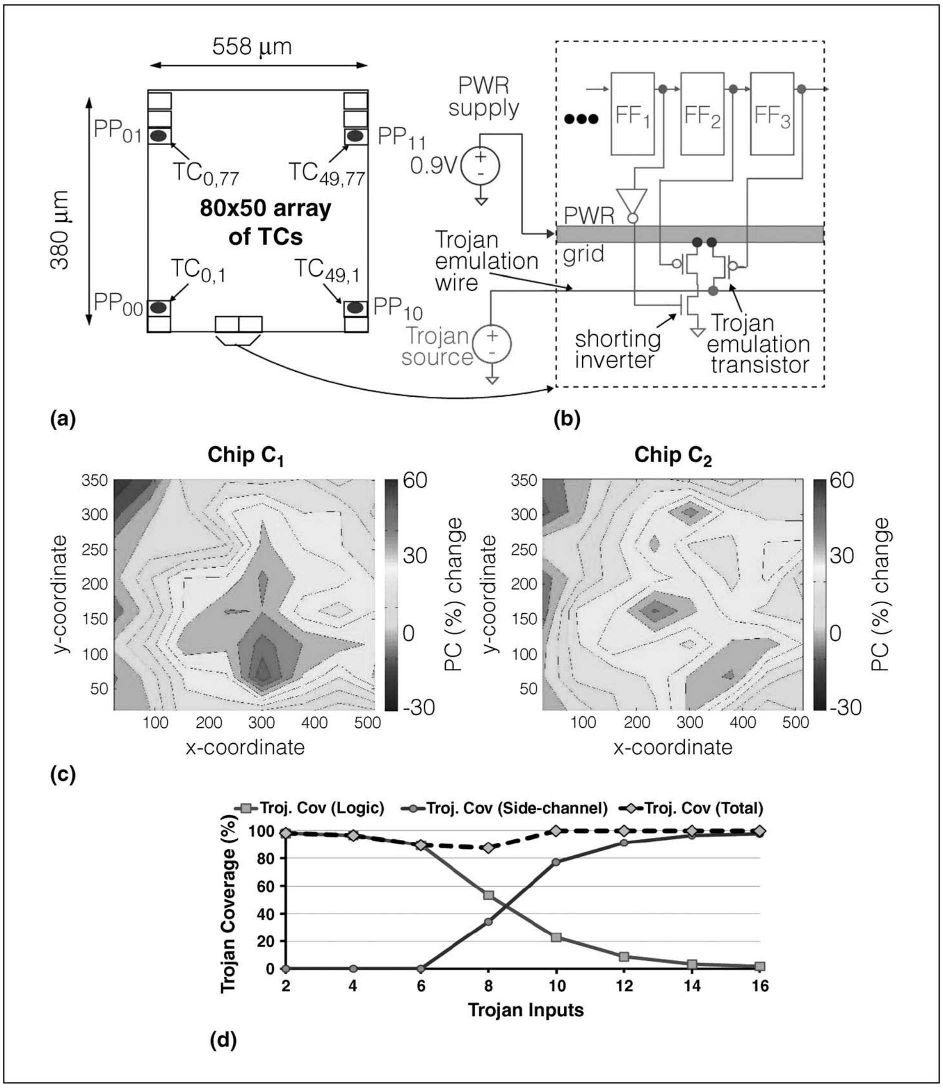

Figure 5. (a) Block diagram of the test chip. (b) Details of the test circuit. (c) Within-die leakage current contour plots. (d) An integrated validation approach to increase Trojan coverage.

We have observed the effectiveness of the proposed approach using hardware experiments on a test chip, schematically shown in Figure 5a, which allowed access to the individual power ports, labeled PP_{00} through PP_{11}. The core logic of the test chip is an 80 × 50 array of test circuits (TCs), the details of which are shown in Figure 5b. Each TC consists of three FFs connected in a scan chain configuration, a shorting inverter, and a Trojan emulation transistor connected to a globally routed Trojan emulation wire.

The process calibration method that we propose uses the shorting inverter to create a resistive short on the power grid at locations directly underneath the power ports. Details of the calibration process can be found in [11]. The Trojan emulation wire and transistor are used to introduce small current anomalies that are designed to emulate the leakage current from the Trojan gates added to the layout. By enabling one of the Trojan emulation transistors in the array, a current anomaly can be introduced at any point on the power grid. The objective is to measure the branch currents with a Trojan emulated, calibrate the measured currents, and apply a statistical technique to determine if the anomaly is detectable.

