

**An Effective and Sensitive Scan Segmentation Technique for Detecting Hardware Trojan**

Fakir Sharif HOSSAIN\(^{(a)}\), Nonmember, Tomokazu YONEDA\(^{1}\), and Michiko INOUE\(^{1}\), Members

**SUMMARY** Due to outsourcing of numerous stages of the IC manufacturing process to different foundries, the security risk, such as hardware Trojan becomes a potential threat. In this paper, we present a layout aware localized hardware Trojan detection method that magnifies the detection sensitivity for small Trojan in power-based side-channel analysis. A scan segmentation approach with a modified launch-on-capture (LoC) transition delay fault test pattern application technique is proposed so as to maximize the dynamic power consumption of any target region. The new architecture allows activating any target region and keeping others quiet, which reduces total circuit toggling activity. We evaluate our approach on ISCAS89 benchmark and two practical circuits to demonstrate its effectiveness in side-channel analysis.

**key words:** hardware Trojan, scan segmentation, LoC patterns, power side-channel analysis, TDGP

1. Introduction

Over the last decade, the threat of hardware Trojans in ICs has concentrated investigation by the researchers and governmental entities. Hardware Trojan is defined as a malicious modifications to the existing circuit elements. It creates vulnerability in designing and fabricating of ICs with globalization and outsourcing for cost reduction. Indeed, traditional test methods fall short in revealing hardware Trojans, as they are intended towards identifying modeled defects and, therefore, cannot disclose unmodeled malicious inclusions.

Having the knowledge of IC fabrication and testing, an adversary can design a Trojan that cannot be activated or detected with traditional functional and structural tests [1]. While numerous hardware Trojan detection approaches have been explored in the literature, side-channel analysis has been the most influential investigated among them [2]. In a power-based side-channel signal analysis, it is possible to extract the Trojan contributions even in the presence of process variation and various types of noise. In some literatures, dynamic power based side-channel analysis has proven to be highly effective in extracting information about the internal operation of designs [3], [4]. Although many approaches related to functional and timing analysis are presented for hardware Trojan detection [5], in this paper, we focus on the methods that use dynamic power analysis to target hardware Trojans in integrated circuits.

The Trojan detection methods are proposed based on transient current in [6] and steady state current in [7]. These methods investigate how to extract signals from Trojan against process variation and other noises. However, these do not address how to magnify signals from Trojans. Region based partition and relative toggle count magnification techniques are applied to increase the switching in the circuit in a test-per-clock fashion [8]. The method is not layout-aware and does not consider the distribution of switching activity across the layout of circuit in practice. In [9], the authors present a heuristic scan partition with activity driven test pattern generation to detect Trojan by dynamic power analysis. They target one region while keeping other scan chains all have 0 vectors, which in turn can minimize noise. However, their scan partition is large and unable to detect Trojans if it is stealth in nature. A scan-cell reordering method is proposed to localize switching activity in an IC-Under-Authentication (IUA) [10]. The method is layout-aware and practical for detecting Trojans. However, the proposed partitions are based on scan chains and one partition might be still large for very large circuit. Thus a small contribution of Trojan is difficult to identify by Trojan-to-circuit activity. Also, they do not consider any specific test vector modification (random only).

The scan segmentation approach is a well-known technique to minimize power consumption during test. In this paper we use this technique for Trojan detection by maximizing dynamic power consumption of the Trojan inserted regions instead. Our proposed detection technique is similar to the method of [10], but the basic difference is that we consider small scan segmentation and Trojan detection golden patterns (TDGP) generation technique so that tiny Trojans can be identified. We focus on the side-channel methods that use power analysis to target hardware Trojans in integrated circuits. The major contributions of this paper are as follows.

- Our scan chain repartitioning and segmentation are layout aware, thus we can identify the locations of Trojans.
- Higher detection sensitivity. This proposed method has a freedom of getting any number of regions and is capable of restricting background switching from other regions.
- Optimize the detection time by selecting a small number of TDGPs.

The experimental results for ISCAS’89 benchmark circuits
and practical circuits evince the effectiveness of the proposed method for both combinational and sequential Trojans.

The organization of this paper is as follows. Section 2 provides the overview of the proposed method. In Sect. 3 we describe our experimental results to demonstrate the effectiveness of our approach. A comparative discussion with related works is carried out in Sect. 4, and finally, Sect. 5 summarizes our main results and future work.

2. Proposed Method

Initially, we address some issues about our proposed detection method. First, our assumption is that Trojans are inserted in fabrication stage. If Trojans are inserted into the design or layout phase it is easy to identify them as we have a complete handle on layout level characteristics. Our concern in the toughest one (in physical chip). Second, in the detection step we need golden fingerprints from golden IC to compare with the measured value from manufacturing ICs. We propose a practical method of detecting hardware Trojan, which, if fabricated as in real chip would be capable of detecting Trojan effectively. We divide our proposed method into two phases: the design phase, which consists of scan chain repartition, scan segmentation and Trojan detection pattern generation including a modified LoC test technique. In the detection phase, the Trojan detection patterns are applied to circuits and the power is measured, which later on are compared to the fingerprints from a golden IC.

