J Electron Test (2017) 33:107–124
DOI 10.1007/s10836-016-5632-y

Golden-Free Hardware Trojan Detection with High Sensitivity Under Process Noise

Tamzidul Hoque[1] · Seetharam Narasimhan[2] · Xinmu Wang[2] · Sanchita Mal-Sarkar[3] · Swarup Bhunia[1]

Received: 19 September 2016 / Accepted: 13 December 2016 / Published online: 28 December 2016

Abstract Malicious modification of integrated circuits in untrusted design house or foundry has emerged as a major security threat. Such modifications, popularly referred to as *Hardware Trojans*, are difficult to detect during manufacturing test. Sequential hardware Trojans, usually triggered by a sequence of rare events, represent a common and deadly form of Trojans that can be extremely hard to detect using logic testing approaches. Side-channel analysis has emerged as an effective approach for detection of hardware Trojans. However, existing side-channel approaches suffer from increasing process variations, which largely reduce the detection sensitivity and sets a lower limit of the sizes of Trojans detectable. In this paper, we present TeSR, a Temporal Self-Referencing approach that compares the current signature of a chip at two different time windows to isolate the Trojan effect. Since it uses a chip as a reference to itself, the method completely eliminates the effect of process noise and other design marginalities (e.g. capacitive coupling), thus providing high detection sensitivity for Trojans of varying size. Furthermore, unlike most of the existing approaches, TeSR does not require a golden reference chip instance, which may impose a major limitation. Associated test generation, test application, and signature comparison approaches aimed at maximizing Trojan detection sensitivity are also presented. Simulation results for three complex sequential designs and three representative sequential Trojan circuits demonstrate the effectiveness of the approach under large inter- and intra-die process variations. The approach is also validated with current measurement results from several Xilinx Virtex-II FPGA chips.

Keywords Hardware Trojan detection · Sequential Trojans · Side-channel analysis · Self-referencing · Trust in IC

Responsible Editor: S. Hamdioui

✉ Tamzidul Hoque
thoque@ufl.edu

Seetharam Narasimhan
sxn124@case.edu

Xinmu Wang
xxw58@case.edu

Sanchita Mal-Sarkar
s.malsarkar@csuohio.edu

Swarup Bhunia
swarup@ufl.edu

[1] University of Florida, Gainesville, FL, USA

[2] Case Western Reserve University, Cleveland, OH, USA

[3] Cleveland State University, Cleveland, OH, USA

1 Introduction

Due to global outsourcing of fabrication services to foreign countries, there is an emerging security concern with integrated circuit (IC) manufacturing, regarding potential malicious modification during fabrication in untrusted foundry [15]. Such malicious hardware modifications, also referred to as *Hardware Trojans*, can give rise to undesired functional behavior of a chip, for example, providing covert channels or *back doors* through which sensitive information such as cryptographic keys can be leaked, or simply malfunction under certain circumstances. Besides tampering the functional robustness of generic consumer electronics, hardware Trojans can cause catastrophic consequences during in-field operation of security-critical applications

such as military, communication and national infrastructure. Conventional structural and functional testing fails to reliably detect these Trojans due to their stealthy nature and inordinately large number of instances an adversary can exploit [14].

Hardware Trojan circuits can be either *combinational* or *sequential* in nature [13]. A *combinational Trojan* depends on the occurrence of rare logic values at one or more internal circuit nodes to trigger, while a *sequential Trojan* acts as a *time-bomb*, exhibiting its malicious effect after a sequence of rare events during long period of field operation. Figure 1a shows a generic model for sequential Trojan. Examples of sequential Trojan circuits are k-bit synchronous counter, as shown in Fig. 1b and Finite State Machine (FSM) which is triggered by rare events in the internal nodes of the original circuit, as shown in Fig. 1c. Trojan activation condition is referred as *Trigger condition*, while the node that can be affected when the Trojan is triggered is referred as *payload*. The individual state transition conditions are referred as *partial trigger conditions* (PTC). Another kind of sequential Trojan [27] with a *passive* payload, consists of a Linear Feedback Shift Register (LFSR) which is used to leak the secret key in cryptographic hardware by aiding side-channel attacks, as shown in Fig. 1d.

Various hardware Trojan detection techniques have been presented earlier, each of them having their own merits and limitations. In fact, most of the techniques act as complimentary detection mechanisms providing their unique coverage for particular Trojan models. For example, leakage current based Trojan detection schemes can be extremely powerful for large sequential Trojans that contribute significantly to the leakage power traces, whereas logic testing-based schemes are suited for activating and identifying the presence of small combinational Trojans which can easily be missed in the process noise. Generally speaking, side-channel based approaches are advantageous in detecting sequential Trojans compared to logic testing as they do not require functionally triggering the Trojan. Instead of functional activation, triggering only the PTCs is easier to accomplish, yet beneficial to side-channel based techniques. However, most of the previous side-channel approaches suffer from reduced sensitivity due to process variations, and rely on a golden IC instance which is usually hard to obtain. Procuring a golden chip may require destructive reverse engineering through decapsulation, delayering and imaging of the chip [8].

There exists very few side-channel based post-fabrication Trojan detection techniques that do not require a golden IC. In [28] researchers are trying replace the requirement of the golden IC by using golden parametric signature obtained by combining trusted simulation model, parameters from die or wafer kerf obtained by process control monitors (PCMs), and advanced statistical tail modeling techniques. However, the requirement of precise model of the process variation makes the technique difficult. The idea described in [49]

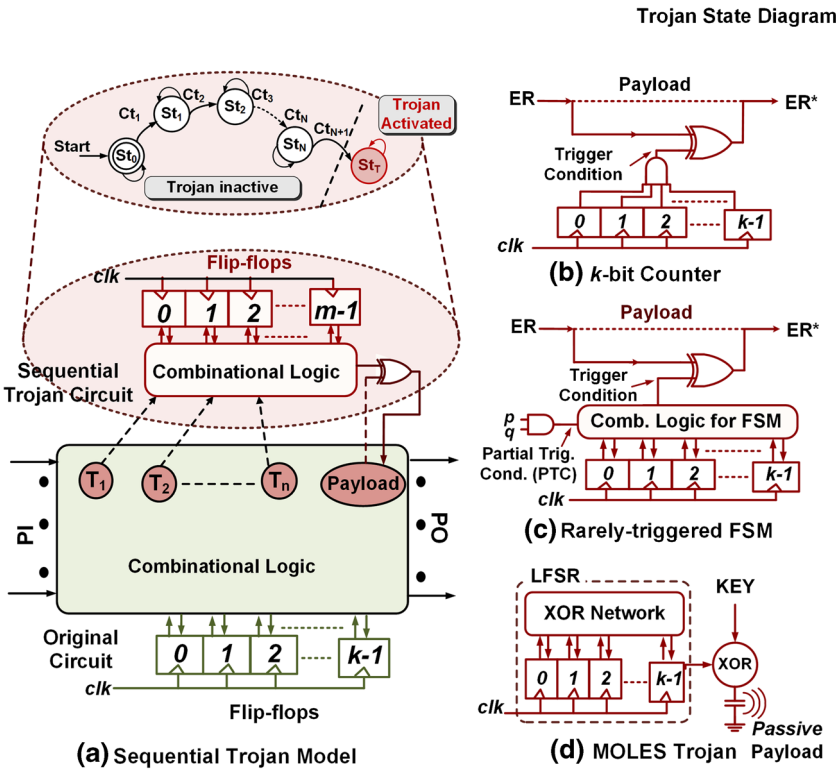

Fig. 1 **a** Sequential Trojan model and examples: **b** Synchronous Counter, **c** Rarely-triggered Finite State Machine (FSM), **d** MOLES Trojan [27]

could also be used as a golden chip free Trojan detection technique assuming only some of the ICs would have the Trojan. The technique involves recording the side-channel signatures for a given set of input vectors which would help define a Trojan infected signature as the outlier. However, if all the ICs contain the same Trojan, the mechanism would not be able to detect the Trojan. Besides, the effectiveness of the method relies on the accuracy of generated signatures which could be affected due to process and measurement noise.

In [32] we proposed a novel side-channel analysis approach referred as **Te**mporal **S**elf-**R**eferencing or **TeSR** for efficient detection of these Trojans. The proposed approach can eliminate the effects of both die-to-die and within-die process variations, as well as local noise induced by other design marginalities. It also avoids the requirement of a reference or golden IC to isolate Trojan effects by comparing a chip's transient current signature with itself - but at a different time window. Besides, unlike [49] the proposed technique could detect the Trojan even if all the ICs are infected with same Trojan. In this paper, we extended our work in [32] in the following aspects: 1) We developed an algorithm to systematically generate TeSR test sets to guarantee the effectiveness of the approach and improve Trojan detection coverage; 2) We elaborated the analysis of TeSR with respect to sequential Trojan types, and presented a Design-for-Security (DfS) technique to facilitate TeSR in testing against very hard-to-detect sequential Trojans; 3) In addition to validating TeSR on three large sequential IP cores, the effectiveness of the approach is proved by comparing its Trojan detection capability with an intra-die process calibration based approach; 4) Experimental validation was conducted on a FPGA platform with Xilinx Vertex-II XC2V500 device.

TeSR focuses on identifying the sequential Trojans, which typically represent a greater threat than their combinational counterparts, since an intelligent attacker can take advantage of few state elements to create a complex Trojan with rare trigger conditions. The main insight on which the work is based is that when a Trojan-free circuit is made to undergo the same set of state transitions multiple times, the transient current "signature" should remain constant over different time windows. However, in a Trojan-infected circuit, the overall current signature varies over multiple time windows for the same set of state transitions of the original circuit, due to uncorrelated state transitions in the Trojan. TeSR can be employed without process calibration of golden chips, as it is performed independently for each IC. To the best of our knowledge, this is the only side-channel analysis approach for Trojan detection that concurrently offers the following advantages: 1) completely mitigates the effect of die-to-die and within-die process noise (both random and systematic); 2) cancels the effect of other design marginalities; 3) avoids the need of having golden reference chip or a precise model of the process variation; and 4) detects a malicious modification even if all the ICs are infected.

The rest of the paper is organized as follows. In Section 2, we present the related work in the field of Trojan detection and the scope of the proposed approach in comparison to other complementary approaches for hardware Trojan detection. In Section 3, the temporal self-referencing concept is illustrated with a motivational example which indicates how TeSR overcomes the challenge of process variations and other design marginalities. The main methodology is described in Section 4, along with details of test generation and circuit characterization. Section 5 contains the simulation results and experimental validation of the proposed approach. We conclude in Section 6.

