TrojanZero: Switching Activity-Aware Design of Undetectable Hardware Trojans with Zero Power and Area Footprint

Imran Hafeez Abbassi*, Faiq Khalid†, Semeen Rehman†, Awais Mehmood Kamboh*,
Axel Jantsch†, Siddharth Garg‡ and Muhammad Shafique†
*School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan
Email: {imran.abbasi, awais.kamboh}@seecs.edu.pk
† Vienna University of Technology, Vienna, Austria
Email: {faiq.khalid, semeen.rehman, axel.jantsch, muhammad.shafique}@tuwien.ac.at
‡ New York University, New York City, U.S.
Email: sg175@nyu.edu

***Abstract*—Conventional Hardware Trojan (HT) detection techniques are based on the validation of integrated circuits to determine changes in their functionality, and on non-invasive side-channel analysis to identify the variations in their physical parameters. In particular, almost all the proposed side-channel power-based detection techniques presume that HTs are detectable because they only add gates to the original circuit with a noticeable increase in power consumption. *This paper demonstrates how undetectable HTs can be realized with zero impact on the power and area footprint of the original circuit.* Towards this, we propose a novel concept of *TrojanZero* and a systematic methodology for designing undetectable HTs in the circuits, which conceals their existence by gate-level modifications. The crux is to *salvage* the cost of the HT from the original circuit without being detected using standard testing techniques. Our methodology leverages the knowledge of transition probabilities of the circuit nodes to identify and safely remove expendable gates, and embeds malicious circuitry at the appropriate locations with zero power and area overheads when compared to the original circuit. We synthesize these designs and then embed in multiple ISCAS85 benchmarks using a 65nm technology library, and perform a comprehensive power and area characterization. Our experimental results demonstrate that the proposed TrojanZero designs are undetectable by the state-of-the-art power-based detection methods.**

***Index Terms*—Hardware Trojans, Power Analysis, Area, Signal Probability, ATPG**

I. Introduction and Related Work

The emerging complexity of modern embedded devices and associated cost of advanced CMOS fabrication have increased the trend of outsourcing integrated circuits (ICs) manufacturing processes to untrusted third-parties [1]. The IC supply chain comprises of various development stages that typically involve untrusted entities such as third-party IP vendors, electronic design automation (EDA) tools and fabrication foundries as shown in Fig. 1. Consequently, they are vulnerable to a wide range of HT attacks at some stage of the manufacturing process, which may lead to *leakage* of sensitive information to an adversary, *modification* in functionality, and *degraded performance* of integrated circuits [2].

A. HT Detection Techniques

To mitigate the potential threats of HT attacks in the supply chain, various HT detection techniques have been proposed. Typically, HT detection is performed at the design time (pre-silicon), or after manufacturing (post-silicon) depending on the un-trusted entity involved in the entire process as depicted in Fig. 1. These techniques are broadly classified into logic-based testing [3], and side-channel analysis [4].

a) ***Logic-based Detection****:* Logic-based detection includes equivalence checking [5], and exhaustive simulation [6] which provide 100% coverage. However, such techniques are not scalable, and applicable only to smaller circuits. Moreover, equivalence checking can only be deployed at the pre-silicon stage. Techniques based on automatic test pattern generation (ATPG) can be used at the post-silicon stage to generate a small set of test vectors that can excite HT trigger nodes [7]. Such techniques maximizes the probability of triggering HTs [8], however, the detection probability and coverage cannot be guaranteed to be 100%.

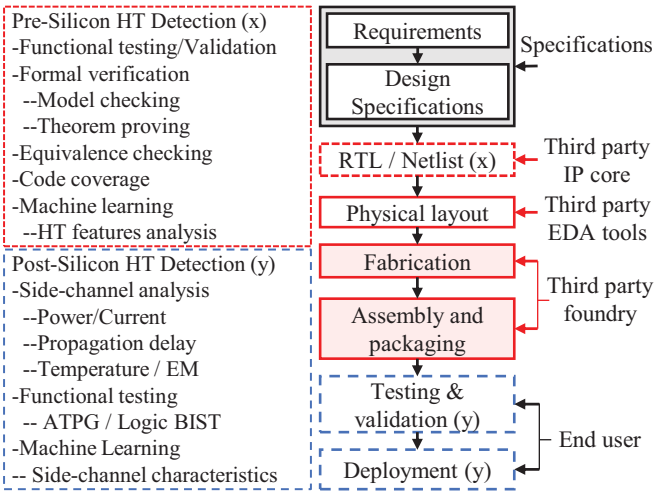

Fig. 1: IC supply chain stages susceptible to malicious insertions include: (a) Third party IP core (b) EDA tools (c) Untrusted foundries. Pre-silicon detection techniques are employed at the design stage, and post-silicon methods are used at testing and deployment phase.