The overall flow diagram is presented in Fig. 1. Initially, in the design phase for a given circuit and its physical placement information about scan cells, we perform scan chain repartitioning and reordering to create Trojan detection-aware scan chains. Then, each scan chain is partitioned further into smaller segments using clock-gating technique so that each segment has independent clock control of others. Next, transition delay fault test patterns are generated with automatic test pattern generation (ATPG) tool. We modify them so that only one segment receives the launch and capture clock while freezing other segments to maximize the detection sensitivity for small Trojans. Formerly, we perform a logic simulation to select patterns for each segment as TDGP.

In the detection phase TDGPs are applied to target circuit and the dynamic power is measured. The measured powers are compared with golden fingerprints and if a detection criterion, which will be discussed in Sect. 2.5, is satisfied, we can say the circuit is Trojan inserted or not. The next subsection describes a detail view of our proposed technique.

2.1 Layout-Aware Scan Chain Repartitioning

Scan-cell reordering techniques has already been proposed to reduce power during scan test [11] or minimize routing constraints in scan paths [12]. In this work, we also develop a scan chain repartitioning method, but for exploiting the power consumption to detect Trojans. Our main goal behind repartitioning scan chain is to get a sensitive partition of every circuit so that any Trojan (small or large) can be detected efficiently. Algorithm-1 is actually the same as a repartitioning algorithm presented in [9]. This algorithm is used not only to partition into scan chain but also to determine the order of scan cells in each scan chain. We can further partition one scan chain into segments in the order of scan FFs in the chain. The proposed algorithm takes layout information; the number of scan-chains and the number of scan cells as input and give a reordered design as output. Note that the proposed scan chain partitioning does not conflict with power reduction during a test or scan chain routing minimization since the method localizes scan cells in the same scan chain.

Algorithm-1 recursively partition one scan cell group into two groups. The groups are determined according to

![Algorithm 1: Scan Chain Repartitioning](image)
their geometric position in the circuit. If $\Delta X$ is greater than $\Delta Y$, all scan cells are ordered in ascending form and cut in the middle and vice versa. Thus, by this recursive cut finally we get equal number of scan cells in each scan chain and these are layout aware cells. After that, each scan cell in the group are stitched together to form scan chain. The order of reconnections is maintained in the direction of X-axis or Y-axis depending on CUT function. Finally, the reordered design is saved.

An example to show how the scan chains are repartitioned and reordered based on Algorithm-1 is depicted in Fig. 2. Figure 2 (a) is the output of IC compiler for s1238 benchmark circuit. To partition this circuit, initially, stitches are removed among FFs but keeping their position fixed. Then partition points are determined by the FFs position (see Fig. 2 (b)). First, $N_{cut}$ is calculated from number of scan chains, for example, s1238 has 2 scan chains, so $N_{cut}$ equal to 1 ($\log_2(2)$). Second, the $\Delta X$, $\Delta Y$ are determined according to (b) and based on the condition ($\Delta X > \Delta Y$) the recursive procedure is performed 3 times. Finally, the partition point is created to group two scan chains according to their geometric position of FFs in the layout. The dotted line in (b) indicates the partitioned line. As for this circuit $\Delta X$ is greater than $\Delta Y$, Fig. 2 (c) shows how Algorithm-1 re-stitch the scan chain based on ascending order of the X-axis.

2.2 Proposed Scan Segmentation

The partitioned scan chain from previous procedure exhibits further segmentation in this step. Figure 3 shows the typical clocking to scan chains and the proposed clock gating technique. In the typical procedure, the global clock feeds all scan chains during scan in scan out (see Fig. 3 (a)) operation. In contrast, in our proposed scheme, the scan chain is split into a fixed number of length-balanced segments and each segment can be activated independently during test application. The gated clock controller (see Fig. 3 (b)) is the added components for that purpose, which contributes small hardware overhead. We would present the gating hardware cost in Sect. 3 which will show that for large circuit this cost can be neglected.

During scan in, all segments are activated, while only one target segment is activated with clock gating in the launch and capture cycle. Therefore, this designated scan segment is updated with new values from combinational core’s switching. Based on this hardware modification, we propose a new test application method to improve the Trojan detection sensitivity discussed in the next subsection.

2.3 Proposed Pattern Application Technique

In the typical LoC pattern application technique, in order to detect transition faults, we need to make a transition on a node and propagate a transition to the observable outputs. Therefore, two vectors ($v_1, v_2$) are needed in transition delay testing, where $v_1$ initialize the value of the node, $v_2$ makes a transition on the node and sensitizes the transition to the observable outputs. Once the scan data is loaded through scan chains, scan enable signal transitions to 0. Subsequently, the launch and capture clock are applied (see Fig. 4 (a)). The launch vector ($v_2$) is calculated from the response of a circuit under test (CUT) for the shifted in pattern $v_1$.

