Test Pattern Superposition to Detect Hardware Trojans

Chris Nigh and Alex Orailoglu
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California
chnigh@cs.ucsd.edu, alex@cs.ucsd.edu

***Abstract*—Current methods for the detection of hardware Trojans inserted by an untrusted foundry are either accompanied by unreasonable costs in design/test pattern overhead, or return results that fail to provide confident trustability. The challenges faced by these side-channel techniques are primarily a result of process variation, which renders pre-silicon expectations nearly meaningless in predicting the behavior of a manufactured IC. To overcome this hindrance in a cost-effective manner, we propose an easy-to-implement test pattern-based approach that is self-referential in nature, capable of dissecting and understanding the characteristics of a given manufactured IC to hone in on aberrant measurements that are demonstrative of malicious Trojan hardware. By leveraging the superposition principle to cancel out non-Trojan noise, we can isolate and magnify Trojan circuit effects, all within a regime considerate of practical test and design-for-test infrastructures. Experimental results performed on Trust-Hub benchmarks demonstrate the proposed method provides a clear and significant boost in our ability to confidently certify manufactured ICs over similar state-of-the-art techniques.**

I. Introduction

As electronics components have become a greater part of everyday life, ensuring their trustworthiness has become increasingly critical. This has been demonstrated by the need for cutting-edge certification techniques in test, verification, validation, and other areas of the design flow, and similarly advanced methods are now needed to certify manufactured ICs are not corrupted by malicious circuitry. If undetected, these *hardware Trojans* may have catastrophic consequences, like denying essential services or leaking sensitive information.

The detection of Trojans inserted by an untrusted foundry has been targeted primarily by side-channel methods, which aim at producing and observing those second-order anomalies like increased circuit delay [1], power [2] [3], temperature [4], and other signals [5] that may point to an IC's deviation from the norm. However, the significant process variation effects of recent technology nodes create a paradigm in which some level of deviation is common and expected among non-Trojan ICs. This raises the bar for Trojan detection techniques, requiring a level of sophistication to discern which atypical signals are reasonable, and which are not.

In many side-channel methods though, this sophistication does not come for free. Some strategies have proposed design enhancements to improve the likelihood of Trojan detection, resulting in area overhead or design complexity that may not be affordable in all cost/performance product domains [6] [7] [8]. Other test pattern-based methods are significantly easier to implement, but often require an unreasonably large amount of test patterns, or provide results that are unable to deliver the needed confidence to truly certify trust [9] [10].

In an effective side-channel test pattern-based method, there are typically two somewhat contradictory goals to consider: 1) cover as much of the design as possible in hopes of hitting a Trojan, and 2) reduce the impacts of process variation from the non-Trojan circuit. This leaves the fundamental conundrum of power-based side-channel methods for Trojan detection: how can a methodology activate large portions of the design to produce the desired Trojan signal, while also preventing overshadowing the signal with the increased process variation noise from other activated parts of the circuit?

We are thus left with a paradigm that requires advanced and novel techniques to cut through these challenges. We must further magnify the signal of the Trojan, even in the face of the logistic challenges raised by conforming standard test and design-for-test methodologies. In this paper, we present a method that leverages the superposition principle for this purpose, enabling us to cancel out the process variation noise of a given IC-under-certification and leave a significant Trojan circuit signal in its place.

The key contributions of this work are as follows:

1) A novel proposal to use self-referencing superposition on circuit activity to enhance side-channel Trojan detection
2) A methodology to guide this superposition under practical test/DFT considerations
3) An approach which requires no additional hardware overhead, unlike many previous Trojan detection proposals
4) Demonstration of near-certain successful Trojan detection on Trust-Hub Trojan Benchmarks under even extreme magnitudes of process variation

II. Background

A. Trojan Attack Threat Model

The work of this paper focuses on *invisible* Trojan attacks that are not present in the pre-silicon netlist, likely inserted by an untrusted manufacturer.[1] These Trojan circuits are commonly modeled with a two-part structure, consisting of a *Trigger* and a *Payload* [1]. The Trigger acts as the gate-keeper, keeping the Trojan hidden from defenders by forming activation criteria dictated by the attacker. A satisfied Trigger will activate the Payload, which performs the malicious behavior that may corrupt or intercept some signal in the original circuit. With the Trigger aiming to shield the Trojan from functional detection, the attacker is motivated to make its full activation conditions near-impossible to sensitize by chance.

[1]This is in contrast to the *visible* Trojan attacks which are present in pre-silicon netlists, likely injected by an untrusted designer or third-party IP supplier. This visible model motivates more efficient detection through circuit analysis methods like [11] or [12].