2 Background and Scope

2.1 Related Work

A detailed taxonomy of hardware Trojans and their detection mechanisms is presented in [42]. A common classification of Trojans [9, 13, 19] is based on the activation mechanism (referred as *Trojan trigger*) and the effect on the circuit functionality (referred as *Trojan payload*). Trojans can be both combinationally and sequentially triggered. Typically, an adversary would choose an extremely rare activation condition so that it is highly 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 of rare events, or after a period of continuous operation. The output of the Trojan circuit can maliciously affect the functionality of the circuit by affecting the logic values at its internal nodes (payload) as shown in the above examples. Another kind of Trojan which has a passive payload is used to leak the secret key used in cryptographic hardware by aiding in side-channel attacks. A classification of Trojans designed for information leakage is presented in [21].

Traditionally, two types of hardware Trojan detection techniques have been proposed in literature: (a) Logic-testing based techniques and (b) side-channel analysis-based techniques. Sequential Trojans can be extremely hard to detect using logic testing approaches [14] because the sequence of rare events required to cause all the state transitions in the Trojan, finally leading to the activation of its payload, is highly unlikely to be satisfied during test-time. Logic testing approaches depend on comparing the functional behavior of a circuit under test (CUT) with that of a golden or reference circuit. These approaches are usually more effective for detecting combinational Trojans activated

by rare values at the internal circuit nodes [42]. Furthermore, if the Trojan trigger mechanism is independent of the circuit operations (e.g. Fig. 1b), logic testing techniques become completely ineffective.

On the other hand, since side-channel analysis is based on noting the Trojan effect on physical side-channel parameters such as current or delay, they can be very effective in detecting large sequential Trojans. These approaches do not require the Trojan to be completely activated and its malicious effect propagated to primary outputs in order to be detected. However, traditional side-channel analysis approaches suffer from reduced sensitivity with ever-increasing inter-die and within-die *process variation* effects [11]. Though the Trojan circuit's activity is reflected in the supply current, the effect can be easily masked by process noise, leading to false positive/negative decisions [4]. Hence, existing approaches tend to use process calibration techniques with known set of golden ICs in order to obtain the golden trend. Any deviation from the trend (beyond a pre-defined threshold) signifies the presence of a Trojan circuit.

In [36], the authors use current measurement from multiple ports along with calibration techniques and statistical analysis to alleviate the effect of process and environmental variations. In [31], correlations between multiple side-channel parameters like transient supply current and maximum operating frequency are used to identify golden trend line which minimizes effect of process noise. Further experimental analysis to study the effect of inder-die and intra-die variations on Trojan detection sensitivity is presented in [2, 26]. In [22], a formal extension of this method to combine multiple modalities for Trojan detection is discussed. Region-based test vector generation [6, 7, 16, 39] has been shown to increase Trojan detection sensitivity for large circuits. Besides, statistical test generation for side-channel analysis has been proven to be effective to amplify the Trojan's side channel within a large design [18]. Other methods include path-delay fingerprint calibration [19, 45], ring-oscillator based delay calibration [37, 48], post-fabrication backside optical imaging [51], electromagnetic emanation (EM) measurement [41], and gate-level characterization [5, 34, 44] of leakage and delay parameters for all gates in the original design under process variations for identifying presence of extra gates post-fabrication. The novel approach of self-referencing for Trojan detection is first proposed in [16], where transient current of two similar or dissimilar regions are compared. This idea is extended to detect recycled chips and hardware Trojans in [50]. In [47], self-referencing approach has been integrated with delay based technique to compare path delays between similar paths to observe Trojan-induced deviation. The method eliminates the need of golden chips. However, original netlist for test generation is needed. Furthermore, it would be difficult to detect Trojans with negligible impact on critical path delays.

Nevertheless, existing side-channel methods cannot completely mitigate the influence of process variations in hardware Trojan detection since they depend on measurements made from multiple ICs to compare and make the decision. In addition, all of the earlier approaches rely on the availability of a set of golden ICs (usually obtained by destructive testing of a sample of untrusted ICs) or complete characterization of the golden design, which can be of exponential complexity for large designs. Complementary to these approaches are run-time functional validation approaches [3, 10], which require high design overhead, but provide a last line of defense for identifying presence of Trojans in mission-critical systems. Apart from run-time functional validation, run-time thermal and power tracking based side-channel analysis approaches are demonstrated in [17, 20, 33] which usually come with area overhead or requirement of thermal imaging equipment.

2.2 Scope of the Proposed Trojan Detection Approach

The *temporal self-referencing* approach is effective for generic sequential Trojans, which are modeled as in Fig. 1a. The state transition conditions (C_{t_i}) are derived from combinations of rare internal node values ($T_1, T_2, \ldots T_n$). The Trojan causes a malfunction at its payload in state St_T after it goes through the state transition $St_0, St_1, \ldots St_N$. We can assume without loss of generality that the Trojan FSM is often confined to the states before St_T during test-time, otherwise it would cause a malicious effect at its payload, and be detected by functional testing approaches [14]. This accounts for detection of the combinational Trojans and small sequential Trojans (very few states). We can also detect large, distributed, sequential Trojans, which are very likely to cause sufficient change in the side-channel parameter like leakage current beyond the process noise [4, 31, 36]. The various challenges for non-invasive side-channel approaches are presented in Fig. 2 along with their relative scope. The attack model exploited in this paper assumes trusted RTL design but considers Trojan insertion as possible in any stages of IC development after RTL design and verification, given the various verification techniques and phases in the front-end IC design, which can prevent Trojan insertion by insiders to a large extent.

It is generally accepted that there is no silver-bullet solution which can detect Trojans of all possible sizes and types. While the proposed TeSR scheme is suitable for sequential Trojan attacks of various forms and sizes, it provides distinct advantage for small sequential Trojans, which easily evade logic testing and existing side-channel approaches due

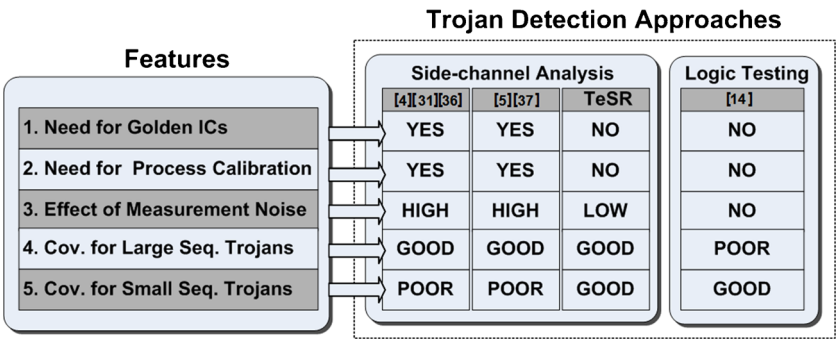

Fig. 2 Comparison of challenges and scope of different Trojan detection approaches

to process variations. Statistical logic testing approaches [14] can be used as a complementary technique to TeSR since they have high coverage for ultra-small combinational Trojans which do not produce significant effect on a side-channel parameter but can get triggered easily. The only known limitation for the TeSR approach is due to temporal variations induced by measurement noise, and solutions to reduce their effect have also been discussed here.

3 Motivational Examples

As a motivational example of self-referencing based sequential Trojan detection, we simulated a 32-bit DLX processor circuit (with ∼20,000 logic gates) in HSPICE using 70nm Predictive Technology Model (PTM) [35]. Test vector sets are designed to fill the pipeline with repeated "NOP" or "ADD" instructions, causing controlled activity in one pipeline stage at-a-time. Multiple instances of the processor were considered - non-infected and infected, at different process corners, to demonstrate the existence of time-invariant (but process-dependent) signature in each non-infected IC. The Trojan circuit is a free-running synchronous 8-bit binary counter (see Fig. 1b), which causes malfunction when it reaches the maximum count value (i.e. after 64 cycles of continuous operation), which is considered, for illustrative purposes, to be beyond the test application time. The measured side-channel parameter is the average transient supply current in each clock cycle. Due to process variations, device parameters shift from their nominal values. Figure 3a shows the effect of process variations on the transistor *threshold voltage* (V_T) and they can vary due to inter-die variations as well as intra-die variations, which can have both random and systematic components [11]. The effect of process variations was simulated using *Monte Carlo* simulations in *HSPICE* with ±20 % variations in inter-die V_T and intra-die variations having a standard deviation (σ) of 10 %. These variations can mask the effect of an inserted Trojan circuit, as evident from the overlap in the simulated average current distribution of the golden and Trojan-containing circuits in Fig. 3b. The overlap is prominent for large circuits with small Trojans, which makes it difficult to choose a single threshold value to distinguish between infected and non-infected ICs, causing large mis-classification errors.

For side-channel analysis based Trojan detection techniques to be dependable, the effect of process variations and design marginalities must be eliminated. We note that if we compare the current signature for the same IC, when it is subjected to the same test stimulus under the same experimental setup but in different time windows, we can isolate the temporal variations in Trojan current (if present). This is because the recorded current trace over different time windows consist of two components - (a) a correlated component because of identical state transitions of the circuit, and (b) an uncorrelated component due to the switchings in the Trojan circuit.

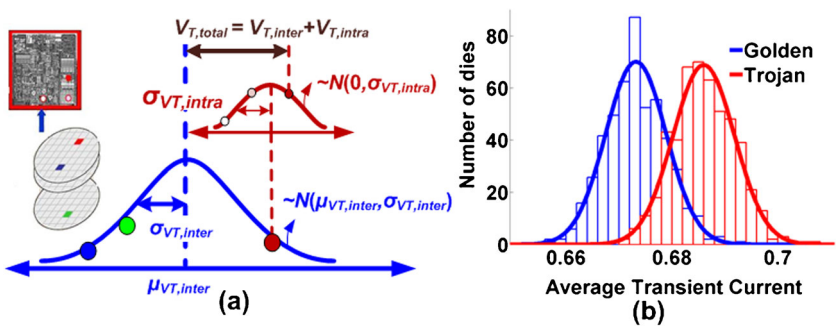

Fig. 3 **a** Circuit-level parameter variations can be due to inter-die or intra-die variations in device parameters. **b** The effect of process variations on the average transient current can mask the effect of a Trojan circuit