We consider the same LoC pattern but proposing a novel method to apply these test patterns for Trojan detection. The proposed test application technique is based on the LoC mode as shown in Fig. 4 (b). When the Scan Enable = 1, all segments are active and the first vector $v_1$ of LoC pair is shifted into scan chains. After the scan in is finished, Scan enable goes down and the first functional clock is applied to get the second vector $v_2$ of LoC pair to launch transitions in the combinational block. As this second vector of the pair is accountable for gate transition, we do evaluate the dynamic power consumption as our fingerprint at launch cycle only for Trojans detection. Our method uses LoC patterns since
such patterns are generated to activate gates related to target segments. In addition, we adopt LoC patterns, even for frozen scan chains since such inactive segments still contribute the activity of the target segment.

In the example shown in Fig. 4 (b), only the segment 1-2 is triggered to create transitions at the launch cycle. Then, the second functional clock (capture cycle) is applied and the response is captured. However, the response is ignored because our interest is to evaluate the power consumption at launch cycle for Trojans detection, not to detect faults.

2.4 TDGP Generation

As we discussed in the previous section, a small segment may magnify the impact of Trojan activity because there is less activity in the entire design. In contrast, small segment may lessen the activation probability of the Trojan since some of the Trojan inputs may be from inactive segments. In side-channel analysis, it is not necessary to fully trigger Trojan as partial activation seems to be enough to trace them. Also, the technology library has limitations in the number of gate inputs (or fan-in). The high fan-in gate greatly impacts the delay characteristics of the design and can easily be detected using delay-based techniques[1], [14]. Therefore, we propose a test pattern generation method that maximizes the toggling coverage only in the target segment.

We use test patterns generated from transition delay faults as our starting point since they are inherently designed to create a lot of switching at launch cycle. For each scan segment, we first modify the clock distribution scheme of the patterns so that only the target segment receives the launch clock while freezing other segments. Then, we perform a logic simulation to evaluate toggling of gates at the launch cycle for each pattern. The patterns from ATPG are now one of the inputs of our proposed Algorithm-2 to select a TDGP considering toggling coverage per circuit. A set of segments (S) and a maximum number of TDGPs are the rest two inputs for the proposed Algorithm-2 whose outputs are the selected TDGPs. Initially, for all the generated modified LoC patterns (i) we obtain the toggling gates. The WHILE loop selects TDGP patterns in a greedy fashion based on toggling coverage until reaching the maximum toggling coverage or the limit of pattern count. This step continues for all the segments of each circuit. Therefore, the number of TDGPs is different in different segments in S. Only the TDGPs are applied at the detection phase.

2.5 Detection Technique

The detection steps in our proposed method are to measure the power values and compare them with golden power fingerprints. We measure an IUA (IC under authentication) and a golden IC for several test patterns. We can detect the Trojan if the measured value exceeds some threshold. We consider how to determine a threshold value.

In addition, our proposed method does not separate scan path for Trojans detection as it keeps the number of scan chains constant and does not physically separate one scan chain. By clock gating, some part of the scan chain become inactive only at the launch-capture cycle in the Trojan detection. Normal test can be applied as usual. Therefore, this method doesn’t require additional testing time for normal test.

Fabricated ICs suffer a process variation and the measured values are different from nominal (ideal) values. The process variation affects the dynamic and the leakage power of transistors with a variance depending on a process technology and transistor size. Therefore, every source of power

**Algorithm 2: TDGP selection**

**Input:**
- S: set of segment
- V: set of LOC test patterns
- Pmax: maximum pattern count for the selection

**Output:**
- TDGP: set of Trojan detection golden patterns for segment i

```plaintext
For(i=1; i<= |S|; i++){
  Generate modified LoC patterns for i ( Vi)
  For(j=1; j<= |Vi|; j++){
    Obtain toggling gates by Vi[j] (Tij)
  } // End of For j
  Set TDGP = φ, Kbest = K = φ;
  While (|TDGP| < Pmax){
    For(i=1; i<= |Vi|; i++){
      Ktemp = Tij ∪ K
      If (|Ktemp| > |Kbest|){
        Kbest = Ktemp
        Vbest = Vi[j]
      } // End of For i
    } // End of For j
    If (K = Kbest) break
    TDGP = TDGP ∪ {Vbest}
    K = Kbest
  } // End of While
} // End of For i
```
is different from its nominal value within some factor determined by a process technology that is $|P - P_n| \leq \alpha P_n$ for an actual power $P$ and a nominal power $P_n$. Since the power consumed at a launch-capture cycle for a test pattern is a sum of dynamic and leakage power of power sources, an actual (or measured) total power $P_m$ is affected within a factor $\alpha$ from its nominal power $P_n$, that is $|P_m - P_n| \leq \alpha P_n$. This implies that the power difference between two circuits with the same design is different within a factor of $2\alpha$. Let us use the following notations to consider the threshold for Trojan detection.