978-3-9819263-4-7/DATE20/©2020 EDAA

B. Related Work

1) Design-based Methods: To improve the ability to control, observe, or isolate the suspicious area, some approaches elect to insert additional hardware into the Trojan-free circuit, similar to the *Design-for-Test* logic used to enhance design testability. Methods designed around improving Trojan-related activity often target those low-activation probability portions of the design which an attacker may be more likely to use as input criteria for their Trojan. In [13], dummy scan flops are inserted as additional test points, turning those previously low-probability signals into ones that can be influenced directly. The approach described in [14] facilitates gate-level characterization through test point insertion to break reconvergent paths that may challenge delay measurements.

Similarly, other detection approaches may elect to enhance the design to induce increased per-region isolation, reducing the potential signal that can originate from those non-suspicious areas. In [15], existing scan chains in a design are reordered to facilitate improved activation of one circuit region at a time. In a similar manner, [7] alters the clock tree structure and scan flop configuration to promote physical adjacency of equal-power regions for use in a comparative analysis that mitigates the impacts of intra-die process variation.

2) Test Pattern-based Methods: Other methods focus on the selection of particular test patterns rather than design information. To improve the likelihood of Trojan activation, [9] takes a statistical approach by promoting multiple switching events in those rarely activated nets. [10] uses a strategy built on a genetic algorithm to tackle a similar objective.

In [16] the alternate approach is taken, using a per-region testing technique to limit overall circuit activity by generating random test patterns which target logic of only a single region at a time. [17] proposes a self-referencing method in which similar pattern sequences applied at different time stamps can be compared to identify activity from sequential Trojans. In [18], the authors employ strategic test pattern selection to leverage side-channel benefits in delay and voltage analysis.

3) Superposition: The principle of superposition for linear systems dictates that the net composition of responses for multiple independent processes is equivalent to the response of those processes applied concurrently. This concept has previously been applied to address VLSI-related problems in analysis of delay [19] or temperature [20], as well as diagnosis [21] [22], where multiple observations from a given system are used to filter away benign effects or build up interesting ones.

Methods for side-channel Trojan detection have also used somewhat similar concepts, either through the use of superposition in temporal self-referencing to identify side-channel anomalies [23], or through using linear programming ideas to characterize individual gate performance [14]. While both certainly have benefits, both also have drawbacks, either in costly design overhead, ineffectiveness on certain types of Trojans, or lack of scalability to reasonably-sized designs. As will be described, the superposition applied in this paper is used differently, in a more targeted manner to neutralize noise-inducing non-Trojan circuit effects.

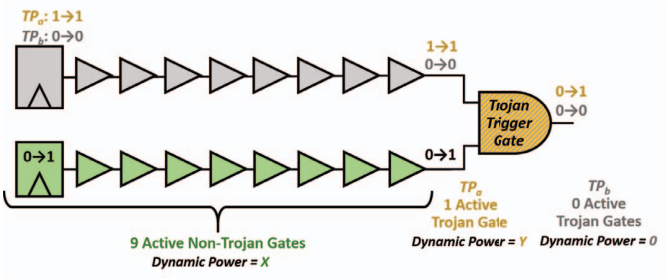

Fig. 1: Test pattern pair leveraging superposition to fully-magnify Trojan signal.

III. Superposition for Hardware Trojan Detection

A. Motivation for Further Enhancement

While design-based methods can provide detection value, it is apparent that they may not be viable in many cost- or performance-sensitive design environments. This immediately reduces the set of ICs that can be certified, which may critically impact the overall security in today's multi-chip heterogeneous systems. Therefore, this cannot be reasonably deemed a comprehensive solution to invisible Trojan attacks.

Conversely, though test-pattern based methods are a much more applicable solution across product types, it is clear that the achievements of current methods either fall well short of the high bar for Trojan detection, or may start to venture into an exhaustive-like search, which is by its nature prohibitively expensive. A test pattern-based technique which is both effective and feasible is clearly needed, which is the objective of the method proposed in this paper.

B. Key Challenge Of Single Test Pattern Trojan Detection

Consider the common scenario in which, by the nature of our test/design-for-test infrastructure and the combinational logic of the design, any test which partially activates a Trojan will inherently require the production of some additional ancillary circuit activity that cannot be eliminated. Many of these potential counter-productive phenomena can arise: our need to produce non-controlling Trojan values may require the creation of activity in some other ancillary paths; transitions which reach Trojan gates must first propagate through power-consuming non-Trojan circuitry; even the scan cells required to initiate the transitions will themselves create switching activity. All of this activity in non-Trojan portions of the circuit will reduce the relative magnitude of the Trojan circuit and create a higher degree of process variation noise.