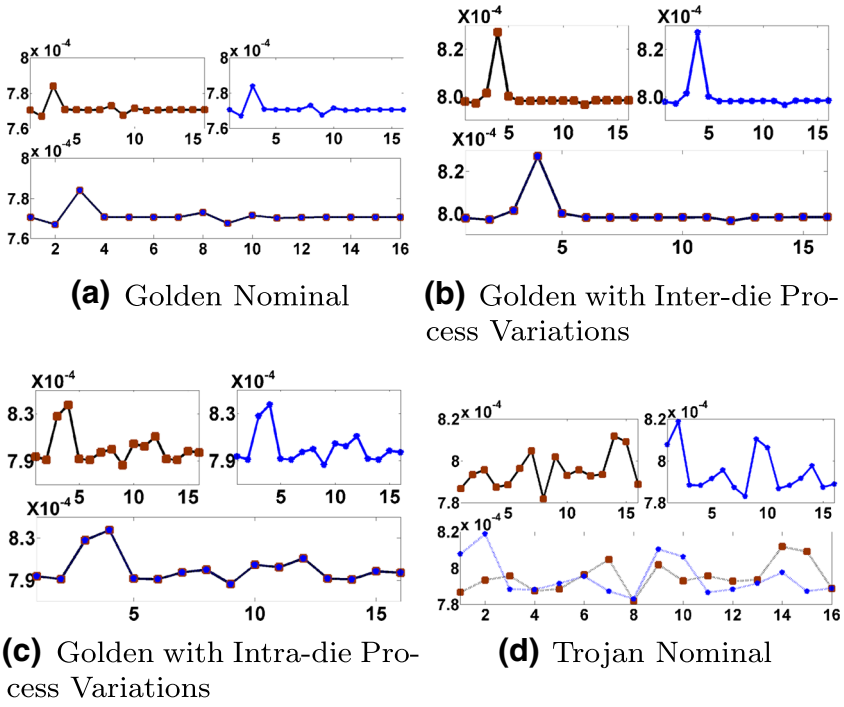

Fig. 4 Effectiveness of temporal self-referencing in detecting Trojans even amidst process variations

Figure 4a shows the cycle-by-cycle average transient current trace of the DLX circuit for two windows, where it was repeatedly brought to the same state and made to go through the same set of state transitions. The current trace corresponding to the state transitions can be clearly distinguished by its repetitive nature, and forms a "current signature" for this state transition sequence, as seen in the bottom plot of Fig. 4a, where the current signatures from the two windows are superimposed on each other. In Fig. 4b, the current signatures for the same two windows are plotted for a non-infected die at a different inter-die process corner. Process variations cause considerable change in the golden signature from chip-to-chip, but the signature for the same IC instance remains time-invariant. The same effect holds true even under intra-die process variations, as seen in Fig. 4c. Now, let us consider the current signatures for a Trojan-infected DLX circuit in Fig. 4d. Since the Trojan state machine undergoes a set of state transitions uncorrelated to the original circuit, the current trace in the two time windows differ substantially. This example motivates the use of temporal self-referencing as a high-sensitivity Trojan detection scheme.

4 TeSR Methodology

Figure 5 illustrates the basic concept of the proposed Temporal Self-Referencing (*TeSR*) hardware Trojan detection approach. First, the FSM in the original circuit is excited to go through a sequence of states S_{D1}, S_{D2} ... S_{Dn} for the purpose of Trojan detection, which is called *Test trial* #1. These state transitions are triggered by specifically derived test patterns ($\{V_{test}\}$) that can maximize Trojan circuitry activity. Therefore, certain Trojan FSM transition can be expected to take place. In this example, a single transition of Trojan FSM is assumed for the sake of simplicity. When the original FSM reaches state S_{Dn}, another set of test patterns is applied to bring the FSM back to state S_{D1}, during which the Trojan FSM can have zero or more transitions.

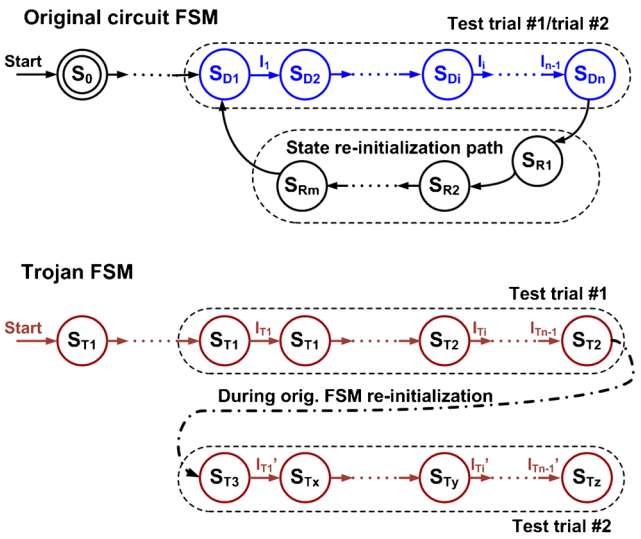

Fig. 5 Basic concept of TeSR

Without loss of generality, we use S_{T3} to denote the Trojan FSM state after re-initialization, where S_{T3} is not necessarily different from S_{T2} but expected to be different from S_{T1}. At this point, $\{V_{test}\}$ is applied again to excite the original FSM to traverse the same sequence of states S_{D1}, S_{D2} until S_{Dn}, which is denoted as *Test trial* #2. During this process, the Trojan FSM can have certain state transitions starting from S_{T3}, which would be different from its transitions in Test trial #1 given the fact that S_{T3} is not the same as S_{T1}. Transient current I_{DDT} is characterized for each clock cycle during Test trial #1 and Test trial #2. Comparison is then performed between the two trials. The original circuit will exhibit exactly the same switching activities in the two trials, given the same initial state and the same set of test patterns. However, since the Trojan FSM starts from two different states in both test trials, the state transitions as well as combinational logic switching will be different, leading to different I_{DDT}. Therefore, if there is any difference between the measured I_{DDT} of the circuit under test during the two test trials, one can infer that it is caused by a sequential Trojan. The underlying assumption is that Trojan FSMs will have a state transition diagram uncorrelated with that of the original circuit FSM; otherwise the Trojan can be detected easily with logic testing. The major steps of *TeSR* methodology are shown in Fig. 6. It involves both *test generation* and current measurement-based *circuit characterization*. Starting with the set of generated test vectors, input vectors are applied to take the circuit to each of the starting states S_{init} of the pre-determined test trials. Once the circuit is in a desired starting state S_{init}, a corresponding set of test vectors $\{V_{test}\}$ are applied which take the circuit through a fixed set of state transitions in order to produce the characteristic current trace. The current signature is computed by taking the average of the transient current waveform for each cycle. The difference metric for comparing the current signature of two windows is taken as the Euclidean distance between the two current signatures. If one or more of the current traces differ from the average current trace over multiple windows by a pre-defined threshold to account for the temporal measurement noise, the IC is inferred to contain a Trojan. This technique is repeated for various test sets starting from different states S_{init} in order to ensure detection coverage of different kinds of possible Trojan instances.

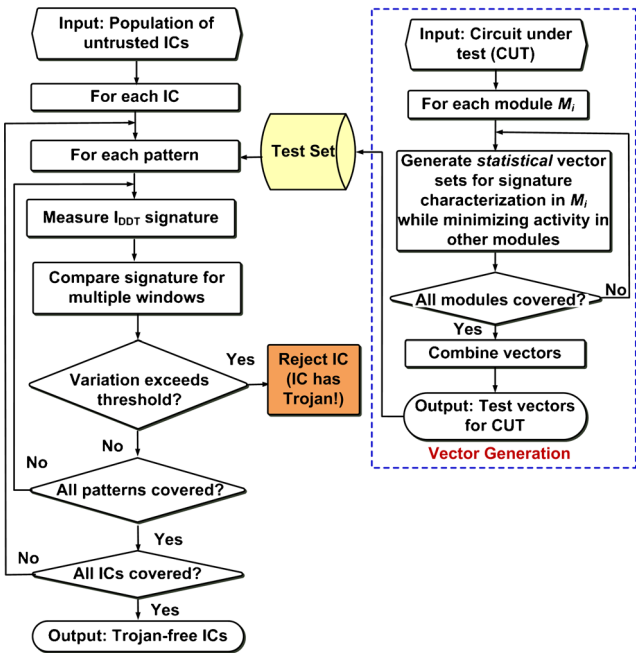

Fig. 6 The major steps of the TeSR for Trojan detection

The recorded current traces might have small, random temporal variations due to transient measurement noise, supply voltage fluctuations or temperature variations. The effect of these noise components can be minimized by maintaining stable experimental conditions during testing. Also, random measurement noise can be largely eliminated by averaging the transient current waveform over multiple measurement runs which have to start from power-on reset in order to ensure that the Trojan circuit is also re-initialized. The pre-defined decision threshold also determines to some extent the limit of the detection efficiency of the proposed approach. For ultra small Trojans, the variation in current signature might fall beneath the noise floor. However, the detection sensitivity can be increased by using previously proposed approaches, like region-based test generation [6] and use of measurements from multiple supply pins [36] in order to decrease the background current and increase Trojan detection sensitivity.

4.1 Test Generation

The test generation procedure is divided into two parts. First, a statistical test pattern generation approach MERO [14] is applied to generate sets of test patterns (corresponding to the test trials) that can maximize the Trojan circuitry activities. MERO basically identifies low probability conditions at the internal nodes of a circuit and models candidate Trojans that could be triggered by a subset of these rare conditions. As an output, MERO generates a set of test vectors that can excite these rare nodes individually to their rare condition, multiple times, which guarantees switching activity inside a Trojan circuit while achieving around 85 % reduction in test length. After MERO is applied, reachability analysis is performed on the FSM of the original circuit to identify transition paths which bring the FSM back to the first state of each test trial.

Algorithm 1 TeSR test pattern generation

Step 1: Perform FSM state traversal/reachability analysis & form a database of $p = \{S_s, S_e, v\}$
Step 2: Group p according to S_e
Step 3: Sort the groups in descending order of size & select the first N S_e as the initial states of MERO vectors (S_{init})
Step 4: $\forall S_0 \in \{S_{init}\}$ Apply MERO vector set V_M from S_0, which ends at state S_F
$P_{BB} \leftarrow$ # of paths $p = \{S_F, S_0, v_i\}$ in the database
if $P_{BB} = 0$ **then**
  abandon this V_M
else if $P_{BB} = 1$ **then**
  TeSR set=$\{V_M, v, V_M\}$
else
  TeSR set=$\{V_M, v_1, V_M, v_2, ...V_M, v_{P_{BB}}, V_M\}$
end if

The detailed steps of the test generation procedure is provided in Algorithm 1. First, reachability analysis is performed on the original FSM to identify states which can seldom be reached (in terms of number of paths) by other states (step 1, 2, 3). These states are not suitable to be used as the first state of a test trial. In particular, step 1 performs a reachability analysis on the original circuit FSM to identify all paths between every state pair. Here v represents the complete information of the path starting from state S_s and ending at state S_e, which contains all intermediate states on the path and input vectors to trigger each intermediate state transition. In step 2, the identified paths are grouped according to the destination states S_e. These destination states are to be used as the first state for MERO test trials. The groups are then sorted in descending size (step 3), where a larger size indicates S_e that can be reached through more paths. Only the first several destination states are selected as the candidates to apply MERO sets from, in order to guarantee a good chance of re-initializing the FSM. These states form the MERO intial state set $\{S_{init}\}$. The aim would be to obtain a set of starting state that guarantees the chances of reinitialization from many different states. This facilitates testing of original FSM using different paths. Furthermore, multiple different initial states also diversifies the possible paths that could be explored. A Trojan could mimic the original FSM for certain paths, but to be able to follow the original FSM for all possible paths within different initial-reinitializing state pairs, the Trojan would have to be extremely large.