- $P_m(T, t)$: measured power of a Trojan inserted circuit for a test pattern $t$
- $P_m(G, t)$: measured power of a golden circuit for $t$
- $P_n(T, t)$: nominal power of a Trojan inserted circuit for a test pattern $t$
- $P_n(G, t)$: nominal power of a golden circuit for $t$
- $\delta_T(|\delta_T| \leq \alpha)$: a factor of difference of a Trojan inserted circuit from a nominal one
- $\delta_G(|\delta_G| \leq \alpha)$: a factor of difference of a golden circuit from a nominal one

Factors of difference express effects of a process variation as follows.

$$P_m(T, t) = (1 + \delta_T)P_n(T, t)$$
$$P_m(G, t) = (1 + \delta_G)P_n(G, t)$$

We can detect a Trojan if measured power difference exceeds an acceptable difference. The threshold for Trojan detection is derived as follows.

$$P_m(T, t) - P_m(G, t) > (2\alpha + \beta)P_n(G, t)$$

where $\beta$ is a safety margin for detection.

With Eqs. (1) and (2), we can derive the following condition for Trojan detection.

$$(1 + \delta_T)P_n(T, t) - (1 + \delta_G)P_n(G, t) > (2\alpha + \beta)P_n(G, t)$$

Then the following condition is derived.

$$RDP(t) = \frac{P_n(T, t) - P_n(G, t)}{P_n(G, t)} > \frac{\delta_T - \delta_G + 2\alpha + \beta}{(1 + \delta_T)}$$

Since $\alpha, \beta, \delta_T, \delta_G$ are independent of the test pattern $t$, a ratio RPD of nominal power difference of $P_n(T, t) - P_n(G, t)$ to a nominal power of a golden IC $P_n(G, t)$ expresses a detection sensitivity for the test pattern $t$.

3. Experiment Results

For the validation of our proposed method, we apply our proposed technique on ISCAS89 benchmark circuits and 2 practical circuits, RS232 micro URT and AES crypto processor from Trust-HUB [13] (see Table 1). The original design is synthesized using the Synopsys Design Compiler and IC Compiler with 90nm technology.

### 3.1 Inserted Trojans

In our experiment, we use 2 types of Trojans of Trust-HUB benchmarks [13] named as T1 and T2* (see Table 2). T1 and T2* are classified as combinational cell dominant sequential Trojan and sequential cell dominant sequential Trojan, respectively. In general, sequential dominant Trojans are tough to be detected. To insert Trojan, we used an Engineering changed order (ECO) option for placement and routing. This option enables to insert Trojan without changing the layout of the original circuit and Trojan cells are inserted close to the wires that connect Trojans and the original circuit. Figure 5 for an example, displays positions of FFs in each segment and position of Trojan cells in particular segments (S4 and S8) for s13207 benchmark. It shows how segments are formed with scan cells in our method and how Trojan cells are inserted without changing the layout of original circuits. Here all Trojan cells are placed by freeze silicon ECO.

T1 is s15850-T100 in Trust-HUB which is a 32-bit comparator with two flip-flops (see Fig. 6(a)). The Trojan trigger consists of two comparators and one flip-flop (FF) at the output of each comparator. The comparators drive the clock inputs of the FFs. The output of the second FF is gated by the inverted test enable signal to ensure the Trojan acti-
Table 3
Test coverage for s35932 benchmark circuit.