Take Figure 1 as a prime example of this challenge. With multiple levels of non-Trojan combinational logic sitting between scan cells and the Trojan gates, any activating transition which produces the desired Trojan signal must also necessarily create activity in those intermediate non-Trojan gates. Each of the non-Trojan gates that must be traversed to reach the Trojan will effectively reduce the magnitude of the Trojan signal, and increase the amount of process variation noise.

In this case, there are nine non-Trojan gates lying along this path, such that even if we can effectively isolate this one path and entirely remove all other circuit activity, at best we can have one of ten activated gates be from the Trojan

circuit. While this will produce a significant magnification of the Trojan power signal, the process variation impacts of the nine non-Trojan gates may still be too great to overcome.

Furthermore, it is often unlikely that we will be able to eliminate all of the other ancillary path activity in the circuit due to the practical considerations of typical test application techniques. Often, scan chain or combinational logic configuration will force some additional transitions in order to produce static non-controlling values which are needed for Trojan gate activity. These scenarios will produce even more non-Trojan activity and more process variation noise, opening the potential for many paths which cannot be reasonably quieted.[2]

C. Multiple Test Pattern Superposition as an Alternative

While process variation may prevent accurate prediction from pre-silicon analysis, note that within a given IC, the process variation impacts are fixed. Thus, if a certain activity created on a set of gates in a single IC produces a given power response, we would expect a similar response when that same activity is again created on that same set of gates on that same IC. As this principle is extended to separate but similar test patterns, we can create multiple observations of this same activity, while additional uniquely-activated regions produce distinguishing power characteristics between patterns. By composing the behaviors of these test patterns into a broader view, we can attempt to factor away the common behavior and instead highlight the uniquely-observed effects.

Refer back to Figure 1, and the pair of test patterns TP_a and TP_b which produce differing values on the non-transitioning scan cell, resulting in differing Trojan gate activity. While it might seem counter-intuitive, the second test pattern TP_b which deactivates the Trojan gate is especially valuable. With this pattern's reading, we have identified the switching power associated with this set of non-Trojan gates, including their process variation effects. When compared with the first test pattern TP_a, we identify that for two test patterns that are expected to have the exact same activity, we get two very different power readings. This application of superposition identifies a difference between our first and second test patterns which isolates and exposes the Trojan signal to its full magnitude, giving the ideal case for Trojan detection.

However, similar to the earlier highlighted challenges, this ideal magnification may not always be feasibly possible within the framework of typical test application techniques, as modifications in test pattern stimulus to change a non-controlling value to a controlling one may produce new transitions in the circuit. Where allowed by the scan chain or combinational logic configuration, superposition can be used as an effective means to cancel or mitigate the effects of other circuit activity.

With superposition, we aim for test patterns that have significant overlap, cancelling out their common effects and leaving only those smaller, unique portions uncovered. For this reason, in a Trojan detection methodology, we need to have the Trojan itself in this difference, magnified as the common activity is eliminated. Therefore, when considering two test patterns, one of them should activate the Trojan, while the other should deactivate the Trojan. In the above example this is demonstrated perfectly, with the first test pattern creating Trojan activity and the second removing it, leaving the Trojan signal standing alone when superposition is applied.

[2]While the particularities of how these cases may arise depend on other design-for-test and test pattern application methodology decisions, any technique used in attempts to ensure avoidance of such cases will typically incur excessive design cost overhead.

IV. Proposed Methodology

A. Test Regime Considerations

While the use of superposition appears effective in theory, there are critical considerations that must be given to the feasibility of implementing such an approach. As a key example, there are two ways in which transition-producing test patterns can be applied to a circuit. In the *Launch-on-Capture* (LOC) technique, the value sitting at the data input of each scan cell is launched at the next clock pulse [24]. This could be catastrophic in our paradigm which values activity control, as there is the potential for a large number of input bit value definitions required to satisfy the creation or removal of a transition, and each of those definitions could themselves have unintended collateral impact to other parts of the circuit.

We find the alternative *Launch-on-Shift* (LOS), which leverages the preceding scan cell in the scan chain to launch data [25], to be a much more viable method for our purposes, as the ability to influence a path's transition is directly identifiable and controllable from the input test pattern stimulus. Whenever two adjacent bits are set to opposite values (...01... or ...10...) at the position of a scan cell, it is immediately apparent that a transition will be launched from that scan cell. This also allows us to easily understand the potential transition-creating or transition-reducing effects of changes we make to the test pattern without the need for circuit simulation.