Next, in step 4, MERO vectors are generated and applied from each element S_0 in the optimal initial state set $\{S_{init}\}$, and the corresponding end states S_F are recorded. Note that in the test generation procedure, attention is only paid to the logic activity of the circuit; while the transient current signature is only captured in the signature characterization phase. Next a search is performed within the reachability database for elements $p = \{S_F, S_0, v_i\}$, where v_i represents a generic path. If there is no such a path, it means the current initial state S_0 is not reachable by the end state of the current set of MERO vectors. Then the current MERO set is abandoned for this S_0. However, this does not mean the MERO set cannot be applied for other elements in $\{S_{init}\}$. If a single path $p = \{S_F, S_0, v_i\}$ exists in the reachability database, the end state of MERO vectors can be brought back to S_0 through p. And the entire TeSR test set can be expressed as $\{V_M, v, V_M\}$, where v stands for the test vector sequence to realize transition path p.

In fact, the procedure of re-initializing and re-applying MERO sets can be repeated multiple times to increase the chance of capturing Trojan circuit activities during the repetitive MERO tests. As stated before, the basic requirement to guarantee the effectiveness of TeSR is different initial states of Trojan FSM when repetitively applying MERO test sets. Upon satisfaction of this requirement, the circuit IDDT signatures will vary among multiple test trials, and the difference can contain combinational and sequential switching components. In particular, if Trojan FSM state transitions only occur during the re-initialization procedure and not in the test trials, the captured IDDT discrepancy would be only due to different switching of the combinational next-state logic. And the amount and distribution of the discrepancy depend on to what extent and in what frequency the MERO sets can trigger the Trojan activity. On the other hand, if the Trojan FSM have state transitions within the test trials, the IDDT difference will be partially contributed by the sequential switching and can be much more significant. Repeating the MERO sets along with the re-initialization test vectors multiple times can improve the probability of Trojan FSM state transition during both procedures by statistically increasing Trojan circuitry activities. This will lead to a more remarkable IDDT difference, hence improve the chance of Trojan detection.

The algorithm for performing reachability analysis based on breadth-first traversal is explained in Appendix A. While reachability analysis is widely used in formal methods, it may face state space explosion for very large designs. Nevertheless, highly efficient approaches have been developed to reverse engineer a state machine from the gate level netlist [29, 40]. The reachability information could be easily obtained from the extracted state transition graph. Besides, [40] proposes an automatic test pattern generation (ATPG) based FSM extraction technique to generate the state transition graph from the gate level netlist. Even if none of the above mentioned methods work, the designer can use the paths from the partially extracted state space information, since all possible paths are not required for executing the side channel experiment. If appropriate reinitializing paths are not found within the partially extracted state space, the designer can even integrate dummy transition paths as an alternative.

4.2 Circuit Characterization

From the generated test vector set, a sequence of test vectors are applied which takes the circuit to the state S_{init}, followed by the set $\{V_{test}\}$ that makes the circuit go through a fixed set of state transitions in order to produce the characteristic current trace. The current signature is computed by taking the average of the transient current waveform for each cycle.

The *difference metric* for comparing the current signature of two windows is taken as the point-wise Euclidean distance between the two current signatures. If one or more of the current traces differ from the average current trace over multiple windows by a pre-defined *noise threshold*, the IC is inferred to contain a Trojan. This pre-defined noise threshold value can be obtained by taking multiple current measurements with constant activity (reset state) to characterize the background noise in the measurement setup. However, unlike other side-channel Trojan detection approaches, we do not require one or more golden ICs to determine the threshold.

4.3 Trojan Detection Sensitivity

The sensitivity of a simple side-channel approach based on comparison of measured physical parameter I can be defined in terms of various noise effects and different calibration techniques. For example, in a simple side-channel approach, considering ideal situation with no noise, any golden circuit is expected to have the measured parameter value as I_{orig}. The deviation introduced by an extra Trojan circuit causes the measured value for an infected chip to be $I_T = I_{orig} + \Delta I_T$. The sensitivity, in the absence of noise, is proportional to ΔI_T and inversely proportional to I_{orig}. Now, with the presence of measurement noise $I_{n_{meas}}$ and process noise, $I_{n_{proc}}$, the measured values of the non-infected circuits can vary from I_{orig} by $I_{n_{meas}} + I_{n_{proc}}$. The process noise $I_{n_{proc}}$ is a time-invariant constant which affects different ICs differently. It can further be decomposed to contain inter- and intra-die components, with the intra-die component having systematic and random sub-components. The measurement noise $I_{n_{meas}}$ has a temporal variation (due to temperature and other factors) for the same IC and a dc offset due to measurement circuitry. Considering the simple side-channel analysis approach, sensitivity can be defined as:

$$Sens = \frac{I_T - I_{orig}}{(I_{orig} + I_{n_1}) - (I_{orig} + I_{n_2})} = \frac{\Delta I_T}{\Delta I_{n_{meas}} + \Delta I_{n_{proc}}}.$$

Existing side-channel approaches tend to perform process calibration by using normalization (or process-corner estimation) and measurement noise calibration by averaging over multiple measurements to get rid of random noise. In order to get rid of inter-die variations and calibrate systematic intra-die variations, region-based approaches are used where measurements from multiple power pins corresponding to activation of distinct regions, help to compare the measured parameter from the same IC under different circumstances (self-referencing). By using a region-based approach, one can also increase the sensitivity since the I_{orig} value gets reduced and any noise which is proportional to the measured value (e.g. process noise) gets reduced as well. However, by using a temporal self-referencing approach like TeSR, we can completely eliminate process noise since we are comparing measurements for the same input vectors for the same IC under different time windows. Hence, $Sens_{TeSR} = \frac{\Delta I_T}{\Delta I_{n_{meas}}}$. The ΔI_T in this case is the difference in activity within the Trojan circuit at different time windows, since the original circuit will have the same value of the measured parameter for the same set of vectors. The only factor which limits the sensitivity of this approach is the time-varying component of measurement noise. This can be reduced by performing the measurements under temperature-controlled test environment with high-quality test equipment, as done in standard semiconductor testing facilities in the industry. Moreover, by averaging measurements over multiple cycles, we can further increase the sensitivity.

4.4 Role of Scan Chain

In order to improve testability of sequential circuits, test engineers typically use various "Design for Test" (DfT) measures such as *scan-chain insertion*. If the sequential elements (flip-flops) in the design are implemented as *scan flip-flops* and connected in a chain, any value can be loaded into them in the testing phase, thereby reducing the test generation problem to that for a combinational circuit which is much tractable computationally. The degree of testability and the associated design overhead provide a trade-off which causes circuit designers to go for *partial scan-based* approaches where only a few selected flip-flops are part of the scan-chain. If the design is equipped with full-scan, it is easy to initialize the entire circuit at any particular state from which the current signature is to be measured. For the state diagram shown in Fig. 7a it is possible to take the circuit to the desired state S_{10} to start the test application procedure for Trojan detection. However, it must be noted that the easily identifiable standard *test control* (TC) signal can be used by the attacker to disable the Trojan or to synchronize the Trojan state machine with the test application phase. This would defeat the purpose of temporal self-referencing as, in this case, the Trojan current signature would be invariant for each application of the same test sequence. In order to avoid this, we need to perform side-channel current signature

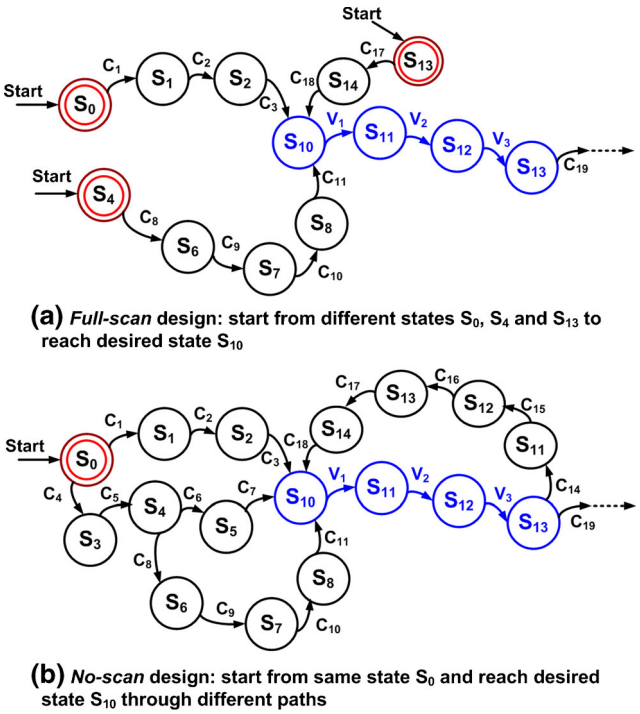

Fig. 7 Test application strategy considering the state transition diagrams for **a** full-scan and **b** no-scan designs. The example test signature consists of the average current for vectors I_1, I_2 and I_3 applied when the circuit is in state S_{10}. Different paths are used to arrive at state S_{10} to get same current signature for the golden circuit but different signatures if a Trojan is present and shows some activity for the particular test set under consideration

measurement in the normal functional mode of the circuit and not in any easily-identifiable test mode.

However, full-scan designs can still be used to aid in the test application process for temporal self-referencing. As shown in Fig. 7a, one of the desired test sequences to be applied is $\{V_1, V_2, V_3\}$ starting from the initial state S_{10}. For each window, we need to compare the current signature obtained during application of this test pattern for Trojan detection using temporal self-referencing. For this, we need to ensure that the Trojan is not at the same state each time we take the circuit to the state S_{10}. We note that by using full-scan capability of the design, we can initialize the circuit to different initial states S_0, S_4 or S_{13}, which are close to the desired state S_{10}, and then use input vector sequences obtained from *reachability analysis* to direct the circuit to the desired state along different paths. In other words, scan chain allows one to set the original FSM to an easily re-initializable (to S_{10}) state. This can reduce the test time and increase diversity in the re-initialization paths, thus improving the chance of causing Trojan switching activity.