<table>
<thead>
<tr>
<th>Segment</th>
<th>W/o TDGP selection</th>
<th>With segmentation (16 × 2 segments/scan chain)</th>
<th>W/o TDGP selection</th>
<th>With segmentation (16 × 2 segments/scan chain)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PORG, TCnt, TCov.</td>
<td>PORG, TCnt, TCov.</td>
<td>Pmax=14(=40% of PORG)</td>
<td>Pmax=14(=40% of PORG)</td>
</tr>
<tr>
<td>S1</td>
<td>35 116 2.39% 8 116 2.39%</td>
<td>35 116 2.39% 8 116 2.39%</td>
<td>35 116 2.39% 8 116 2.39%</td>
<td>35 116 2.39% 8 116 2.39%</td>
</tr>
<tr>
<td>S2</td>
<td>35 341 7.02% 14 338 6.95%</td>
<td>35 341 7.02% 14 338 6.95%</td>
<td>35 341 7.02% 14 338 6.95%</td>
<td>35 341 7.02% 14 338 6.95%</td>
</tr>
<tr>
<td>S3</td>
<td>35 144 2.96% 14 143 2.94%</td>
<td>35 144 2.96% 14 143 2.94%</td>
<td>35 144 2.96% 14 143 2.94%</td>
<td>35 144 2.96% 14 143 2.94%</td>
</tr>
<tr>
<td>S4</td>
<td>35 424 8.72% 14 408 8.39%</td>
<td>35 424 8.72% 14 408 8.39%</td>
<td>35 424 8.72% 14 408 8.39%</td>
<td>35 424 8.72% 14 408 8.39%</td>
</tr>
<tr>
<td>S5</td>
<td>35 154 3.17% 14 150 3.09%</td>
<td>35 154 3.17% 14 150 3.09%</td>
<td>35 154 3.17% 14 150 3.09%</td>
<td>35 154 3.17% 14 150 3.09%</td>
</tr>
<tr>
<td>S6</td>
<td>35 443 9.11% 14 435 8.95%</td>
<td>35 443 9.11% 14 435 8.95%</td>
<td>35 443 9.11% 14 435 8.95%</td>
<td>35 443 9.11% 14 435 8.95%</td>
</tr>
<tr>
<td>S7</td>
<td>35 141 2.90% 10 141 2.90%</td>
<td>35 141 2.90% 10 141 2.90%</td>
<td>35 141 2.90% 10 141 2.90%</td>
<td>35 141 2.90% 10 141 2.90%</td>
</tr>
<tr>
<td>S8</td>
<td>35 441 9.07% 14 433 8.91%</td>
<td>35 441 9.07% 14 433 8.91%</td>
<td>35 441 9.07% 14 433 8.91%</td>
<td>35 441 9.07% 14 433 8.91%</td>
</tr>
<tr>
<td>S9</td>
<td>35 130 2.67% 13 130 2.67%</td>
<td>35 130 2.67% 13 130 2.67%</td>
<td>35 130 2.67% 13 130 2.67%</td>
<td>35 130 2.67% 13 130 2.67%</td>
</tr>
<tr>
<td>S10</td>
<td>35 414 8.52% 14 404 8.31%</td>
<td>35 414 8.52% 14 404 8.31%</td>
<td>35 414 8.52% 14 404 8.31%</td>
<td>35 414 8.52% 14 404 8.31%</td>
</tr>
<tr>
<td>S11</td>
<td>35 159 2.86% 14 158 2.84%</td>
<td>35 159 2.86% 14 158 2.84%</td>
<td>35 159 2.86% 14 158 2.84%</td>
<td>35 159 2.86% 14 158 2.84%</td>
</tr>
<tr>
<td>S12</td>
<td>35 412 8.48% 14 402 8.27%</td>
<td>35 412 8.48% 14 402 8.27%</td>
<td>35 412 8.48% 14 402 8.27%</td>
<td>35 412 8.48% 14 402 8.27%</td>
</tr>
<tr>
<td>S13</td>
<td>35 132 2.72% 14 132 2.72%</td>
<td>35 132 2.72% 14 132 2.72%</td>
<td>35 132 2.72% 14 132 2.72%</td>
<td>35 132 2.72% 14 132 2.72%</td>
</tr>
<tr>
<td>S14</td>
<td>35 411 8.46% 14 390 8.02%</td>
<td>35 411 8.46% 14 390 8.02%</td>
<td>35 411 8.46% 14 390 8.02%</td>
<td>35 411 8.46% 14 390 8.02%</td>
</tr>
<tr>
<td>S15</td>
<td>35 35 2.78% 14 134 2.76%</td>
<td>35 35 2.78% 14 134 2.76%</td>
<td>35 35 2.78% 14 134 2.76%</td>
<td>35 35 2.78% 14 134 2.76%</td>
</tr>
<tr>
<td>S16</td>
<td>35 414 8.52% 14 391 8.04%</td>
<td>35 414 8.52% 14 391 8.04%</td>
<td>35 414 8.52% 14 391 8.04%</td>
<td>35 414 8.52% 14 391 8.04%</td>
</tr>
<tr>
<td>Total</td>
<td>560 4187 86.13%</td>
<td>213 4088 84.09%</td>
<td>290 4156 85.49%</td>
<td>341 4179 85.97%</td>
</tr>
</tbody>
</table>

3.2 Results for TDGP Selection and Its Coverage

Table 3 presents the selection criterion of TDGP and the selection result for s35932 benchmark for an example. The table shows the detail of original pattern count PORG, maximum pattern count constraint Pmax and selected pattern count PSEL, toggling count TCnt, along with toggle coverage (a ratio of active gates in an entire circuit) TCov. without TDGP selection or with TDGP selection under several Pmax constraints for several numbers of segments. For whole circuit (without segmentation), the number of ATPG LoC patterns is 35 and it covers 86.17% toggle coverage, however, it reduces with a little (0.02%) while segmenting with 35 × 16 = 560 patterns. For segments S1, S7, S9 and S13, they achieve the same toggle coverage as the original LoC patterns under the constraint of Pmax = 14. Especially, the segment S1 needs only 8 patterns.