b) ***Side-channel Analysis****:* The detection techniques based on analysis of side-channel measurements are premised on the assumption that HTs distort the parametric profile of the IC. This includes measuring the variations in the observable physical parameters, such as power, delay and temperature to detect any alteration in the IC [9]. Of these, the most commonly used techniques for HT detection are based on power analysis at the post-silicon stage. Some notable works include statistically analyzing the power traces through multiple ports [10] to identify HTs. Gate-level characterization using a set of power measurements is proposed in [11] to determine the increase in leakage power due to the addition of malicious gates. Similarly, a sparse gate profiling technique is proposed to detect increase in the leakage power of circuits using statistical learning [12].

In short, the above-discussed techniques primarily rely on the assumption that HTs are *additive*, i.e., the malicious circuitry is embedded by *adding* gates to the HT-free circuit, resulting in an increase in circuit area and power consumption. **The goal of this paper is to challenge these underlying assumptions, i.e., are HTs necessarily additive in terms of power and area?**

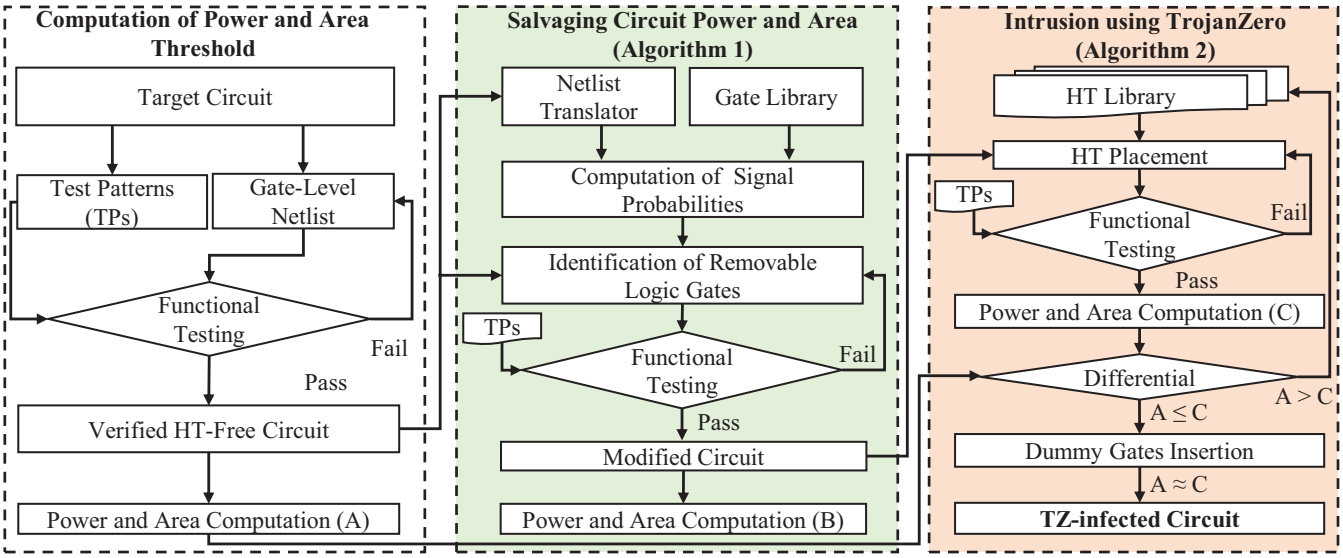

Fig. 2: Proposed flow for our *TrojanZero* Methodology: (a) HT-free circuit is functionally verified using defender's Test Patterns (TPs), and analyzed for computing power and area thresholds. (b) Signal probabilities of the nodes are computed and expendable gates are identified. (c) An HT (i.e., a TrojanZero instance) is embedded in the modified circuit with zero power and area overheads.