There is another constraint on the lengths of the scan-facilitated initialization paths (e.g. from S_0, S_4, or S_{13} to S_{10}). Suppose the Trojan is of the free-running synchronous counter type, and it gets reset with the TC signal (in the full-scan case) or the reset signal (in the no-scan case). In this case, if the lengths of all scan-facilitated initialization paths are equal, the Trojan would be at the same state irrespective of the actual path taken to arrive at S_{init}. In that case, the effect of the Trojan on the overall current signature would be identical for every run, and temporal self-referencing would be unable to detect the Trojan. Hence, the length of the scan-facilitated initialization paths should all be different, which would ensure that the Trojan state machine is at a different state for each of the runs. Ideally, the lengths of the paths should be mutually prime, which would eliminate the possibility of Trojan FSM state coincidence at S_{init} in different test trials. Hence, the overall set of patterns required to record a characterization dataset is given by $V = \{(S_{scan}, \{C_p\}, \{V_{test}\})\}$, where the set of vectors $\{C_p\}$ takes the circuit from state S_{scan} (set via scan chain) to S_{init}, and $|C_p|$ is mutually prime for different paths p for the same S_{init}.

4.5 DfS for Detecting Transition-Proof Trojans

The effectiveness of TeSR demands the Trojan FSM to start at a different state in each test trial. To achieve this, MERO test generation algorithm is applied to maximize switching activity of Trojan circuitry thus the frequency of Trojan FSM state transition. However, it is particularly difficult to make certain types of sequential hardware Trojans to have captureable transitions as they tend to get stuck in certain state(s) stably. STG of one such Trojan is provided in Fig. 8. It can be seen that each state transition towards the final state requires an input vector from a pre-defined set; upon any other input vector, the FSM will go back to the initial state. Since the difficulty of satisfying a rare event sequence grows exponentially with the length of the sequence, FSM of this type of Trojans stays in the initial state for most of the time. Examples of such Trojan are the ones monitoring a particular input or internal variable sequence, which triggers the payload effect only when the expected input sequence is satisfied in consecutive clock cycles, otherwise returns to the initial state. We name this type of sequential Trojans as *Transition-Proof Trojans* (TP Trojans). It is difficult to capture TP Trojans in states other than the initial states. Therefore, in this case, TeSR will lose the power because the Trojan FSM, besides the original circuit FSM, also starts from the same (initial) state in each test trial.

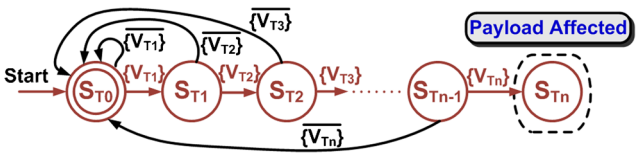

Fig. 8 STG of transition-proof Trojan

To solve this problem, we propose a *Design-for-Security* (DfS) technique which can freeze the original circuit FSM in any state (provided full-scan-chain) and test the circuit I_{DDT} signature on a per-cycle basis. The DfS-enhanced design is illustrated in Fig. 9. The original state elements of the circuit take inputs from the next-state logic and produce output to the next-state logic. With the DfS-enhanced feature, the flip-flops work as usual in normal mode, but can retain their values in the Trojan detection mode (en=1) while the next-state logic still switches due to the test vectors applied at primary inputs. Therefore, in the Trojan detection mode, original circuit switching activity depends purely on the primary inputs, and consists of only combinational switching. Any switching current uncorrelated with the input vectors must be caused by the Trojan state elements. This means, if we apply the same input vectors multiple times and observe different I_{DDT}, we can claim the existence of sequential Trojans. It is worth noticing that the attack model assumes trusted RTL, which means Trojan insertion can only happen in back-end design (in IC layout) or foundries. Our DfS technique will be implemented in RTL as explained in more details later, a Trojan FSM will not have the DfS feature to freeze its states. Therefore, it can still cause different IDDT signatures during multiple test trials due to its sequential property. In particular, we expand each MERO test set to an enhanced set V_{MEROe} as follows:

$$V_{MERO} = \{v_1, v_2, ..., v_m\} \Rightarrow \quad (1)$$

$$V_{MEROe} = \{v_1, v_1, v_1, v_2, v_2, v_2, ..., v_m, v_m, v_m\} \quad (2)$$

By tripling each test vector in V_{MERO} we could zoom in our observation by comparing the circuit I_{DDT} cycle by cycle. In the three cycles when applying v_i, I_{DDT} of the second and third cycles are measured. Since in the first cycle, v_i is applied and next-state logic outputs computation is completed, when applying v_i again in the second and third cycle, there should not be any switching activity measured. However, if input vector v_i triggers a Trojan state transition, we should be able to observe non-zero (and different) switching current either due to Trojan state transition again (e.g. return to the initial state) or different Trojan combinational logic switching (because of different FF values serving as Trojan next-state logic inputs). Such "zoom-in" test allows us to identify TP Trojans. Considering that Trojan state transitions may be only triggered under certain but not all original FSM states, the above test need be performed for various original FSM states. The initialization can be realized by shifting in the desired state values through full-scan-chain, or by running the circuit under normal mode for certain deterministic time and assert the Trojan detection enable signal.

One drawback of directly inserting multiplexers (MUXes) in the design netlist is that the array of MUXes renders the enable signal (*En*) easily visible to attackers who explore the design layout or netlist. To avoid such exposure, two tricks are employed when implementing the DfS technique. First, the FSM state freezing is realized during RTL design phase by modifying the STG to include an arch from each state to itself. For states that do not have a path to itself, such path is added with the transition condition of "$En = 1$". For states that already have a path to itself, the transition condition is altered to "$Cond \| En = 1$", where "$Cond = 1$" is the original transition condition. Second, a separate primary input En may be identifiable especially when appearing repetitively near the flip-flops. Therefore, we implement a small FSM to activate En with a particular sequence of input vectors:

$$seq(V_{en1}, V_{en2}, ..., V_{enN}) \rightarrow En = 1 \quad (3)$$

where N is the number of input patterns in the sequence. These input vectors are beyond the functional vector set to make sure En will not get asserted unintentionally when the chip is normally functioning. En gets deactivated with the global reset signal. The entire design modification is done in RTL hence the enable signal and the MUX logic will be merged into the next-state logic then mapped to the designated technology libraries. No DfS signals are left outstanding and the DfS feature is difficult to detect without a thorough design reverse engineering.

4.6 Trojan Correlated to the Original FSM

Since the proposed Trojan detection methodology relies on the assumption that the state transition diagrams of the Trojan FSM and the FSM of the original circuit are uncorrelated, it is important to discuss if a Trojan could be designed that mimics the original FSM state transitions to a sufficient

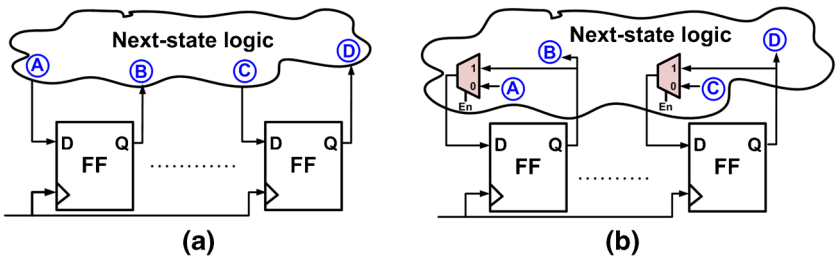

Fig. 9 **a** Flip-flops in original circuit FSM; **b** DfS-enhanced flip-flops.

extent in order to evade detection. We define the aforesaid concept of Trojan as "Correlated Trojan". If the design of the original FSM is known, a correlated Trojan is easier to model considering it is mostly a replica of the original FSM with some additional state transitions that would trigger a malicious activity. Since it ideally contains all of the original states and corresponding transition paths, the Trojan is most likely to be able to return to its original state irrespective of which state is selected as the initial and re-initializing state for TeSR. Therefore, TeSR methodology would be unable to detect such Trojans alone. However, such model of Trojan is extremely unlikely to be inserted for the following reasons:

1) To be able to design a Trojan that can circumvent TeSR, an almost equivalent version of the original FSM is required. Such obligation is completely conflicting with Trojan design concepts that are followed to evade most of the traditional Trojan detection methodologies which necessitate the Trojan induced overhead and corresponding side channel to be extremely limited [9, 21]. Since its unlikely that an outsider knows what Trojan detection methodology would be in place, it would be irrational for the attacker to insert an undisguised Trojan solely to evade TeSR.

2) In order to implement such Trojans the attacker at the foundry would have to reverse engineer the particular target FSM from the original design that generally contains a number of functional blocks and corresponding FSMs. It has been shown that in the presence of obfuscation or locking mechanism that prevents IP theft, reverse engineering would be extremely difficult [12, 38]. If the reversed design is not absolutely coherent with the original FSM, the Trojan would be prone to easy detection since, a missing state or an uncorrelated state transition could cause the correlated Trojan to traverse different paths in different test trials.

3) An attacker could attempt to design a low overhead correlated Trojan by excluding certain states or specific state transitions of the original FSM. However, unless those states/transitions are not replaced by some other states/transitions, the inserted Trojan would not be able to sync with the original one in different test trials. The mathematically proof supporting the above statement is described in Appendix B.

4.7 Summary of Test Considerations

In this work, we target the detection of Trojan instances which affect the transient current signature of a circuit. In particular, the effectiveness of our approach is due to the following features:

- *Nature of the inserted Trojan*: An inserted Trojan instance is sequential in nature, either running independently (e.g. the binary counter of Fig. 1b), or triggered by rare events at the internal nodes of the circuit (e.g. the asynchronous counter of Fig. 1c). An DfS technique can be adopted to facilitate TeSR in testing against *Transition-Proof Trojans*. The proposed approach needs to be augmented with functional testing approach which is effective for detecting ultra-small combinational Trojans which do not have time-varying signature.
- *Test application*: The circuit can be brought to the same state multiple times by state transitions along different paths. Starting from this state, the circuit can then be made to traverse a pre-determined set of state transitions to produce current signatures that can be used for comparison.
- *Variation of Trojan current signature*: The effect of an inserted Trojan on the current signature varies with the change in state of the Trojan circuit. Earlier circuit-level design techniques have been proposed to equalize the switching currents for all state transitions in CMOS circuits in the context of securing cryptographic circuits against power-analysis attacks [43]. However, such circuits are known to cause over 2X increase in area/power which can make them easily detectable by simple current analysis. Besides, current balancing techniques suffer from reduced effectiveness under process variations. Process variation tolerant current balancing circuits require asynchronous design techniques, which increases the overhead further [23].
- *Elimination of process variation effects and design marginalities*: The effects of both *intra-die* and *inter-die* process variations as well as effects of design marginalities on the transient current signature depends solely on the IC instance under test and the set of state transitions for which the dataset is recorded. Hence, self-referencing eliminates all effects of process induced variations and design marginalities on the current signature by comparing between multiple datasets for the same set of state transitions of the same IC instance.