Table 4 provides the overall toggle count and toggle coverage. We prepared ATPG LoC patterns with stable PI inputs, so some of the circuits have low toggle coverage even for the original ATPG patterns. The column “w/TDGP selection” shows the minimum number of selected patterns without compromising toggle count and coverage when applying all the TDGP patterns. We find that TDGP selection can reduce pattern count with preserving toggle coverage.

3.3 Overall Detection Results

For the validity of our proposed detection technique, Trojans in Table 2 are inserted as shown in Table 5. All the results are associated with 1.2V DC and 250MHz operating voltage only in the functional mode. Unless the Trojan gets activated, the output port functions as normal; otherwise, the trigger_select signal activate a Trojan payload circuit. In our experiment, we consider only Trojan’s triggering circuit not payload. For any successful triggering, Trojans can create circuit denial-of-service, hot spot by activating ring oscillator, do malfunction, get secret key for cryptography processor and so on.

Another Trojan, s38584-T200 from Trust-HUB is a rare vector triggering circuit which comprises of 32-bit adder with 32-bit counter. We didn’t use the same one from Trust-HUB because that Trojan (3122μm²) is too big and seems easy to be detected for small circuits like ISCAS89. Therefore, we modified s38584-T200 to 32-bit counter with 32-bit comparator named as T2* for large circuit like AES whose area is reduced by approximately 45% listed in Table 2. For the more detection sensitivity, we further reduce the area of T2* and modify it to 4-bit counter with 16-bit comparator named as T2 for small circuit (see Fig. 6(b)). The functionality of T2* and T2 is same. Trojan trigger is a counter of a rear vector. When the counter value is between 100 and 110, the Trojan payload circuit is activated.

Fig. 6 (a) Sequential comparator Trojan T1 (b) Sequential 4-bit counter T2.
frequency. In Table 5 we showed Trojan insertion and their detection results. The three columns under the Trojan insertion category designate the types of Trojans, their ratio to the total area and the location where they are inserted. The Trojan location means which segments the Trojan is sensitive to. That is, the Trojan is activated when FFs in the segment are activated and does not always imply that cells of Trojan are placed in the segment. For example, we inserted Trojan T1 in one segment S2 of s1238. That actually means that we applied TDGPs that activate S1 to the original circuit, checked toggle activities of cells, randomly selected active cells, and connected inputs of Trojan to the outputs of the selected cells. In this case, the Trojan is easily activated from TDGPs for inactive segments. In cases of s3578 and s38417 for T1, Trojans are inserted in two segments. In these cases, Trojans are sensitive to two segments and are not fully activated from one segment. In cases of s35932 for T2, s38584 and s38417 for T2, these seem to be a most difficult for Trojan detection. In these cases, Inputs of Trojan are randomly selected over entire circuits and each TDGP is not expected to activate Trojan well.

The detection results are shown in ‘with segmentation’ and ‘without segmentation’ where each case has three columns to show an activated segment, absolute power difference and RPD when the maximum RPD is obtained. We found that the proposed segmentation increased RPDs from cases without segmentation while achieving close absolute power difference. That means our proposed method increases the detection sensitivity. For a large circuit like AES, though the area ratio (0.11%) is very small, our technique can detect Trojans effectively. We inserted Trojans in a single segment for s1238, s13207, s38584 and RS232 circuits, whereas for circuits s5378 and s38417 we insert Trojan in adjacent two segments. Furthermore, we distributed Trojans over the entire circuit for s1238, s35932, s38417 and AES to show whether our technique can identify the locations of the Trojans. Table 5 shows that with little exception, we successfully identify the locations.

We also enlisted the maximum absolute power difference for a single TDGP in both cases with or w/o segmentation. Max absolute power differences are similar between cases. This means frozen patterns in inactive segments do not affect the toggle counts so much. Ratios RPDs of power difference in power of golden ICs are also shown in Table 5.

As mentioned, RPD represents a detection sensitivity, our method achieves much higher RPDs than cases without segmentation. For most circuits, including all the results of practical circuits, RPDs are increased by more than

---

**Table 4** Test coverage for benchmark circuits.

<table>
<thead>
<tr>
<th>Circuits</th>
<th>W/o segmentation</th>
<th>With segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1238</td>
<td>26</td>
<td>149</td>
</tr>
<tr>
<td>s3578</td>
<td>105</td>
<td>456</td>
</tr>
<tr>
<td>s13207</td>
<td>86</td>
<td>687</td>
</tr>
<tr>
<td>s35932</td>
<td>35</td>
<td>4189</td>
</tr>
<tr>
<td>s38417</td>
<td>172</td>
<td>4880</td>
</tr>
<tr>
<td>s38584</td>
<td>246</td>
<td>4495</td>
</tr>
<tr>
<td>RS232</td>
<td>72</td>
<td>775</td>
</tr>
<tr>
<td>AES</td>
<td>457</td>
<td>165154</td>
</tr>
</tbody>
</table>