B. Motivational Analysis and Research Challenges

Our analysis in Fig. 3 shows the percentage overhead in terms of power and area that is assumed by some of the state-of-the-art methodologies [10]–[12] for successful detection of a single HT in ISCAS benchmark c499. Following key observations are made from this experimental analysis:

1) Dynamic and leakage power of the HT-infected circuit are perceptibly increased when compared to that of HT-free circuit for successful detection. For instance, the dynamic power of the HT-infected circuit is assumed to exceed at least by 0.265% as depicted by observation point X. Similarly, the percentage increase assumed for leakage power-based HT detection is shown by Y1 and Y2.
2) The area of the circuit is assumed to increase due to presence of an HT. For instance, points A1, A2 and A3 show an increase in the area by 0.7%, 1.95% and 0.58% compared to that of HT-free circuit.

The above-discussed defence techniques appear to work if the increments in power and area are discernible, hence we are going to refrain these additive effects such that HTs are undetected by the existing state-of-the-art methodologies. The open question that is not addressed in the literature is: *how to modify a given circuit in order to insert an HT such that its total power consumption and area are equal to that of a HT-free circuit. We refer to this as devising a* ***TrojanZero*** *methodology with zero power and area overheads.*

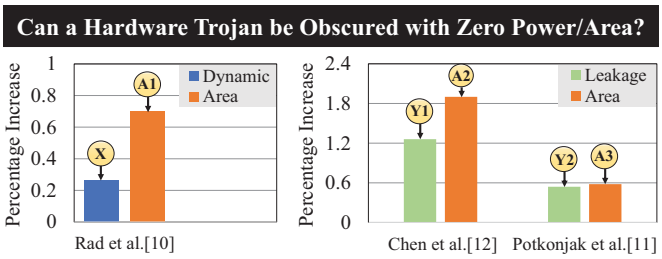

Fig. 3: Minimum power and area overheads that are assumed by state-of-the-art techniques [10]–[12] for successful detection of a single HT in ISCAS benchmark c499.

C. Attack Model

In the proposed attack model, we assume that the attacker resides at the foundry where she can modify the circuit in the form of addition, deletion or modification of the gates during fabrication [1]. Moreover, the attacker acquires the knowledge of specific testing techniques that are used by the defender for the functional validation of circuit after fabrication. The attacker seeks to modify the circuit such that:

1) It deviates from the original functionality for certain inputs.
2) Functionality is not altered on the defender's testing inputs.
3) Power and area are not increased and HTs are undetectable by the power-based post fabrication analysis.

Prior works have identified two types of HTs: (i) *untargeted* HTs only seek to arbitrarily modify the circuit behavior for certain inputs; and (ii) *targeted* HTs modify the circuit behavior for attacker chosen inputs in an attacker-specified manner. While the primary goal of the *TrojanZero* is to introduce targeted HTs, we will notice that it necessitates the introduction of additional untargeted HTs as well.

D. Our Novel Contributions

In this work, we propose a novel concept of *TrojanZero* along with an HT insertion methodology that may subvert the normal operation of ICs, and has **no additional overheads** in terms of power and area. Our primary contributions in a nutshell are:

1) Devising a scheme to identify rarely-activated nodes in the circuit with extremely low signal probabilities.
2) An algorithm to explore the space of circuit modifications that leave the circuit's functionality on the defender's test patterns unchanged.
3) A methodology to embed HTs in the target circuit such that there is no increase in total power along with its components, and area.
4) Implementation of an HT with a low triggering probability ($< 10^{-4}$) during the functional testing phase.

II. Methodology to Design and Implement TrojanZero

The high-level flow of our proposed TrojanZero methodology is shown in Fig. 2. It comprises of three main steps as depicted, and explained in the subsequent sections.

A. Computation of Power and Area Thresholds

In the first phase, thresholds of the HT-free circuit are computed in terms of power, its components, and area footprint.

1) **Generating Test Patterns and Functional Verification:** A set of defender-specific test patterns is generated for the HT-free circuit through logic-based testing techniques. Functional verification of the circuit is performed by monitoring the outputs against the test patterns (TPs) through simulations. The circuit is re-verified in case of functional failure.

2) **Power and Area Analysis:** The verified HT-free circuit is synthesized using the technology library while optimizing it for minimum power. This follows computation of the total power and its constituents, i.e., dynamic and leakage power. The area of the circuit is computed in terms of number of gate equivalents (GE) with respect to the technology after synthesis using the ASIC design tools. This analysis of the HT-free circuit is used to specify the power and area thresholds that are to be strictly adhered while embedding HTs.

B. Salvaging Circuit Power and Area

We propose Algorithm 1 to salvage power and area from the HT-free circuit with *n* nodes. It executes as follows:

1) **Inputs:** HT-free circuit (N), a set of q defender's functional testing algorithms with generated TPs, along with the reference power (P(N)), and area (A(N)) computed during analysis.