5 Results

In this section we present the effectiveness of the temporal self-referencing based validation of sequential Trojans. We present both simulation as well as measurement results obtained from FPGA experiments.

5.1 Test Setup

We used three test circuits to validate the proposed Trojan detection approach: 1) an AES cipher circuit with an equivalent area of 22,386 two-input NAND gates (i.e $\sim 10^5$ transistors) and about 30 % of the total area contributed by memory elements, 2) a 32-bit pipelined Integer Execution Unit (IEU) with 20,775 two-input gates and 3) a 32-bit DLX processor with a 5-stage pipeline with about 19,338 two-input gates (mentioned in Section 3). We introduced three types of sequential Trojan circuits in each of the designs to investigate the scalability of the approach. The first Trojan is a k-bit synchronous counter as shown in Fig. 1b, where k was varied from 1 to 10 bits. The second Trojan is a synchronous Finite state Machine(see Fig. 1c) with 6 flip-flops. The partial trigger condition is a 9-bit value derived from the rare-valued internal nodes of the original circuit. The third Trojan, as shown in Fig. 1d is a Linear Feedback Shift Register (LFSR) with 20 flip-flops, modeled on the MOLES Trojan described in [27], which leaks the secret key inside the AES circuit by modulating a pseudo-random number generator to assist in side-channel attacks.

All circuits were designed (or obtained from [1]) in Verilog and synthesized using *Synopsys Design Compiler* and a *LEDA* library. Circuit simulations were carried out for the 70nm *Predictive Technology Model* (PTM) [35] using the *HSPICE* simulator. We used *Monte Carlo* simulations to model the effect of inter- and intra-die variations in V_T. The measurement noise from recorded current waveforms (as explained later in Section 5.3) was characterized to generate random Gaussian noise in MATLAB, which was used in our simulations to model the effect of temporal variations. The test vectors were generated based on the algorithm described in Section 4.

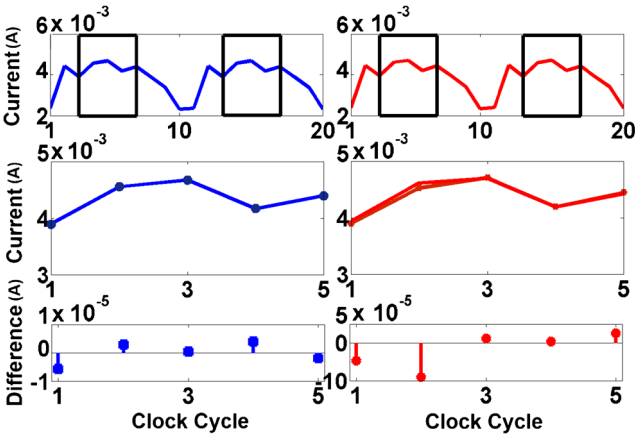

Fig. 10 IEU with a 8-bit counter. The two trials of the design without Trojan is shown in blue (*on left*) while the one with Trojan is shown in red (*on right*). For one of the points, Euclidean distance shows significantly higher difference in current within two trials that indicates a presence of a Trojan

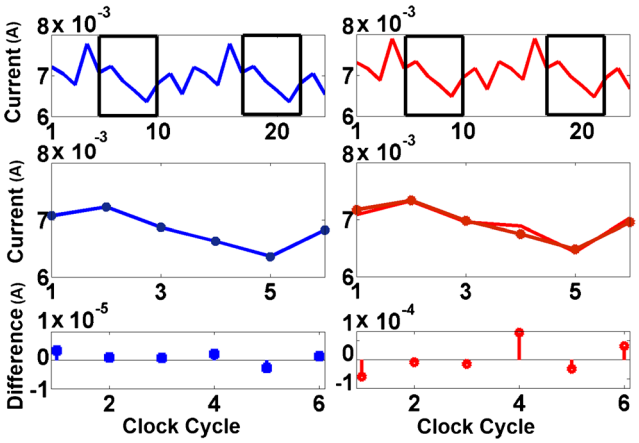

Fig. 11 AES with a MOLES Trojan LFSR (Linear Feedback Shift Register). Please note the the distance shown for with and without Trojans are shown in different scales (e.g. 10^{-5} and 10^{-4}) which indicates a larger difference

5.2 Simulation Results

Figure 10 shows the plot of average current over each clock cycle for a 32-bit Integer Execution Unit (IEU) as the original design with and without an 8-bit counter as the Trojan. The average current trace (blue for non-infected and red for Trojan) shows repetition between the two windows corresponding to the signature, which are highlighted using the black rectangles. The current signatures for the two windows are superimposed and the difference between the two signatures are also plotted. It can be clearly observed that

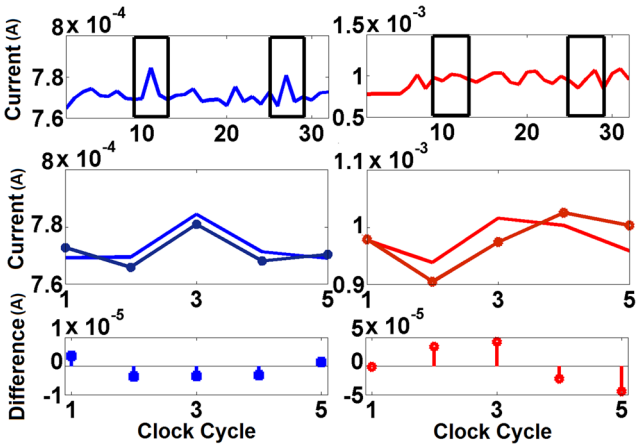

Fig. 12 DLX with a FSM Trojan. Anomaly caused by the Trojan is very high within two different trials

Table 1 Difference metric and Test Length for three designs with three types of Trojan instances

	Test	Difference Metric (μA)			
	Length	No Trojan	Counter	FSM	LFSR
IEU	752	2.68	47.26	214.30	89.88
AES	1161	3.11	87.09	215.30	78.28
DLX	605	2.96	4.10	33.90	33.63

there is a significant difference in the signatures for the two windows due to presence of Trojan. Similar waveforms are plotted for different non-infected and Trojan-infected circuit combinations in Figs. 11 and 12 respectively. Note that, unlike existing process calibration approaches, we do not need to compare the current signature between non-infected and infected ICs to detect presence of Trojan. The slight difference in current signatures for the original circuit is due to measurement noise, which is superimposed on the supply current waveforms. The noise threshold is obtained from the measurement data and any difference larger than that is attributed to the presence of a Trojan. The difference metric values for the different circuits with different Trojan instances are shown in Table 1. The difference for a non-infected IC is also shown for comparison, which falls within the noise threshold.

Table 1 also lists the test length obtained using our test vector generation tool, which causes each rare node to go to its rare value $N = 20$ times in order to activate arbitrary combinations of rare nodes, as possible Trojan state transition conditions. Note that, for ICs containing Counter- or LFSR-type Trojans, the entire test set need not be applied for detecting their presence. Next, we insert Trojan counters of different sizes in the IEU circuit to estimate the sensitivity of the TeSR approach and compare with existing approaches which perform process calibration. We varied the Trojan size from 20 bits to 2 bits and the corresponding values of the difference metric are plotted in Fig. 13. The process calibration technique is modeled using normalization of measured current to estimate the process corner and reduce it to the nominal value. The uncalibrated process noise is 1.6 mA, which is reduced to 84.43μA after calibration. Hence, counters of size greater than 14 flip-flops or equivalent Trojans can be detected by using process calibration techniques. To further increase sensitivity, we use the TeSR approach which can detect Trojans having more than 2 flip-flops and is limited only by measurement noise of 2.76μA. Any smaller Trojan will activate its malicious payload in less than 4 cycles and be detected using logic testing approaches. Note that, since TeSR compares the difference in Trojan activity over multiple time windows, the difference metric values are less than the normalized I_{DDT} metric used in the process-calibration approach.

5.3 Experimental Validation

Hardware validation of the proposed side-channel approach was performed using an FPGA-platform where FPGA chips were used to emulate the ASIC scenario. We wanted to observe the effectiveness of the proposed approach to isolate the Trojan effect in presence of process variations, when a golden design and its variant with Trojan are mapped to the FPGA devices. Such an FPGA-based test setup provides a convenient platform for hardware validation using different Trojan types, sizes and even different designs.

The selected FPGA device was Xilinx Virtex-II XC2V500 fabricated in 120nm CMOS technology. In order to measure I_{DDT}, we measured the voltage drop across a sense resistor (0.5Ω), using high-side current sensing strategy. To increase the accuracy of measurements amidst measurement noise, a custom test board was designed with the sense resistor connected between the core V_{DD} pins and the on-board bypass capacitors. A differential probe was used to measure the voltage waveforms, which were recorded using an Agilent mixed-signal oscilloscope (100MHz, 2Gsa/sec). The waveforms were synchronized with a 10 MHz clock input from the oscilloscope and are recorded over 16 cycles corresponding to a pattern of 16 input vectors. A "SYNC" signal is used to correspond to the first input vector in the set, so that the current can be measured for

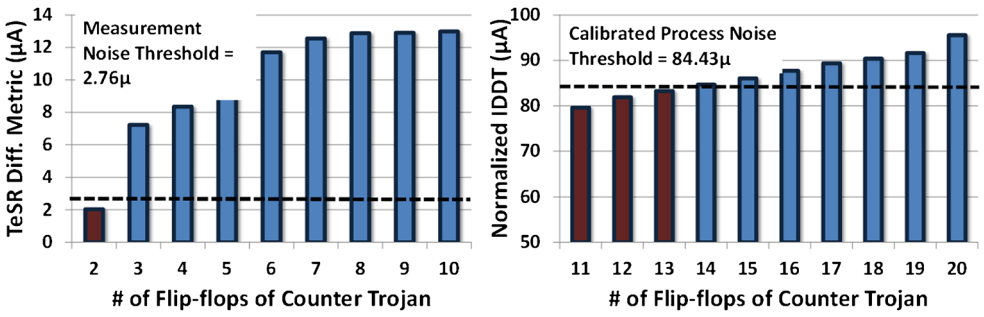

Fig. 13 Difference Metric for varying size of a sequential Trojan inserted in 32-bit IEU circuit, using TeSR and other process-calibration approaches