---

**Table 5** Overall Trojan detection results for all benchmark circuits.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Inserted Trojan</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td>% of Trojan to circuit area</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Without Segmentation</td>
<td>With Segmentation (the max power diff. w/ TDGP = w/o TDGP selection)</td>
<td></td>
</tr>
<tr>
<td>s1238</td>
<td>T1</td>
<td>10.42%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>11.51%</td>
</tr>
<tr>
<td>s3578</td>
<td>T1</td>
<td>2.85%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>3.14%</td>
</tr>
<tr>
<td>s13207</td>
<td>T1</td>
<td>1.78%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>1.97%</td>
</tr>
<tr>
<td>s35932</td>
<td>T1</td>
<td>0.30%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>0.33%</td>
</tr>
<tr>
<td>s38584</td>
<td>T1</td>
<td>0.38%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>0.42%</td>
</tr>
<tr>
<td>s38417</td>
<td>T1</td>
<td>0.32%</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>0.36%</td>
</tr>
</tbody>
</table>

Practical circuits

|          | T1  |          | S2  | n/a | 72.9  | 9.1%   | S2  | 66.9  | 41.66% |
|          | T2  | 1.34%    | S2  | n/a | 55.3  | 10.5%  | S2  | 54.5  | 31.23% |
| AES      | T2* | 0.11%    | Entire ckt | n/a | 460   | 1.19%  | S22 | 413   | 13%    |
10% than the cases without segmentation. That means our method can effectively detect inserted Trojans even though both Trojan inserted circuit and golden IC are affected by a process variation. Figure 7 illustrates an example of relative power difference per TDGP. It indicates that if the variation occur up to 20%, our method can still detect Trojans.

In our experiments, we inserted Trojans in a single segment in some circuits, while the Trojans are distributed in two segments or an entire circuits in other circuits. Table 5 shows that RPDs are 9% to 51% and increased by 8% to 26% of cases without segmentation when Trojons are broadly distributed to an entire circuit. That means TDGPs can have sufficient activated Trojans and they can be detected even though they are distributed in the entire circuit.

Figure 8 presents the distributions of absolute power differences over segments for TDGPs that have enough high RPDs to detect inserted Trojans. These figures show the detailed information about the results shown in Table 5, where we show only the maximum power difference for an entire circuit. From these figures, we find that TDGPs achieve a high power difference to active segments and effectively detect Trojans. Figure 8 (c) shows a case where Trojan T2 is activated by one segment, S1. In this case, though the actual placement of Trojan is distributed over two segments, S1 and S2, the Trojan is activated by S1 and can be detected. Also Fig. 8 (f) shows a case for AES circuit where the Trojan T2* is distributed into an entire circuit. Though it is considered the most difficult case for Trojans detection, the Trojan is sensitively activated by some TDGPs in S2, S17, S22 and S27, even though they activate only one segment.

### 3.4 Detection Sensitivity Depending Segment Numbers

To analyze more deeply, what happens to RPDs if the segment size changes, we present Table 6 to show the detection sensitivity for s1238 circuit. The listed five TDGPs can detect the Trojan T1 in different sensitivity levels. We listed only those five TDGPs which have large RPDs. It can be observed that more segments achieve higher detection sensitivity. From this result, it can be stated that our proposed method has a degree of freedom to control the detection sen-

<table>
<thead>
<tr>
<th>TDGP ID</th>
<th>One Segment</th>
<th>Two Segments</th>
<th>Four Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Entire</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>17.02</td>
<td>16.8</td>
<td>0.92</td>
</tr>
<tr>
<td>2</td>
<td>6.66</td>
<td>30.22</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>6.16</td>
<td>11.04</td>
<td>0.32</td>
</tr>
<tr>
<td>4</td>
<td>23.05</td>
<td>14.78</td>
<td>0.76</td>
</tr>
<tr>
<td>5</td>
<td>11.34</td>
<td>25.04</td>
<td>0.88</td>
</tr>
<tr>
<td>max</td>
<td>23.05</td>
<td>30.22</td>
<td>45.58</td>
</tr>
</tbody>
</table>

**Fig. 7** TDGP ID vs. Relative power difference for T1 in s38417.

**Fig. 8** Detection results for (a) T1 in s38584 (b) T1 in s35932 (c) T2 in s5378 (d) T2 in s13207 (e) T1 in RS232 circuit and (f) T2* in AES circuit.
4. Discussion

We want to put a little discussion on the effectiveness or sensitivity of our technique in this section. As mentioned earlier in the Introduction section that X. Mingfu [9] and H. Salmani [10] have also proposed a layout aware Trojan detection method based on scan partitioning technique. The method [9] reorders scan cells to maximize power consumption while scan shifting. That means they try to maximize total power consumption during test application. However, both of their method and our method try to detect Trojan by measuring current from power ports, and we think peak power is more important to evaluate detectability. That is the reason why we introduce RPD to evaluate Trojan detectability.