Nonetheless, this aspect of LOS does introduce unique challenges when given test patterns are modified, as the circuit activity produced by each bit is directly related to its immediately adjacent values. Changes made in the name of desired circuit activity production/reduction have the potential for counter-productive effects depending on the surrounding values. The transparency that LOS provides highlights the interrelated nature of test pattern values, which will be addressed, handled, and even leveraged in the proposed methodology.[3]

B. Leveraging Adaptivity

While superposition can clearly provide significant detection benefits, it, much like some other potential test pattern-based methods, could not reasonably be applied as is to the entire circuit. Test pattern sets aimed at covering the whole design are necessarily going to be quite diverse, as the designs are too large to allow spending of critical test pattern resources on similar regions. This is in contrast to the benefits provided by superposition, which is better suited as a fine-grained analysis working on an identified signal that we would like

[3]LOC is typically preferred as a manufacturing test application technique, as it requires scan cells to functionally observe defect effects. This requirement can incur high design overhead in a LOS scheme (primarily through at-speed scan enable signal timing closure). Functional observation is not a requirement in power-based side-channel Trojan detection, enabling us to leverage LOS with the existing lower-overhead architecture by simply launching data through shift, reading its behavioral effects, and forgoing the capture pulse used for functional observation.

to magnify by eliminating the surrounding effects. As such, another method is required to identify and place our test patterns in a position where superposition techniques can take over to achieve more complete magnification effects.

From a test pattern which produces some initial power measurement, we must first sense whether there appears to be a potential Trojan-related signature embedded within it. However, there will often be too much non-Trojan noise to motivate this lower-level study. Instead, we must build confidence in a signal, which will allow us to make a smart investment. This can be done through an adaptive technique, which aims at reducing activity in the circuit by making small modifications to the test pattern and evaluating the elevated/degraded suspicious signal. By pursuing those potential Trojan-related effects, we can boost the confidence that our investment may uncover the signal for which we search.

Note that in order for this adaptive method to get started in Trojan signal magnification and deem superposition worth employing, there must first exist a Trojan signal to magnify. Our selection of LOS has further benefits here, as the improved controllability over LOC lends itself to easier transition production and increased overall coverage [26]. Through experimentation on Trust-Hub benchmarks, LOS TDF tests generated on Trojan-free designs by commercial ATPG tools are reliably capable of partial Trojan activation when applied to the corresponding Trojan-inserted design. While this suggests standard ATPG may be viable, the adaptive methodology is agnostic as to the source of the test pattern (provided LOS is used). This strategy could easily pair with other methods like [9] or [10] to improve Trojan activation likelihood.

C. Application of Superposition

As stated above, let us assume a Trojan gate exists in the circuit and is activated by our original test pattern. This adaptive approach's consistent reduction in scan cell transitions will guarantee that some adjacent pair of test patterns will meet the needs of superposition, with one pattern activating the Trojan and the next deactivating it. As these pattern modifications are made, we can also observe and analyze the magnitude of any changes to such a signal. The observation of a suspiciously-large drop in the magnitude of this atypical signal will be our indicator to deploy superposition in attempts to expose it further by reducing ancillary activity.

When such a case has been identified, we pause our adaptive approach and place a focused effort on the analysis of this test pattern pair through an application of superposition to consider their activity differences and commonalities. Note that as we are defending against invisible Trojan attacks, we are aware of only the non-Trojan circuitry, which will be used in determining the reasonableness of the pair of observed power measurement. By highlighting potential abnormal relationships between the responses of these test patterns, any Trojan circuitry can be exposed should it exist.

D. Modification for Signal Enhancement

With increased overlap in activity between adjacent test patterns, the unique activity decreases. With the reduction of this unique activity in non-Trojan circuitry while maintaining key differences in Trojan circuit activity, we have the opportunity to magnify and expose the Trojan signal. As demonstrated in Section III-C, the ideal case would provide full overlap in non-Trojan activity, with differentiation only on the Trojan circuit.

#	Modification	Original		Updated
1	*Introduce Two Transitions*	00000	$\longrightarrow$	00**1**00
	Eliminate Two Transitions	11011	$\longrightarrow$	11**1**11
2	*Move Transition Right*	000111	$\longrightarrow$	000**0**11
	Move Transition Left	000111	$\longrightarrow$	00**1**111
3	*Introduce Single Transition*	11111	$\longrightarrow$	**0**1111
	Eliminate Single Transition	00001	$\longrightarrow$	0000**0**

Fig. 2: Suite of strategic test pattern modifications.