A key benefit of calibration is that it eliminates the adverse effects of die-to-die resistance variations in the power grid. However, leakage variation is another major challenge to achieving high levels of SNR in advanced technologies. Figure 5c shows the leakage profiles of two chips plotted as percentage change in contour plots. From the patterns, it is clear that the leakage characteristics of the two chips are very different. These leakage variations can appear as "current anomalies" in the MSP measurements for a chip, producing false positives. Fortunately, the massively connected nature of the power grid tends to "average out" the local leakage variations, reducing their adverse impact on detection strategies.

In order to evaluate the MSP technique in combination with calibration, we carried out three sets of experiments. In each experiment, 90 emulated Trojans were investigated in each chip. We have 45 copies of the test chips, and, therefore, the total number of emulated Trojans is 4050. The first analysis is designed to model the traditional application of I_{DDQ} testing in which the global currents are used in a 1-D statistical outlier detection technique. The analysis shows that only one chip produced outliers, and in that chip, only 45 of the 90 Trojans were detected, yielding 1.1% detect ratio. On the other hand, the ratio using MSP and a regression-based (2-D) statistical analysis is 7.2% using uncalibrated data and 53.8% using calibrated data. These are substantial increases in detection sensitivity over the results obtained using traditional I_{DDQ} test methods. Moreover, the test chip is very small ($558 \times 380\ \mu\mathrm{m}^2$) in comparison to product chips, which would have several hundred power ports. The additional power ports would increase the modest 7.2% detection ratio (and, correspondingly, the 53.8%) substantially, with expected gains of about two and three orders of magnitude compared to traditional single-port methods.

Figure 5d compares the Trojan coverage between transient-current-based and logic-testing-based validation. It shows that effectiveness of postsilicon validation can be significantly enhanced by integrating both approaches, since they provide complementary strengths.

Test generation: A region-based approach

The design techniques described in Section III help side-channel-analysis-based Trojan detection by increasing activity inside a Trojan circuit. They, however, need to combine with efficient Trojan-aware test generation techniques that further improve the Trojan activity to maximize the detection sensitivity. In this section, we propose such a technique. We partition the circuit into smaller subcircuits that we call "regions." A "radius" defines the extent of a "region," as shown in Figure 6a. For a gate, the region around it comprises all the transitive fan-in and fan-out gates that are within the defined radius. The "regions" are restricted across clock boundary. Once we have identified the regions, we attempt to create an activity on a per-region basis [12]. The Trojan is most detectable when the power consumed in the entire genuine circuit is kept low [8], but at a nonzero value. Thus, we aim to stimulate activity within a small region while keeping the rest at low or zero activity. If the Trojan is connected to portions of one or more such regions, the circuit activity in the genuine chip will be different from the tampered one, owing to the extra activity of the "Trojan" portion.

Sustained vector technique. Circuit activity can be induced in two ways: 1) changing inputs; and 2) changing state. While the primary inputs are fully controllable, the state bits are not. In order to limit the switching activity within the circuit, we can restrict the input variations to an extent such that the state bits are the only factor inducing toggles. This is achievable by sustaining the same vector at the