2) **Computation of Switching Probabilities:** The circuit is translated into its corresponding model using a netlist translator, which computes the signal probabilities at each node for being at logic 0 or 1 as depicted in Lines 2, and 3. For this, we develop a library comprising of basic and complex gates. Each gate computes the probabilities ($P_{g=0}$, $P_{g=1}$) at its output node based on the probabilities of signals at its inputs. *Similar to other approaches in this field, we also assume that the signal probability at each primary input is 0.5.*

3) **Identification of Expendable Gates:** Based on the attacker-specified probability threshold (P_{th}), a list of candidate gates (C) is obtained that comprises of circuit nodes with signal probabilities close to zero or one as shown in Line 8. Each candidate gate is checked for its possible removal from the circuit after detailed functional testing. For this, each node from C with a signal probability close to one is removed and connected to logic 1. Similarly, the output node of gate with signal probability close to zero is connected to logic 0 as depicted in Line 11. After removing a single element from C, each of the previous gate is eliminated safely if its output is not connected to any other node of the circuit. This follows detailed functional testing using the defender's q validation algorithms with all TPs. If functional tests are successful then the gates under consideration are successfully removed. However, if any of the test fails, the changes made in the circuit are reverted and next gate from C is tested as depicted in Line 18. This procedure would ensure that such a change in the circuit would go undetected by the defender's post-silicon test techniques.

4) **Power and Area Differential Gains:** The removal of identified gates from C follows the computation of power and area of the modified circuit (N') as shown in Line 23. Moreover, the differential with respect to the HT-free circuit (N) is computed to determine the salvaged cost in terms of total power, its components and area.

C. HT Insertion using TrojanZero Methodology

We propose Algorithm 2 to insert an HT with zero power and area (TZ), employing our *TrojanZero* methodology. The algorithm provides a procedure for an attacker to systematically exploit the salvaged cost of the modified circuit. This not only makes TZ extremely hard for the defender to be triggered even with bespoke functional test patterns, but renders its existence undetectable from the power and area based analysis techniques.

Algorithm 1 : Salvaging Power and Area

Input:
$N = \{N_1, N_2, ..., N_n\}$: Verified HT-free circuit with *n* nodes
$T = \{T_1, T_2, ..., T_q\}$: q testing algorithms of defender
P_{th} = Attacker specified threshold probability
P(N) = Power of N, A(N) = Area of N
Output:
$N' = \{N'_1, N'_2, ..., N'_t\}$: Modified circuit with *t* nodes
P(N') = Power of N', A(N') = Area of N'
Goal:
ΔP = P(N) - P(N'), and ΔA = A(N) - A(N')
Initialize: i = 1, j = 1, k = 1, m = 1, s = 1, X = {}, Y = {};

```
1:  while i ≤ n do
2:      P(N_i = 0) = P_g=0, P_g=0 ∈ {P_(NAND=0), ..., P_(OR=0)};
3:      P(N_i = 1) = P_g=1, P_g=1 ∈ {P_(NAND=1), ..., P_(OR=1)};
4:      if P(N_i = 0) ≥ P_th then X = X ∪ N_i; j = j + 1;
5:      end if
6:      if P(N_i = 1) ≥ P_th then Y = Y ∪ N_i; k = k + 1;
7:      end if
8:      C = X ∪ Y; i = i + 1;                    ▷ Candidate nodes (C)
9:  end while
10: while m ≤ j + k do                  ▷ Testing each C_m for removal
11:     if C_m ∈ X then C_m = 0;                  ▷ Replace node with 0
12:         Remove preceding gates and update circuit to N';
13:     else if C_m ∈ Y then C_m = 1;             ▷ Replace node with 1
14:         Remove preceding gates and update circuit to N';
15:         while s ≤ q do          ▷ Testing N' using algorithms 1 to q
16:             if T_s == Pass then s = s + 1;                ▷ ∀ TPs
17:             else
18:                 Revert changes in circuit N'; s = q;
19:             end if
20:         end while
21:     end if  m = m + 1;
22: end while
23: Compute P(N'), A(N'), ΔP, and ΔA;
```

1) **HT Placement:** To subvert the desired operation of the circuit, HT from the library of *n* existing malicious circuits is selected and carefully inserted within N'. The payload of the HT can be triggered by the attacker-chosen set of vectors provided either by internal or external means. After placement of an HT with an imperceptible trigger for which *m* locations are available for insertion, functional testing is performed with q algorithms of the defender. If the test fails, the HT is placed at the next target location as shown in line 6.