While we may expect the identified pair of adjacent test patterns to be relatively similar to one another, we recognize that this resulting pair may potentially be sub-optimal, leaving possible opportunities to produce greater overlap in activation. To improve upon the initial application of superposition, we can perform strategic test pattern modifications to more closely align the pair, creating an even further magnified Trojan signal.

For a moment, let us assume that there is at least one Trojan gate which changes activation state between the pair of identified adjacent test patterns, as desired. This suggests that there must be at least one bit in the test pattern stimulus which assumes differing values between the two test patterns, and which has impacted the Trojan in only one of the two test patterns to either:

1) produce a transition which reaches the Trojan gate,
2) help propagate a transition to the Trojan gate, or
3) help produce non-controlling value on a Trojan input.

Maintaining the status of this altered bit will be key, as to effectively apply superposition we must have one test pattern which activates the Trojan and another that deactivates it. While retaining this key desired difference, we seek out and leverage other opportunities to improve the alignment of the two test patterns for those remaining undesired differences in ancillary non-Trojan activity. In light of this problem statement, there are three key test pattern modification methods that can be deployed in our search.

1) Introducing/Eliminating Two Transitions: In some scan chain locations, our pair of test patterns may differ by a single bit. If this discrepancy occurs in the midst of other similar-valued bits (ex: 010 becomes 000), there may be two transitions which are eliminated with the alteration ($0 \rightarrow 1$ becomes $0 \rightarrow 0$, and $1 \rightarrow 0$ becomes $0 \rightarrow 0$). If this activity is unrelated to the Trojan signal, we would prefer the two patterns match behavior, either by both or neither producing the transitions. By altering a test pattern to introduce/eliminate this transition, we can create the desired effect.

2) Moving the Location of a Transition: Changes in test pattern stimulus may potentially not alter the number of transitions, but rather alter their location. By changing the value of a bit lying on the boundary of an existing transition, the launch point of the transition is moved by one bit forward or backward (right or left) in the scan chain. If the difference in transition location between our two test patterns is unrelated to the Trojan behavior, we can choose to align the behavior of the test patterns and create further activity overlap.

Additionally, Trojan activation may be benefited not by the transition itself, but by two bits of differing values present on the same scan chain. This requirement will necessarily induce a transition, producing ancillary activity differences between our adjacent test patterns. However, we can attempt to relocate the transition launch position on the scan chain (within the bounds of our critical bits) to minimize the unique activity.

3) Introducing/Eliminating a Single Transition: Much like the two transition case making modifications in the middle of a scan chain, there is also the opportunity for similar modifications to be made at the ends of a scan chain. Such changes will introduce/eliminate only a single transition, providing a smaller footprint of activity difference. This can be leveraged if there happens to be a two-transition difference between patterns, where one of the two is necessary for Trojan activation. Instead of retaining both, we can consider keeping only the relevant transition while eliminating the ancillary one through "pushing" the undesired transition off the end of the scan chain. This can be quite effective in limiting the magnitude of test pattern differences in certain instances.

V. Experimental Results

A. Evaluation Criteria

To evaluate the magnitude of a given atypical signal of a single test pattern, one may reasonably use the *Relative Power Difference* (RPD) metric proposed in [7]. In this formulation, a given test pattern TP_i is applied, and $RPD(TP_i)$ computes the magnitude of the difference between expected nominal power P_{Ni} and the observed power P_{Oi} with Equation 1.

$$RPD(TP_i) = \frac{P_{Oi} - P_{Ni}}{P_{Ni}} \quad (1)$$

We would like to apply a similar formulation to our new superposition method, focusing on the power response differences between two test patterns TP_a and TP_b. The development of this RPD extension to cover the dual test pattern form requires slightly lower-level analysis.