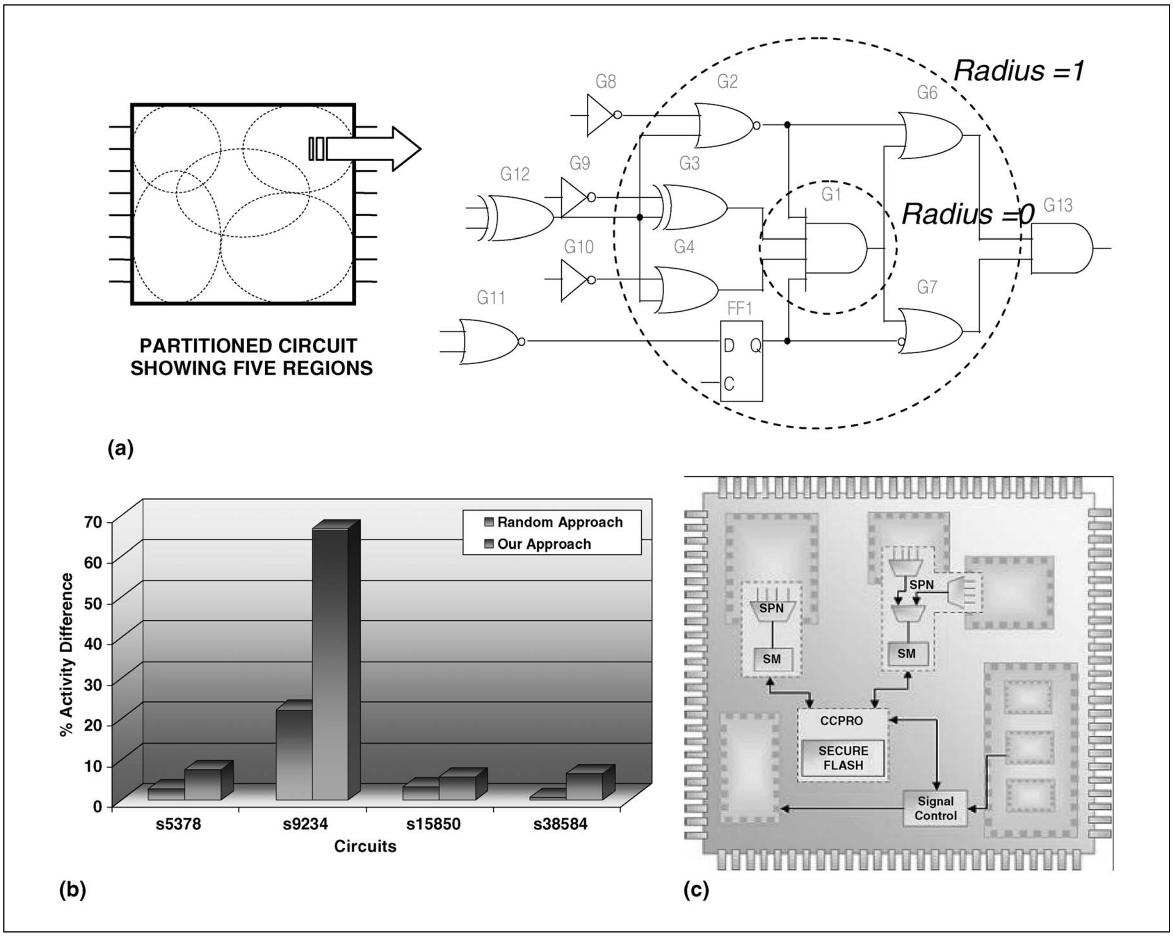

Figure 6. (a) Illustration of the concept of "region" and "radius" in a circuit. (b) Power differential between Trojan-infected and genuine circuits. (c) Runtime monitoring of the Trojan effect using reconfigurable IP infrastructure.

input pins over multiple clock cycles [13]. In addition, we search for a sustained vector that also introduces few toggles in the state variables. If we ensure to avoid any toggles at the primary inputs while simultaneously limiting the activity in the state variables, it helps us reduce the circuit activity to a good extent. Results, as shown in Figure 6b, indicate that the approach is able to generate the right functional vectors that can produce much higher power differential compared with the random vectors. With these higher power differentials, the technique is able to detect the Trojan with very high confidence.

Runtime monitoring approaches

Although detecting Trojans before ICs are deployed in the field is highly desirable, the existing techniques cannot guarantee this ideal outcome. For example, a Trojan inserted in a soft IP core and designed for late activation may not be detected during presilicon verification or during silicon validation. Furthermore, Trojan detection techniques analyze different functional/physical characteristics of an IC with respect to its golden-reference model. However, the presence of a Trojan in the RTL model of the device precludes having a golden model. Even if an RTL golden model does exist, such a model covers only the functional logic. Insertion of infrastructure logic in the design provides many additional opportunities for inserting hidden Trojans. Therefore, we must complement predeployment techniques with postdeployment monitoring of ICs during normal system operation. This monitoring is intended to identify unexpected or illegal behavior created by a Trojan with security monitors (SMs) embedded into the IC during its design.

The main problem is efficiently implementing a large number of security checks with limited hardware resources. We propose a solution that relies on reconfigurable instruments that are repeatedly reconfigured to dynamically implement different security checks. Each check detects an unexpected or illegal behavior created by a Trojan. Each reconfigurable instrument has a configuration register that determines its current function.