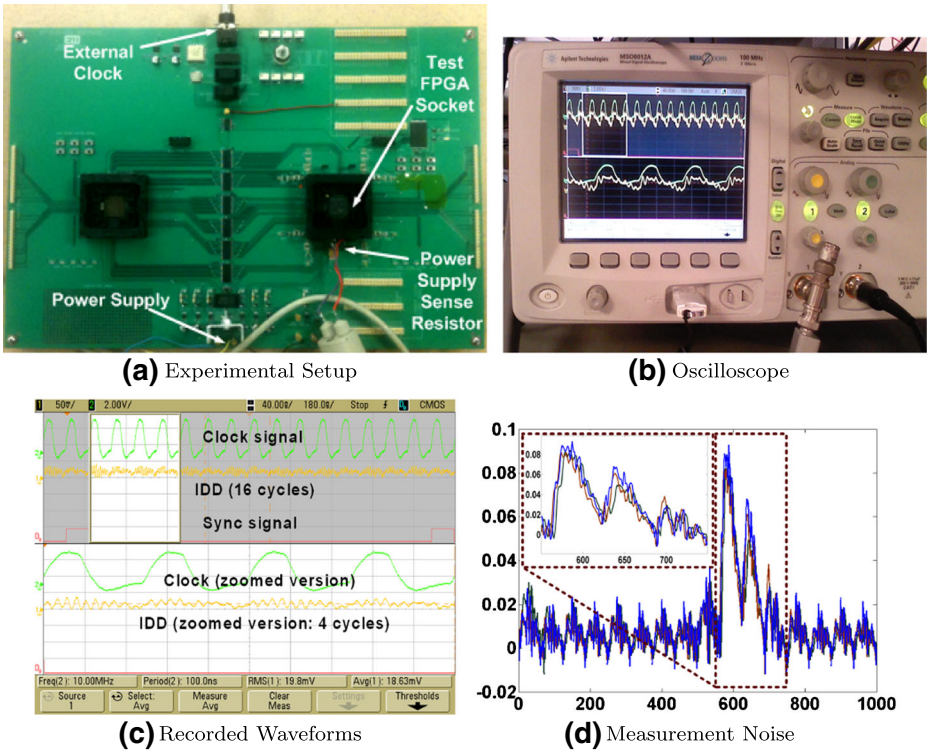

Fig. 14 Experimental setup using FPGA-based board and measured current waveforms for validating the TeSR approach

the same vectors in all cases. The test setup is shown in Fig. 14.

To observe the effect of measurement noise and other temporal variations in our simulation results, we used the characteristics of the noise obtained in real measurements (see Fig. 14d) to generate random Gaussian noise in MATLAB, which was used in our simulations. Here, we performed measurements for the DLX processor mapped to different FPGA chips. The varying current signature for three FPGA chips at different process corners is shown in Fig. 15. One of the chips contains an 8-bit counter Trojan. *It should be noted that the Euclidean distances drawn for the two golden dies are shown in microamp scale whereas for infected die, it is plotted in milliamp scale*. It can be clearly observed that the Trojan-containing instance has a difference metric which falls above the noise threshold.

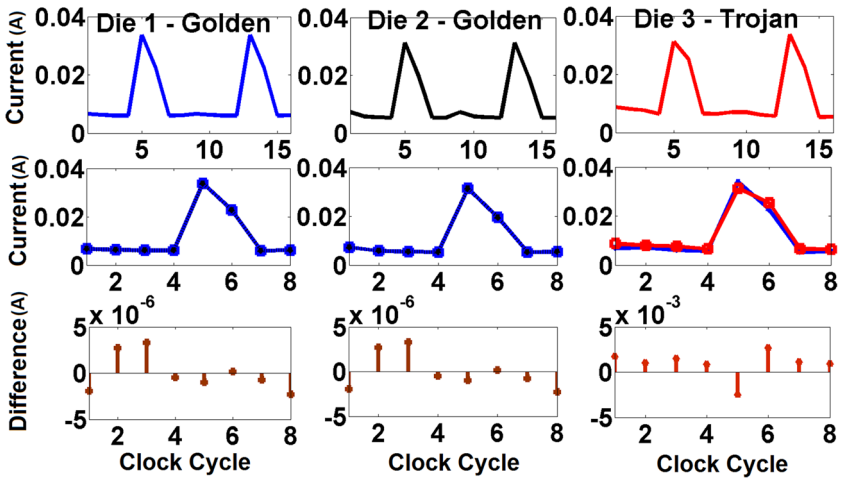

Fig. 15 Measurement results for DLX with 8-bit counter Trojan. Euclidean distances drawn for the two golden dies are plotted in microamp resolution whereas for infected die, it is plotted in milliamp scale. The polarity of the Euclidean measurement has been shown to indicate rise and fall of current

6 Conclusion

In this paper, we have presented TeSR, a hardware Trojan detection approach aimed at detecting sequential hardware Trojans, which are a type of Trojans more difficult to isolate and more capable to perform various malicious functions compared to their combinational counterparts. The proposed approach provides higher detection sensitivity under large process noise, and hence is suitable for nanoscale process technologies.

It facilitates detection of small, rarely-activated sequential Trojans, which can be extremely difficult to detect using existing logic testing or side-channel approaches. The approach leverages on the uncorrelated temporal variations in transient current signature of sequential hardware Trojans to isolate their effect from process and measurement noise. By comparing current signature of a chip for the same input pattern at different time windows, it can completely eliminate the effects of both die-to-die and local within-die parameter variations, as well as various design marginalities, which can cause local deviations in current signature leading to large number of false positives/negatives. The proposed approach also eliminates the need of golden or reference ICs, which are difficult and highly expensive to obtain. The simulation and experimental validation results verify that the proposed method can be very effective in isolating chips with hard-to-detect sequential Trojans of varying forms and size, which can easily evade logic testing and other side-channel approaches.

Appendix A: Reachability Analysis

Algorithm 2 elaborates the reachability analysis, which is based on breadth-first traversal. S_0 is the root state under consideration. G is the FSM state transition graph (STG) in adjacency-list representation, in which each edge (corresponding to one state transition) has an associated property indicating the set of input vectors that can trigger this transition $v(S_1, S_2)$. The reason of using adjacency-list instead of adjacency-matrix representation is that most FSM STGs are sparse graphs, and adjacency-list representation can also favor the image computation of each state. *Reached* stands for the set of states reachable from S_0, which is the goal of the entire calculation. $Frontier$ represents the current frontier states as the breadth-first traversal proceeds. Function $Img(S_i, G)$ calculates the states that are reachable by S_i in one step, and is defined as follows, where S is the set of states in G, and $E \subseteq S \times S$ is the set of edges in G:

$$Img(S_i, G) = \{S' \in S \mid (S_i, S') \in E\} \quad (4)$$

In fact, the image computation can be easily realized by looking into the adjacency-list of the root state, as all the directly reachable states are stored in the same list. As implied by the name, breadth-first traversal expands the search uniformly across the frontier, during which the input vector set dictated by the transition function property $v(S_1, S_2)$ is appended to that of the previous path, and the sequence of input vector sets is associated to each newly identified reachable state as property $S_j \cdot I$. The iterative process is continued until no new states beyond *Reached* are experienced, namely $Frontier$ is empty.

Algorithm 2 FSM reachability analysis

```
Reached = Frontier = Img(S0, G)
for each Sj ∈ Frontier do
  Sj · I = v(S0, Sj)
end for
while Frontier ≠ φ do
  Frontier1 = φ
  for each Si ∈ Frontier do
    Temp = Img{Si, G}
    for each Sk ∈ Temp do
      if Sk ∈ Reached then
        Temp = Temp − Sk
      else
        Sk · I = ⟨Sk · I, v(Si, Sk)⟩
      end if
    end for
    Frontier1 = {Frontier1, Temp}
  end for
  Frontier = Frontier1
  Reached = {Reached, Frontier1}
end while
```

Appendix B: Proof of the Impracticality of Low Overhead Correlated Trojan

Definition 1 A state machine F_o could be expressed as: $F_o = \{S_o : S_o$ is the set of states, $T_o : T_o$ is the set of transition paths$\}$

Definition 2 Function P determines the consumed power during kth clock cycle (or transition) by F_o during a given test trial due to a transition from state $S_{o,i}$ and $S_{o,j}$ over the path $T_{o,k}$.

$$Power_{o,k} = P\left(S_{o,i} \xrightarrow{T_{o,k}} S_{o,j}\right)$$

TeSR checks if for the same $T_{o,k}$, $Power_{o,k}$ is equal in different test trials (i.e. for trial n and $n+1$, $Power_{o,k,n} = Power_{o,k,n+1}$).

Definition 3 Function C determines the clock cycles required to move F_o from state $S_{o,i}$ to $S_{o,j}$.

$$Clock_{o,k} = C\left(S_{o,i} \rightarrow S_{o,j}\right)$$

Definition 4 S_{in}: Initial states in F_o from which MERO patterns are being applied.

Definition 5 S_{re}: Re-initializing states in F_o from which S_{in} is reached back to initiate the next test trial.

Theorem 1 *A TeSR undetectable state machine F_e that is a correlated version of F_o exists if and only if $|T_e| \geq |T_o| \vee |S_e| \geq |S_o|$*

Proof If $|T_e| < |T_o|$, assume $T_o - T_e = T_x$.

To make F_e undetectable: $S_{o,in} \in |S_e|$ and $S_{o,re} \in |S_e|$.

If $T_x \in (S_{o,in} \rightarrow S_{o,re})$, then after F_o and F_e are traversed through path $S_{o,in} \rightarrow S_{o,re}$ simultaneously for n trials:

$$Clock_{o,k,n} \neq Clock_{e,k,n}.$$

Therefore, we can assume that after $Clock_{o,k}$, during the same test trial:

$$Current\_State(F_o)_n = S_{o,re} \neq Current\_State(F_e)_n.$$

Consequently, after $Clock_{o,k}$ in two different test trials:

$$Current\_State(F_o)_n = Current\_State(F_o)_{n+1}.$$

$$Current\_State(F_e)_n \neq Current\_State(F_e)_{n+1}.$$

Therefore, $Power_{e,k,n} \neq Power_{e,k,n+1}$.

Furthermore, if $|S_e| < |S_o|$, assume $S_o - S_e = S_x$.

Since for any S_x, corresponding T_x(s) exists: it can be stated that: $Power_{e,k,n} \neq Power_{e,k,n+1}$.