2) **Power and Area Analysis:** After successful functional testing on defender algorithms, power (P(N'')) and area (A(N'')) of the TZ-infected circuit (N'') are analyzed to ascertain that it does not surpass the defined thresholds. Conversely, if this is not *true*, then the entire process of HT insertion is repeated by selecting another HT from the library. This follows insertion of the dummy logic gates (if required) to meet the baseline conditions for successful insertion of TZ. These conditions assert that power consumption and area are equivalent to the prescribed thresholds, i.e., ΔP(TZ) = 0, and ΔA(TZ) = 0, as depicted in Line 12. It is mandatory to analyze individual components of power, i.e., dynamic and leakage, independently. These components vary depending upon the location and configuration of HT gates within circuit. *It is plausible that one of the components surpasses the defined threshold, while total power consumption remains within the specified constraints.*

The insertion of TZ in the modified circuit using Algorithm 2 is undetectable to the post-fabrication tests performed by the defender.

Algorithm 2 : HT insertion using TrojanZero methodology

Input:
$N' = \{N'_1, N'_2, ..., N'_t\}$: Modified circuit with t nodes
$T = \{T_1, T_2, ..., T_q\}$: Set of q functional testing algorithms
$HT \in \{\{HT_1, HT_2, ..., HT_n\}$: Library with n HTs
$l \in \{\{l_1, l_2, ..., l_m\}$: m potential location in N'
P(N) = Power of HT-free circuit N
A(N) = Area of HT-free circuit N
P(N') = Power of modified circuit N'
A(N') = Area of modified circuit N'
Output:
N'' = TZ-infected circuit
P(N'') = Power of N'', A(N'') = Area of N''
Goal:
ΔP(TZ) = P(N) - (P(HT) + P(N')) = 0
ΔA(TZ) = A(N) - (A(HT) + A(N')) = 0
Initialize: i = 1, j = 1, s = 1;

```
1:  while i ≤ n do
2:      while j ≤ m do
3:          while s ≤ q do
4:              Place HT_i at location l_j;
5:              if T_s == Pass then s = s + 1;        ▷ Test with next Algo
6:              else j = j + 1; goto 2;               ▷ Place HT at next location
7:              end if
8:          end while j = m;
9:      end while
10:     Compute P(N'') = P(HT) + P(N');
11:     Compute A(N'') = A(HT) + A(N');
12:     if ΔP(TZ) = 0 && ΔA(TZ) = 0 then
13:         HT with zero power and area successfully inserted; i = n;
14:     else i = i + 1;
15:     end if
16: end while
```

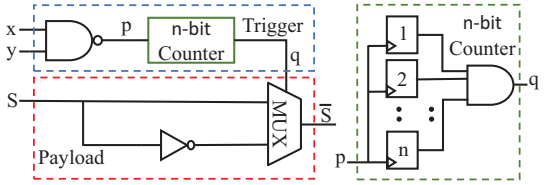

Fig. 4: Asynchronous Counter-based HT [14]

Therefore, it introduces, (i) targeted *explicit* behavior to modify functionality on attacker specified vectors [13]; (ii) targeted *implicit* behavior while evading side-channel analysis, i.e., power and area.

III. Case Study: Intruding 8-bit ALU using TrojanZero

To demonstrate the practicability of the proposed approach, we apply the concepts of *TrojanZero* on ISCAS benchmark c880 [15].

A. Computation of Power and Area

We assume that the defender performs functional testing using ATPG stuck-at model. This model is prominently used for diagnostic test generation of transition faults after fabrication, where the nets of circuits are assumed to be stuck at a fixed logic value [16]. TPs generated from the test algorithm are used to ascertain the functionality of c880. The total power consumption and area footprint in terms of gate equivalents (GE) is determined as 77.2 μW, and 365.4 GE, respectively. Similarly, the cell level dynamic and leakage power components are 70.35 μW, and 6.87 μW, respectively.