Figure 6c outlines a SoC designed with SMs. An SM is a programmable transaction engine configured to implement finite state machines (FSMs) that check the behavior of signals of interest. Signal probe networks (SPNs) are configured to select a subset of the monitored signals and transport them to SMs. An SPN is a distributed pipelined MUX network designed to support multiple clock domains. The configuration and control processor (CCPRO) reconfigures SPNs to select the groups of signals to monitor and reconfigures SMs to analyze the selected signals. All the configurations are stored in a nonvolatile (flash) memory inside the CCPRO. A signal control block allows the CCPRO or an SM to override a system signal. This feature is the basis of deploying countermeasures when one of the SMs detects a security violation.

The SMs perform two types of checks: 1) a set of user-specified security violations, such as an attempt to access a restricted address space or entering test/debug modes during normal operation; and 2) checks consisting of the general correctness properties of the system behavior, usually expressed as assertions. Both type of checks can be implemented based on the design specifications and do not require a golden implementation model. An activated Trojan will make the system operate in an incorrect way, which can be detected by assertion checks. It has been shown that a small number of assertions can achieve very good online transient fault coverage [14]. This means that a large number of distinct faults caused the same assertion(s) to fail. This result can be generalized to incorrect or illegal operations caused by security violations. The security checks run continuously one-group-at-a-time during normal operation of the IC. The groups are incrementally reconfigured so that most checkers are always active. Some checks are self-tests that validate that the monitoring logic is working correctly.

Reconfigurability allows a large number of checks to timeshare the same hardware resources. The number of checks is limited only by the size of memory used to store configurations. Reconfigurability also enables field upgrades to implement new checks to deal with newly discovered security threats. In a powered-off chip, the reconfigurable logic is "blank" (like an unprogrammed FPGA) and thus its function is concealed from attackers trying to reverse engineer the device.

The CCPRO or an SM can initiate basic countermeasures in response to detecting an attack by overriding specific functional signals. Examples of countermeasures include erasing sensitive data from memories, disabling a block exhibiting illegal behavior (by suspending its clock or activating its reset), or even disabling the operation of the entire IC. Countermeasures are specified by the user at the design stage. The monitoring logic does not impact the performance of the circuit, and overriding a signal introduces a MUX delay in the path of the signal. The area overhead is user controlled; typically, monitoring critical control signals and transactions can be done with less than 4% overhead.

The threat of hardware Trojan attacks is escalating with increasing complexity of modern SoCs and intrusion of untrusted third-party tools/IPs/facilities in the IC lifecycle. At the same time, new and more complex attack models, such as the ones which take advantage of nexus between design and fabrication stages, or hardware Trojans (e.g., back-door) exploited by software, are emerging. A "silver bullet" solution which can reliably protect against Trojan attacks of all forms and sizes is extremely difficult to achieve. On the other hand, an integrative solution which combines the complementary benefits of design, test, and monitoring solutions can provide the highest level of trust. In this paper, we have analyzed the IC security issue due to hardware Trojan attacks and presented a comprehensive solution, which integrates a Trojan-aware design approach with a postsilicon validation consisting of a logic-testing- and supply-current-based side-channel analysis and online monitoring of critical circuit functions. We have presented effective techniques for test generation and sensitivity improvement under process variations. Future work would focus on developing a unified trust metric, eliminating the need for golden ICs for predeployment detection, and discovering new forms of Trojan attacks. ■

References

[1] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, ‘‘Hardware Trojan: Threats and emerging solutions,’’ in *Proc. IEEE Int. High Level Design Validation Test Workshop*, 2009, pp. 166–171.

[2] M. Tehranipoor and F. Koushanfar, ‘‘A survey of hardware Trojan taxonomy and detection,’’ *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan.–Feb. 2010.

[3] J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri, ‘‘Design and analysis of ring oscillator based design-for-trust technique,’’ in *Proc. IEEE Very Large Scale Integr. (VLSI) Test Symp.*, 2011, pp. 105–110.

[4] D. Rai and J. Lach, ‘‘Performance of delay-based Trojan detection techniques under parameter variations,’’ in *Proc. IEEE Int. Workshop Hardware-Oriented Security Trust*, 2009, pp. 58–65.

[5] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, ‘‘MERO: A statistical approach for hardware Trojan detection,’’ in *Proc. Workshop Cryptograph. Hardware Embedded Syst.*, 2009, pp. 396–410.

[6] H. Salmani, M. Tehranipoor, and J. Plusquellic, ‘‘A novel technique for improving hardware Trojan detection and reducing Trojan activation time,’’ *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 112–125, Jan. 2011.