Therefore we have established that if $|T_e| < |T_o| \vee |S_e| < |S_o|$, state machine F_e would be detected by TeSR. □

References

1. [Online]. Available: www.opencores.org
2. Aarestad J, Acharyya D, Rad R, Plusquellic J (2010) Detecting Trojans though leakage current analysis using multiple supply pad IDDQs. IEEE Trans Inf Forensics Secur
3. Abramovici M, Bradley P (2009) Integrated Circuit security - new threats and solutions, CSIIR Workshop, pp. 1–3
4. Agrawal D, Baktir S, Karakoyunlu D, Rohatgi P, Sunar B (2007) Trojan detection using IC fingerprinting, Proc. IEEE Symposium on Security and Privacy
5. Alkabani Y, Koushanfar F (2009) Consistency-based characterization for IC Trojan detection, Proc. International Conference on Computer-Aided Design (ICCAD)
6. Banga M, Hsiao M (2008) A region based approach for the identification of Hardware Trojans, Proc. IEEE Workshop on Hardware Oriented Security and Trust (HOST)
7. Banga M, Hsiao MS (2009) A novel sustained vector technique for the detection of hardware Trojans, Proc. 22nd International Conference on VLSI Design, pp 327–332
8. Bao C, Forte D, Srivastava A (2014) On application of one-class SVM to reverse engineering-based hardware Trojan detection, ISQED
9. Bhunia S, Hsiao MS, Banga M, Narasimhan S (2014) Hardware Trojan Attacks: Threat Analysis and Countermeasures. In: Proceedings of the IEEE 102.8, pp 1229–1247
10. Bloom G, Narahari B, Simha R, Zambreno J (2009) Providing secure execution environments with a last line of defense against Trojan circuit attacks. Comput Secur
11. Borkar S, Karnik T, Narendra S, Tschanz J, Keshavarzi A, De V (2003) Parameter variations and impact on circuits and microarchitecture, DAC
12. Chakraborty RS, Bhunia S (2009) HARPOON: an obfuscation-based SoC design methodology for hardware protection. IEEE Trans Comput Aided Des Integr Circuits Syst 28.10:1493–1502
13. Chakraborty RS, Narasimhan S, Bhunia S (2009) Hardware Trojan: Threats and emerging solutions, High-Level Design Verification and Test Workshop
14. Chakraborty RS, Wolff F, Paul S, Papachristou C, Bhunia S (2009) MERO: A statistical approach for hardware Trojan detection, CHES Workshop
15. DARPA (2007) TRUST in Integrated Circuits (TIC). [Online]. Available: http://www.darpa.mil/MTO/solicitations/baa07-24
16. Du D, Narasimhan S, Chakraborty RS, Bhunia S (2010) Self-referencing: A scalable side-channel approach for hardware Trojan detection, CHES Workshop
17. Forte D, Bao C, Srivastava A (2013) Temperature tracking: An innovative run-time approach for hardware Trojan detection, ICCAD
18. Huang Y, Bhunia S, Mishra P (2016) MERS: Statistical Test Generation for Side-Channel Analysis based Trojan Detection, CCS, Vienna
19. Jin Y, Makris Y (2008) Hardware Trojan detection using path delay fingerprint, HOST
20. Jin Y, Sullivan D (2014) Real-time trust evaluation in integrated circuits, DATE
21. Karri R, Rajendran J, Rosenfeld K, Tehranipoor M (2010) Toward trusted hardware: Identifying and classifying hardware Trojans. IEEE Commun Mag
22. Koushanfar F, Mirhoseini A (2011) A unified framework for multimodal submodular Integrated Circuits Trojan detection. IEEE Trans Inf Forensics Secur 6(1)
23. Kulikowski KJ, Venkataraman V, Wang Z, Taubin A (2008) Power balanced gates insensitive to routing capacitance mismatch, DATE
24. Kundu S, Zachariah ST, Chang Y-S, Tirumurti C (2005) On modeling crosstalk faults. IEEE Trans Comput-Aided Design
25. Kuon I, Rose J (2007) Measuring the gap between FPGAs and ASICs. IEEE Trans Comput Aided Des Integr Circuits Syst 26.2
26. Lamech C, Rad RM, Tehranipoor M, Plusquellic J (2011) An experimental analysis of power and delay signal-to-noise requirements for detecting Trojans and methods for achieving the required detection sensitivities. IEEE Trans Inf Forensics Secur
27. Lin L, Burleson W, Parr C (2009) MOLES: malicious off-chip leakage enabled by side-channels, ICCAD
28. Liu Y, Huang K, Makris Y (2014) Hardware Trojan detection through golden chip-free statistical side-channel fingerprinting, DAC
29. Meade T, Zhang S, Jin Y (2016) Netlist reverse engineering for high-level functionality reconstruction, ASP-DAC

30. Nahiyan A, Xiao K, Forte D, Jin Y, Tehranipoor M (2016) AVFSM: a framework for identifying and mitigating vulnerabilities in FSMs, DAC
31. Narasimhan S, Du D, Chakraborty RS, Paul S, Wolff F, Papachristou C, Roy K, Bhunia S (2010) Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach, HOST
32. Narasimhan S, Wang X, Du D, Chakraborty RS, Bhunia S (2011) TeSR: A Robust Temporal Self-Referencing Approach for Hardware Trojan Detection, HOST
33. Nowroz AN, Hu K, Koushanfar F, Reda S (2014) Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 33.12:1792–1805
34. Potkonjak M, Nahapetian A, Nelson M, Massey T (2009) Hardware Trojan horse detection using gate-level characterization, DAC
35. Predictive Technology Model, [Online] http://www.eas.asu.edu/~ptm/
36. Rad R, Plusquellic J, Tehranipoor M (2010) A sensitivity analysis of power signal methods for detecting hardware Trojans under real process and environmental conditions. IEEE Trans Very Large Scale Integr VLSI Syst
37. Rajendran J, Jyothi V, Sinanoglu O, Karri R (2011) Design and analysis of ring oscillator based Design-for-Trust technique, VTS
38. Roy JA, Koushanfar F, Markov IL (2008) EPIC: Ending piracy of integrated circuits, DATE
39. Salmani H, Tehranipoor M, Plusquellic J (2010) A layout-aware approach for improving localized switching to detect hardware Trojans in Integrated Circuits, IEEE Intl. Workshop on Information Forensics and Security
40. Shi Y et al (2010) A highly efficient method for extracting FSMs from flattened gate-level netlist, ISCAS
41. Soll O, Korak T, Muehlberghuber M, Hutter M (2014) EM-based detection of hardware Trojans on FPGAs, HOST
42. Tehranipoor M, Koushanfar F (2010) A survey of hardware Trojan taxonomy and detection. IEEE Design and Test of Computers 27(1):10–25
43. Tiri K, Verbauwhede I (2004) A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation, DATE
44. Wei S, Potkonjak M (2011) Scalable hardware Trojan diagnosis, IEEE Tran. on Very Large Scale Integration (VLSI)
45. Xiao K, Zhang X, Tehranipoor M (2013) A clock sweeping technique for detecting hardware Trojans impacting circuits delay. IEEE Design & Test of Computers, March/April, pp 26–34
46. Xiao K, Forte D, Jin Y, Karri R, Bhunia S, Tehranipoor M (2016) Hardware Trojans: Lessons Learned after One Decade of Research, vol 22.1
47. Yoshimizu N (2014) Hardware Trojan detection by symmetry breaking in path delays, HOST
48. Zhang X, Tehranipoor M (2011) RON: An on-chip ring oscillator network for hardware Trojan detection, DATE
49. Zhang J, Yu H, Xu Q (2012) HTOutlier: Hardware Trojan detection with side-channel signature outlier identification, HOST
50. Zheng Y, Yang S, Bhunia S (2016) SeMIA: Self-Similarity based IC Integrity Analysis. IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 35(1):37–48
51. Zhou B, Adato R, Zangeneh M, Yang T, Uyar A, Goldberg B, Unlu S, Joshi A (2015) Detecting Hardware Trojans using backside optical imaging of embedded watermarks, DAC

Tamzidul Hoque is pursuing Ph.D. in Electrical and Computer Engineering at the University of Florida (UF). At UF, Tamzidul has been working as a research assistant at the Florida Institute for Cybersecurity (FICS) under the supervision of Dr. Swarup Bhunia. His research is focused on hardware security, with emphasis on hardware Trojan detection, device and IP security. He has completed summer internship in Cisco at their RTP campus as a PhD intern within the Security and Trust Organization. Previously he completed M.S. in Electrical Engineering from the University of Toledo, Ohio in 2015 and B.Sc. in Electrical and Electronic Engineering from Ahsanullah University of Science and Technology (Bangladesh) in 2012.

Seetharam Narasimhan is currently working as the Lead Security Researcher at the Security Center of Excellence, Platform Engineering Group of Intel Corporation, Hillsboro, Oregon, USA. He obtained a Ph.D. in Computer Engineering from Case Western Reserve University (USA) in 2012 and a B.E. (Hons.) in Electronics and Telecommunication Engineering from Jadavpur University (India) in 2006. His research interests include: Hardware Security, Ultralow power and reliable nanoscale circuits, as well as Bio-medical circuits and systems. He is the coauthor of three book chapters, and more than 40 publications in international journals and conferences of repute.

Xinmu Wang graduated with a Ph.D. degree from the department of Electrical Engineering and Computer Science, Case Western Reserve University. She received her B.E. in Microelectronics from Harbin Institute of Technology, Harbin, China in 2008. Her current research interests include hardware security including Trojan detection and high temperature VLSI design.

Sanchita Mal-Sarkar received her Ph.D. from the department of Electrical and Computer Engineering, Cleveland State University, OH, USA and M.S. degree from MS, Computer Science, University of Windsor, Canada. Currently she is serving an Associate Lecturer in Cleveland State University. Her research interest lies in hardware security, and Fault tolerant Networks, Soft/ Granular Computing and Risk Assessment Wireless Sensor Networks.

Swarup Bhunia received his B.E. (Hons.) from Jadavpur University, Kolkata, India, and the M.Tech. degree from the Indian Institute of Technology (IIT), Kharagpur. He received his Ph.D. from Purdue University, IN, USA, in 2005. Currently, Dr. Bhunia is a professor in the department of Electrical and Computer Engineering at University of Florida, Gainesville, FL, USA. Earlier, Dr. Bhunia has served as the T. and A. Schroeder associate professor of Electrical Engineering and Computer Science at Case Western Reserve University, Cleveland, OH, USA. His current research interests include low power and robust design, hardware security, trust and protection, and computing with emerging technologies. Dr. Bhunia serves in the technical program and organizing committee of many premier conferences and in the editorial board for several journals. He was a recipient of the National Science Foundation Career Development Award, the IBM Faculty Award, the Semiconductor Research Corporation Technical Excellence Award, the SRC Inventor Recognition Award, and several best paper awards.