Suppose TP_a and TP_b activate sets of gates G_a and G_b, of which there is some common overlapping activated gates $G_{\text{cmn}} = G_a \cap G_b$ and some gates uniquely activated by each pattern $G_{a_{\text{unq}}} = G_a \cup \neg G_b$ and $G_{b_{\text{unq}}} = \neg G_a \cup G_b$. From each of these sets of gates, we can identify the corresponding nominal power $P_{N\text{cmn}}$, $P_{Na_{\text{unq}}}$, and $P_{Nb_{\text{unq}}}$. Using this enhanced variable set we develop Equation 2 as the dual test pattern version of RPD, which we'll call $Super\text{-}RPD$ ($S\text{-}RPD$).[4]

$$S\text{-}RPD(TP_a, TP_b) = \frac{(P_{Oa} - P_{Ob}) - (P_{Na} - P_{Nb})}{P_{Na_{\text{unq}}} + P_{Nb_{\text{unq}}}} \quad (2)$$

B. Experimental Setup

The proposed method is evaluated on the five ISCAS gate-level combinational Trojan benchmarks available from Trust-Hub [27], [28]. All initial LOS TDF test patterns were generated using ATPG on the Trojan-free netlist with Mentor's Tessent tool [29], and modified using custom scripts working on STIL files. The per-cell dynamic power consumption values are from the Synopsys SAED 90nm standard cell library [30].

[4]Note the denominator of Equation 2. In analyzing the differences between two test patterns, we recognize that the magnitude of the process variation effects influencing our suspicious signal is not a function of the difference in power between two test patterns, but rather a function of the sum of the unique power created by the two.

C. Trojan Magnification Achievements

Table I compares the Trojan magnification effectiveness of the original *TDF* test pattern from ATPG, the final test pattern achieved by the adaptive flow alone, the direct application of superposition on adjacent test patterns (from Section IV-C), and the final application of superposition with strategic test pattern modifications (from Section IV-D). For each of these different approaches, the corresponding RPD or $S\text{-}RPD$ value is provided, showing the magnitude of the Trojan signal, along with the gate activity-focused *Trojan-to-Circuit Activity* (TCA) metric from [15].

For all of the benchmark circuits, application of superposition is able to significantly improve the Trojan signal magnitude. In all cases, we surpass a Trojan signal magnitude of 10%, a mark which was not achieved by either ATPG-based or adaptive techniques. Trojan detection capabilities will therefore extend well beyond existing test pattern construction methods. As is also shown, there is measurable improvement in Trojan magnitude through the use of strategic modifications, proving yet further benefits with consideration given to the practical test regime. Direct comparison with other methods is somewhat challenging, given the wide variance in metrics, benchmarks, and reference points for improvement. We will note that other recent test pattern-based techniques [8] [9] boast improvements of at most an order of magnitude over random test patterns – in this work we show magnification of more than $150\times$ in all cases over the typically-stronger test patterns produced by ATPG.

D. Trojan Detection Achievements

Consider these achievements within a process variation environment, and we are able to make significant claims about true Trojan detection capabilities. Consider the two key components of process variation effects, which are *inter-die* variation (with magnitude defined by σ_{inter}) and *intra-die* variation (with magnitude defined by σ_{intra}). Due to the self-referencing nature of this methodology, there is no opportunity for inter-die variation to disrupt behavior. Therefore, the only uncertainty remains as a result of intra-die variation, in which separate gates within the same circuit may have different power consumption characteristics.

As a result, we can use $S\text{-}RPD$ to determine the magnitude of the intra-die variation that can effectively be covered through our enhanced superposition technique. To perform this computation, we assume that the Trojan circuitry does not exist, focusing only on those non-Trojan gate activity differences between the test patterns. By computing the maximum possible $S\text{-}RPD$ achievable under a given intra-die process variation magnitude, we can determine the effectiveness of our achievements in terms of true Trojan circuit detection.

To demonstrate how such a claim can be deduced, suppose the intra-die power variation for a given manufacturing process is $3\sigma_{intra} = \varsigma$. With this assumption, we can rewrite Equation 2 into the below Equation 3 to show the maximum possible observable $S\text{-}RPD$ value is ς itself.

TABLE I: Trojan Signal Isolation Achievements with Various Approaches

Trojan Benchmark	Orig. ATPG-Based		Adaptive Flow		Superposition Alone		Strategic Modification		RPD Mag. over ATPG-Based	RPD Mag. over Adaptive
	RPD	TCA	RPD	TCA	$S\text{-}RPD$	TCA	$S\text{-}RPD$	TCA		
s35932-T200	0.00088	0.0039	0.015	0.0435	0.177	0.50	0.195	0.667	**221.6×**	**13.0×**
s35932-T300	0.00085	0.0044	0.019	0.0952	0.0864	0.222	0.259	0.667	**304.7×**	**13.6×**
s38417-T100	0.00022	0.0013	0.0012	0.00326	0.0282	0.143	0.136	0.50	**618.2×**	**113.3×**
s38417-T200	0.0014	0.0040	0.0019	0.00509	0.0755	0.154	0.218	0.50	**155.7×**	**114.7×**
s38584-T100	0.00071	0.0024	0.084	0.267	0.0922	0.25	0.210	0.667	**295.8×**	**2.5×**