[7] H. Salmani and M. Tehranipoor, ‘‘A layout-aware approach for improving localized switching to detect hardware Trojans in digital integrated circuits,’’ *IEEE Trans. Inf. Forensics Security*, pt. 1, vol. 7, no. 1, pp. 76–87, Feb. 2011.

[8] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, ‘‘Trojan detection using IC fingerprinting,’’ in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 296–310.

[9] J. Aarestad, D. Acharyya, R. Rad, and J. Plusquellic, ‘‘Detecting Trojans through leakage current analysis using multiple supply pad I_{DDQ}s,’’ *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 893–904, Dec. 2010.

[10] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York, NY, USA: Wiley, 1968.

[11] D. Acharyya and J. Plusquellic, ‘‘Calibrating power supply signal measurements for process and probe card variations,’’ in *Proc. IEEE Int. Workshop Current Defect Based Test.*, 2004, pp. 23–30.

[12] M. Banga and M. Hsiao, ‘‘A region based approach for the detection of hardware Trojans,’’ in *Proc. IEEE Symp. Hardware-Oriented Security Trust*, 2008, pp. 43–50.

[13] M. Banga and M. Hsiao, ‘‘A novel sustained vector technique for the detection of hardware Trojans,’’ *Proc. Int. Conf. VLSI Design*, 2009, pp. 327–332.

[14] V. K. Reddy, A. S. Al-Zawawi, and E. Rotenberg, ‘‘Assertion-based microarchitecture improved fault tolerance,’’ in *Proc. Int. Conf. Comput. Design*, 2006, pp. 362–369.

[15] A. Rawnsley, ‘‘Fishy chips: Spies want to hack-proof circuits,’’ *Wired*, Jun. 24, 2011. [Online]. Available: http://www.wired.com/dangerroom/2011/06/chips-oy-spies-want-to-hack-proof-circuits/

Swarup Bhunia s an Associate Professor of Computer Engineering at Case Western Reserve University, Cleveland, OH, USA. His research interests include low-power and robust design, hardware security and protection, adaptive nanocomputing, and novel test methodologies. He has a PhD in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. He is a Senior Member of the IEEE.

Miron Abramovici is an independent consultant. As chief Scientist at Tiger's Lair, Vienna, VA, he developed new techniques to prevent IC tampering and counterfeiting. Before that, he was CTO at DAFCA and Distinguished Member of Technical Staff at Bell Labs. He is a Fellow of IEEE and coauthor of *Digital Systems Testing and Testable Design* (New York, NY, USA: IEEE Press, 1994).

Dakshi Agrawal is a Research Staff Member and Manager of the Network Management Research Group at T. J. Watson Research Center, IBM Corporation, Hawthorne, NY, USA. He is the U.S. Program Director for International Technology Alliance in Network and Information Sciences. He has a PhD in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA. He serves as an Associate Editor of the IEEE/ACM Transactions on Networking and is a Fellow of the IEEE.

Paul Bradley is a Systems Architect at Ray Group International. His research and development interests include secure computing, microelectronics design, and automated verification systems. He has a BS in computer electrical engineering from the

University of Rhode Island, Kingston, RI, USA and the MBA degree from Bryant University, Smithfield, RI, USA.

Michael S. Hsiao is a Professor in the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. His research interests include testing, design verification, and diagnosis of hardware and software. He has a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA. He is a Fellow of the IEEE.

Jim Plusquellic is an Associate Professor in Computer Engineering at the University of New Mexico, Albuquerque, NM, USA. His research interests include hardware-oriented security and trust, design for manufacturability, defect-based test, and process monitors. He has a PhD in computer science from the University of Pittsburgh, Pittsburgh, PA, USA, in 1997. He has published more than 80 papers in journals, conferences, and workshops and is a member of the IEEE.

Mohammad Tehranipoor is an Associate Professor of Electrical and Computer Engineering at the University of Connecticut, Storrs, CT, USA. His research interests include VLSI testing, reliability analysis, and hardware security and trust. He has a PhD in electrical engineering from the University of Texas at Dallas, Richardson, TX, USA. He is a Senior Member of the IEEE and a member of the Association for Computing Machinery (ACM) and the ACM Special Interest Group on Design Automation (SIGDA).

■ Direct questions and comments about this article to Swarup Bhunia, 10900 Euclid Avenue, EECS, Case Western Reserve University, Cleaveland, OH 44106, USA.