B. Maximizing the Differential of Power and Area

We apply Algorithm 1 to salvage the cost in terms of power and area by identifying expendable gates. For each gate of c880, we compute signal probabilities at its output node. We compute the set of candidate gates (C) by specifying P_{th} = 0.992. Choosing high value of P_{th} provides less number of candidates, however, it increases the ratio of the gates that can be removed from the identified candidates. Moreover, the probability of detecting modifications on defender's bespoke vectors decreases with the higher value of selected P_{th}. The set C comprises 27 gate whose signal probabilities are above P_{th} or below $1 - P_{th}$. Fig. 5 shows two segments of c880 comprising of nodes in set C. The set C includes the four AND gates in segment A, i.e., N476, N478, N480 and N482, and the four OR gates highlighted in segment B. Now, we first remove AND (N476) and connect the corresponding input of NOR (N503) to logic 0. This is followed by checking whether any additional independent gates can be removed. However, there is no independent removable gate as AND (N432) is interconnected with adjacent AND gates, and NOT (N310) is connected with other nodes. Next, functional testing is performed and Algorithm 1 determines that this node can indeed be removed from set C. Similarly, all other candidate gates are removed iteratively. If, at any step, functional testing fails, the modified circuit is reverted to the previous step. After iterating over all gates in C, we test that AND (432) can be expended safely, since all the gates driven by this node have already been removed. All such preceding gates are checked for their secure removal. Similarly, the expendable gates in segment B of c880 are highlighted in Fig. 5. It is observed from the application of Algorithm 1 on segments A and B that the gate driving node N287 can also be expended. After detailed analysis, we successfully salvaged the cost of 11 logic gates from c880. We compute the power and area of the modified circuit to be 70.2 μW, and 329.7 GE. This gives us a differential of 7 μW in power, and 35.7 GE in area footprint that we can used to embed HT.

C. TrojanZero Implementation

We execute Algorithm 2 on the modified circuit to insert an asynchronous counter-based HT [14] as shown in Fig. 4. This HT modifies the signal *S*, whenever the select input *q* of the multiplexer is set to logic 1 by the counter. We placed this HT to modify carry-in (N261) of the c880 ALU on a trigger signal from the counter. The inputs to generate the trigger are provided from rarely-activated nodes of the circuit such that it is not activated during the defender's functional testing. With the insertion of 3-bit counter for trigger generation, it is observed that the total power consumed by the TZ-infected circuit is 76.4 μW. Moreover, dynamic and leakage power components are 69.32 μW. and 6.85 μW, respectively. Similarly, the cell area of the TZ-infected c880 has a footprint of 362.8 GE. Therefore, the outputs of Algorithm 2 in terms of *TrojanZero* parameters are: ΔP_T(TZ) = 0.8 μW, ΔA(TZ) = 2.6 GE, ΔP_D(TZ) = 1.03 μW, ΔP_L(TZ) = 0.02 μW, where T, D, and L represents total, dynamic and leakage power. The outputs show that TZ-infected circuit has almost equal power and area compared to the HT-free circuit. Therefore, this counter-based HT will not be detected with the state-of-the-art power analysis based HT detection.

IV. Results and Discussions

We evaluate the proposed methodology on a set of ISCAS85 benchmarks. The experimental setup used for the design and implementation of *TrojanZero* is depicted in Fig. 6. All the simulations are executed on the Red Hat Enterprise Linux (RHEL) 6.8 based computing machine with 8-Core processor @ 2.4 GHz, and 16 GB memory, and using the following tools:

1) *Synopsys TetraMAX (2016)* for automated test generation for the circuits. The circuit is given as input to the tool, which generate the test patterns using the stuck-at model.
2) *Matlab 9.1 (R2016b)* based program to compute node probabilities of the HT-free circuit.