$$
\begin{aligned}
S\text{-}RPD(TP_a, TP_b) = &\Big[\Big((P_{N_{\text{cmn}}} + (1+\varsigma)P_{N a_{\text{unq}}}) - (P_{N_{\text{cmn}}} + (1-\varsigma)P_{N b_{\text{unq}}})\Big) \\
&- \Big((P_{N_{\text{cmn}}} + P_{N a_{\text{unq}}}) - (P_{N_{\text{cmn}}} + P_{N b_{\text{unq}}})\Big)\Big] \\
&\Big/ P_{N a_{\text{unq}}} + P_{N b_{\text{unq}}} \\
=&\frac{\Big((1+\varsigma)P_{N a_{\text{unq}}} - (1-\varsigma)P_{N b_{\text{unq}}}\Big) - \Big(P_{N a_{\text{unq}}} - P_{N b_{\text{unq}}}\Big)}{P_{N a_{\text{unq}}} + P_{N b_{\text{unq}}}} \\
=&\frac{\varsigma P_{N a_{\text{unq}}} + \varsigma P_{N b_{\text{unq}}}}{P_{N a_{\text{unq}}} + P_{N b_{\text{unq}}}} = \varsigma
\end{aligned}
\tag{3}
$$

By comparing this observation with results from Table I, we can compute the portion of the process variation distribution which is reliably covered, and thus the probability of Trojan detection. For example, under a Trojan detection methodology which achieves a $S\text{-}RPD = 2\varsigma$, we can state that detection is reliable up to $6\sigma_{intra}$, or with probability of $> 99.99\%$.

The proposed methodology's detection capabilities under various degrees of intra-die variation are demonstrated by Table II on the same set of Trojan benchmarks. In all cases, Trojan detection is a near certainty, with all benchmarks achieving at least 94% likelihood of detection even under the extreme and unlikely case of $3\sigma_{intra} = 25\%$. It is essentially guaranteed that the method can detect these abnormal Trojan-related signals even in the face of process variation noise, and can help ensure trustability of manufactured ICs.

VI. Conclusion

While hardware Trojan detection has proven to be a difficult problem due to the significant effects of process variation, there are clear improvements that can be made to enhance our abilities. Through strategic application of test patterns on a given IC-under-certification, we can attempt to better understand these process variation characteristics, and apply sophisticated superposition techniques to magnify suspicious circuit behavior. This has the opportunity to turn equivocal, possibly Trojan-related signals into unequivocal, positively Trojan-related ones, should they exist. By paying close attention to the practical considerations of such a test pattern-based regime, this is achieved without expensive design enhancements. As demonstrated by the experimental results on Trust-Hub benchmarks, the method proposed in this paper drives toward a more complete and effective method to ensure trustable ICs.

TABLE II: Trojan Detection Likelihood w/ Intra-Die Variation

Trojan Benchmark	Achieved $S\text{-}RPD$	Various Potential Values of $3\sigma_{intra}$				
		5%	**10%**	**15%**	**20%**	**25%**
s35932-T200	0.195	$> 99.99\%$	$> 99.99\%$	$> 99.99\%$	99.83%	99.04%
s35932-T300	0.259	$> 99.99\%$	$> 99.99\%$	$> 99.99\%$	99.99%	99.91%
s38417-T100	0.136	$> 99.99\%$	$> 99.99\%$	99.67%	97.93%	94.84%
s38417-T200	0.218	$> 99.99\%$	$> 99.99\%$	$> 99.99\%$	99.95%	99.56%
s38584-T100	0.210	$> 99.99\%$	$> 99.99\%$	$> 99.99\%$	99.92%	99.41%

References

[1] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *Intl. Workshop on Hardware-Oriented Security and Trust*. IEEE, 2008, pp. 51–57.

[2] D. Agrawal *et al.*, "Trojan detection using IC fingerprinting," in *Symp. on Security and Privacy*, May 2007, pp. 296–310.

[3] X. Wang *et al.*, "Hardware Trojan detection and isolation using current integration and localized current analysis," in *DFT of VLSI Systems*, Oct 2008, pp. 87–95.

[4] D. Forte *et al.*, "Temperature tracking: An innovative run-time approach for hardware Trojan detection," in *Intl. Conf. on CAD*, 2013, pp. 532–539.

[5] X.-T. Ngo *et al.*, "Hardware Trojan detection by delay and electromagnetic measurements," in *Design, Automation, & Test in Europe*, 2015, pp. 782–787.