The difference from our method is that their partitioning is based on scan chains. They don’t consider segments (small regions). Their method cannot activate the smaller region than the scan chain which as a result, would make tiny Trojans to be escaped. It is known that a small region may magnify the impact of Trojan activity on the design’s power profile because there is less activity in the entire design [10]. In contrast, large regions can increase the probability of generating a transition in a Trojan, but the Trojan impact can be lessened due to an increase of activity in the entire design. Thus, the %RPD may not so significant to guarantee small Trojan detectability.

Their method can activate one scan chain by shifting 0s to other scan chains. Practically, Trojan components can be distributed over the entire layout, and their inputs may originate from several different regions. As 0s in other scan chains keep scan registers inactive, Trojan inputs from those regions have less probability to activate the Trojan. On the other hand, our proposed scheme sets a test pattern for delay faults into all scan chains and freezes inactive segments only at the launch and capture cycle. The frozen pattern is applied to initialize signals along paths which are considered to propagate transitions at a launch cycle. Therefore, even though their values are frozen, it can effectively work to support such transitions. This is why the transition probability of Trojan inputs from regions kept inactive is higher.

Positively, they don’t need to gate clocks. This consideration can save hardware cost little extent. However, our method experiences higher order of freedom by receiving more segmentations per scan chain (increase sensitivity) with little penalty of gating hardware. In contrast, if the scan chain-based method provides the same number of regions as our proposed method, it needs more number of real scan chains. Which increases the number of scan chains and more number of I/O connections are required. Which is a great penalty in semiconductor industry. Compare to physical I/O port, gating hardware is less costly. Interestingly, for a large circuit this area overhead can be ignored [10] (see Table 1). In addition, paper [9] proposed reordering FFs in a scan chain to maximize switching power that may affect layout.

As they [9] took results from 65nm technology library and their layout design tool are different what we use, we cannot compare our results to them directly. But we simulate our method in 250MHz frequency for same number of regions and same Trojan T3 from [9]. Table 7 shows RPDs of two circuits (s35932 and s38417) with our proposed repartitioning method (LoC & LoC) and the other test patterns such as LoC & All 0s and Random & All 0s. Table 7 includes the patterns, region size, area overhead (AOH) and %RPD for all patterns individually. We compare the application of random and LoC pattern in one scan chain and all 0s in others (Random & All 0s, LoC & All 0s) compared to our proposed technique (LoC & LoC) for different region sizes. Results show that putting other scan chains 0s for both random and LoC patterns, the RPD is greater in LoC & All 0s compared to Random & All 0s. Our proposed pattern (LoC & LoC) shows maximum RPDs than any of two patterns which is reported 43.13% to 61.09% with the variation of region size from 32 to 64 for s35932 circuit.

5. Conclusion

This paper has presented an effective approach to detect malicious inserted Trojans. The proposed method partition a circuit into small segments and apply TDGPs that magnify a detection sensitivity. We gave an analysis that has shown a ratio RPD of power difference to a nominal power representing a detection sensitivity against a process variation, and demonstrated that the proposed method achieved high detection sensitivity for ISCAS89 benchmark circuits and some practical circuits. In future effort, we will fabricate IC while the technique will be applied in physical design, and capability of the technique will be evaluated in practice.

References


Fakir Sharif Hossain received the B. Eng. Degree in Electrical and Electronic Engineering from AUST, Bangladesh and M.S. Degrees in Information & Communication Technology from BUET, Bangladesh in 2007 and 2012 respectively. During 2009–2015, he was Assistant Prof. in International Islamic University Chittagong, Bangladesh. He currently studying Ph.D at Nara Institute of Science and Technology, Japan. His research interest is Hardware security and VLSI testing.

Tomokazu Yoneda received the B.E. degree in information systems engineering from Osaka University in 1998, and the M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology (NAIST) in 2001 and 2002, respectively. Presently he is an Assistant Professor of Graduate School of Information Science, NAIST. His research interests include VLSI CAD, design for testability, test pattern generation and built-in self-test. He is a senior member of the IEEE and a member of the IPSJ (the Information Processing Society of Japan).

Michiko Inoue received her B.E., M.E, and Ph.D degrees in computer science from Osaka University in 1987, 1989, and 1995 respectively. She worked at Fujitsu Laboratories Ltd. from 1989 to 1991. Currently, she is a Professor of Graduate School of Information Science, Nara institute of Science and Technology (NAIST), a Program Officer of Japan Society for the Promotion of Science (JSPS) Research Center for Science Systems and a member of Science Council of Japan. Her research interests include distributed algorithms, parallel algorithms, graph theory and design and test of digital systems. She is a senior member of IEEE, a member of the Institute of Electronics, Information and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), and Japanese Society for Artificial Intelligence.