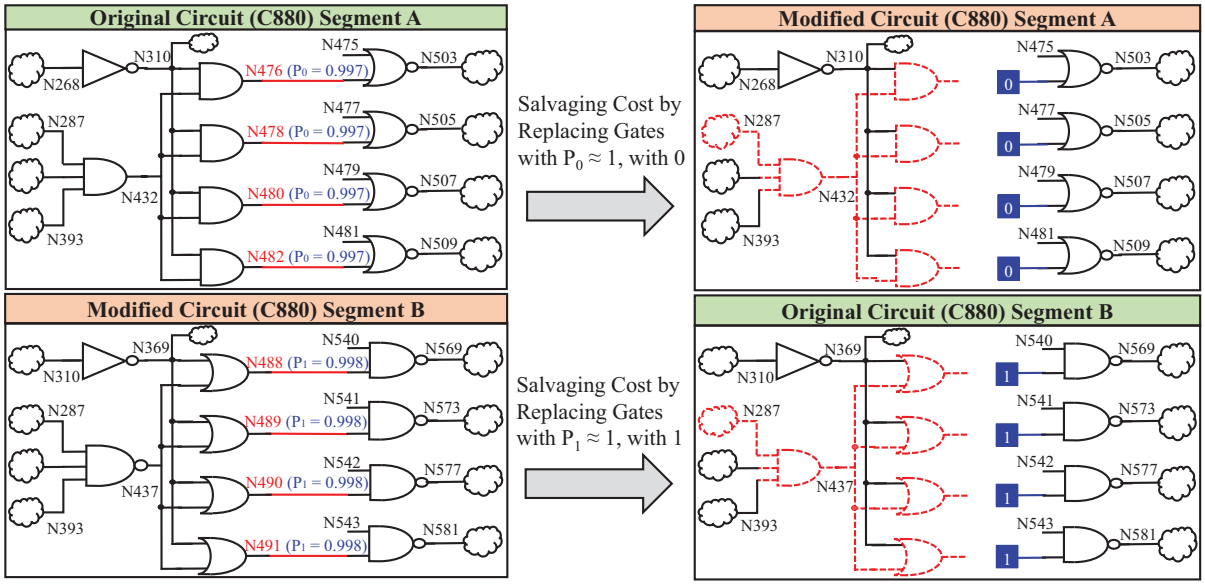

Fig. 5: Expending candidate gates to salvage cost for power and area in ISCAS c880 8-bit ALU.

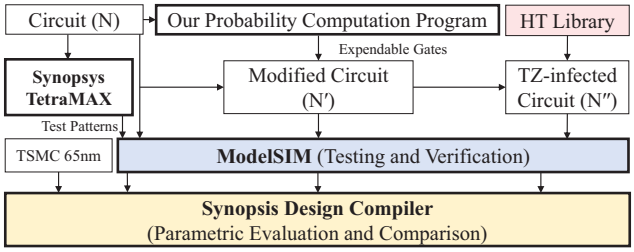

Fig. 6: Tool flow to compute thresholds, salvaging cost by decimating components, and HT insertion.

3) *Mentor Graphics ModelSim 10.5a* for circuit verification during each phase of *TrojanZero* implementation.
4) *Synopsys Design Compiler (2016)* for synthesizing each circuit using 65nm TSMC technology to perform the detailed analysis based on power and area.

Table I depicts the P_{th}, number of elements in the set C, expendable gates (E_g), and the estimated probability (P_{ft}) for activating HT using random functional testing. The summary of power and area analysis of the HT-free (N), modified (N'), and TZ-infected (N'') circuits is given for comparison. Based on our experimental results, following observations are made:

1) **Dynamic, and Leakage Power Analysis:** The power consumption of the circuit along with its components vary subject to the size and location of the inserted HT. This is due to the variability in transition probabilities, and distinct configuration of the different parts of circuit. Fig. 7 presents the comparison of N, N' and N'' in terms of leakage and dynamic power consumption. It is observed that the leakage power is more liable to violate the boundary of defined constraints compared to the other parameters as depicted by X. Therefore, size of the inserted HT is mainly dictated by its leakage power. With the modifications in the circuit and subsequent insertion of an HT, there is a likelihood that the dynamic power decreases, the leakage power increases, while the total power remains within the defined threshold. This observation stems from the fact that the HT gates leak static power, even when the HT is not triggered. However, the dynamic power consumption of the N'' is typically below the defined constraint as depicted by Y in Fig. 7. Therefore, leakage of the circuit is required to be precisely monitored in all phases.
2) **Circuit Configuration and Salvaging Cost:** The admissible size of the HT is proportional to the configuration, complexity, and the salvaged cost of the circuit. There is a potential of inserting multiple HTs of variable sizes in large complex circuit. Typically, complex circuits are likely to have more expendable gates. This is due to the presence of comparatively large number of such nodes that have extremely low probability of transitioning. It is shown in the Table I that the benchmark c1908 comprises of almost half the number of gates compared to that for the c3540, but have 45 expendable gates compared to 57. The configuration of circuit allows for the selection of higher P_{th} and correspondingly more cost is salvaged.
3) **Area of TZ-infected Circuit:** The observation Z in Fig. 7 shows that there may be a case, where an area cap is required to be adhered more strictly compared to other parameters. Table I shows that at the cost of 45 expendable gates, insertion of 5-bit counter HT in c1908 will have a the margin of 0.2% from the area cap. However, the same HT has relatively higher margins for dynamic (4.75%), and leakage (0.8%) power. Therefore, increasing the size of HT will first violate the area constraint instead of leakage power.
4) **Rare-states Combination for HT Trigger:** Choosing the nets with low transition probabilities gives a substantial resistance for triggering an HT on defender's random test vectors as depicted by P_{ft}. However, an HT can be triggered on attacker-chosen vectors.