[6] T. F. Wu *et al.*, "TPAD: Hardware Trojan prevention and detection for trusted integrated circuits," *IEEE Trans. on CAD*, vol. 35, no. 4, pp. 521–534, April 2016.

[7] F. S. Hossain *et al.*, "Variation-aware hardware Trojan detection through power side-channel," in *Intl. Test Conf.*, Oct 2018, pp. 1–10.

[8] X. Mingfu *et al.*, "Detecting hardware Trojan through heuristic partition and activity driven test pattern generation," in *Comm. Security Conf.* IET, 2014, pp. 1–6.

[9] Y. Huang *et al.*, "Scalable test generation for Trojan detection using side channel analysis," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 11, pp. 2746–2760, Nov 2018.

[10] S. Saha *et al.*, "Improved test pattern generation for hardware Trojan detection using genetic algorithm and boolean satisfiability," in *Intl. Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2015, pp. 577–596.

[11] A. Waksman *et al.*, "FANCI: Identification of stealthy malicious logic using boolean functional analysis," in *Conf. on Computer & Communications Security*. New York, NY, USA: ACM, 2013, pp. 697–708.

[12] J. Zhang *et al.*, "VeriTrust: Verification for hardware trust," *IEEE Trans. on CAD*, vol. 34, no. 7, pp. 1148–1161, July 2015.

[13] H. Salmani *et al.*, "A novel technique for improving hardware Trojan detection and reducing Trojan activation time," *IEEE TVLSI*, vol. 20, pp. 112–125, Jan 2012.

[14] S. Wei *et al.*, "Provably complete hardware Trojan detection using test point insertion," in *Intl. Conf. on CAD*, Nov 2012, pp. 569–576.

[15] H. Salmani and M. Tehranipoor, "Layout-aware switching activity localization to enhance hardware Trojan detection," *IEEE Trans. on Information Forensics and Security*, vol. 7, pp. 76–87, Feb 2012.

[16] M. Banga and M. S. Hsiao, "A region based approach for the identification of hardware Trojans," in *Intl. Workshop on Hardware-Oriented Security and Trust*, June 2008, pp. 40–47.

[17] T. Hoque *et al.*, "Golden-free hardware Trojan detection with high sensitivity under process noise," *Journal of Electronic Testing*, vol. 33, no. 1, pp. 107–124, Feb 2017.

[18] S. Narasimhan *et al.*, "Hardware Trojan detection by multiple-parameter side-channel analysis," *IEEE Trans. on Computers*, vol. 62, no. 11, pp. 2183–2195, Nov 2013.

[19] K. Shu-Min Li *et al.*, "Unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI," in *Asian Test Symp.*, Nov 2004, pp. 145–150.

[20] J. Park *et al.*, "Fast computation of temperature profiles of VLSI ICs with high spatial resolution," in *Semiconductor Thermal Measurement and Management Symp.*, March 2008, pp. 50–54.

[21] I. Bayraktaroglu and A. Orailoglu, "Improved fault diagnosis in scan-based BIST via superposition," in *Design Automation Conf.*, June 2000, pp. 55–58.

[22] B. Arslan and A. Orailoglu, "Fault dictionary size reduction through test response superposition," in *Intl. Conf. on Computer Design*, Sept 2002, pp. 480–485.

[23] S. Narasimhan *et al.*, "TeSR: A robust temporal self-referencing approach for hardware Trojan detection," in *Intl. Symp. on Hardware-Oriented Security and Trust*, June 2011, pp. 71–74.

[24] J. Savir, "Skewed-load transition test: Part I, calculus," in *Intl. Test Conf.*, Sept 1992, pp. 705–714.

[25] J. Savir and S. Patil, "Broad-side delay test," *IEEE Trans. on CAD*, vol. 13, no. 8, pp. 1057–1064, Aug 1994.

[26] J. Rearick, "Too much delay fault coverage is a bad thing," in *Intl. Test Conf.*, Nov 2001, pp. 624–633.

[27] H. Salmani *et al.*, "On design vulnerability analysis and trust benchmarks development," in *Intl. Conf. on Computer Design*, Oct 2013, pp. 471–474.

[28] B. Shakya *et al.*, "Benchmarking of hardware Trojans and maliciously affected circuits," *Journal of Hardware and Systems Security*, pp. 85–102, April 2017.

[29] *Tessent Shell Reference Manual*, Mentor Graphics, May 2018.

[30] Synopsys, "SAED_EDK90_CORE - 90nm digital std. cell library," 2008.