The baseline condition for the successful implementation of the proposed attack is to ensure that power, its components and area $\approx$ 0. In some cases, HT-insertion in the modified circuit may result into negative differential, i.e., discernible decrease of power and area compared to the HT-free circuit. In such cases, dummy gates maybe inserted in parallel to the primary inputs with their outputs unconnected, and thus acquiring negligible differential for all parameters.

The TrojanZero methodology relies on the condition that attacker acquires a substantial knowledge pertaining to functional testing techniques of the defender. This scenario is conceivable, since the increasing complexity of system-on-a-chip (SoC) integration has raised the tendency of outsourcing IC testing services to the third-party vendors. This provides an attacker with a realistic opportunity to obtain relevant information from the third-party. Moreover,

TABLE I: TrojanZero Analysis for ISCAS85 Benchmarks (I/P = Inputs, P_{th} = Threshold Probability, C = Candidate Gates, E_g = Expendable Gates, N = HT-free circuit, N' = Modified circuit, N'' = TZ-infected circuit)

Circuit	Gates	I/P	P_{th}	C	E_g	HT (Counter)	Total Power (μW)			Area			P_{ft}
							N	N'	N''	N	N'	N''	
c432	160	32	0.975	8	5	2-Bit	35.6	20.83	27.7	186.8	136	163	$0.9 \cdot 10^{-4}$
c499	202	41	0.993	12	7	3-Bit	181.9	173.4	177.4	463.4	396.4	451.5	$6.1 \cdot 10^{-6}$
c880	383	60	0.992	27	11	3-Bit	77.2	70.2	76.4	365.4	329.7	362.8	$8.0 \cdot 10^{-6}$
c1908	880	33	0.9986	43	45	5-Bit	160.9	151.6	157.4	454.7	446.4	453.6	$6.1 \cdot 10^{-8}$
c3540	1669	50	0.992	41	57	5-Bit	248.5	187.2	241.7	986.8	944.3	980	$2.0 \cdot 10^{-6}$

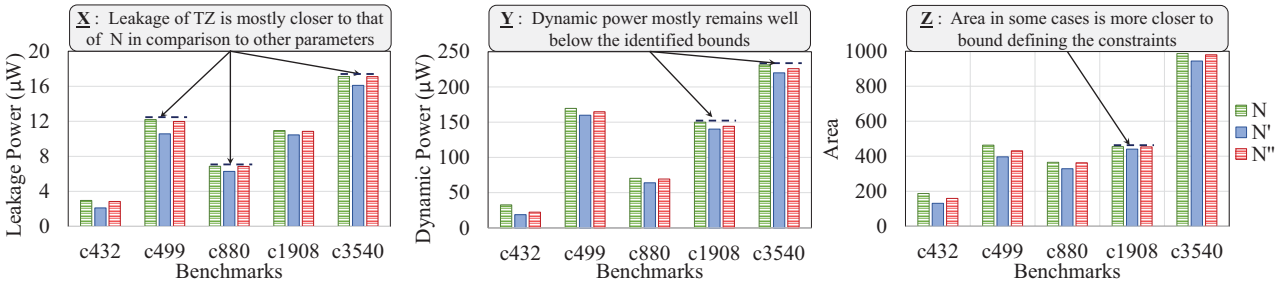

Fig. 7: Comparison of HT-free (N), modified (N') and TZ-infected (N'') benchmark circuits in terms of area, leakage and dynamic power.

the design-for-testability (DfT) techniques e.g., scan-based testing structures provide a reasonable insight to the attacker residing at the foundry about the testing structures employed by the end-user [17].

Apart from the conventional functional testing techniques, the defender may use a set of random (bespoke) vectors for validation which are not known to the attacker. The probability of triggering the targeted HTs using these vectors is very extremely low, as shown by P_{ft} in Table I. Moreover, the probability to reveal un-targeted HTs by the random vectors is determined as follows:

$$P_u = \frac{N_u}{2^n} \tag{1}$$

where, P_u is the probability to trigger an un-targeted HT, N_u represents the number of random input combinations that triggers the untargeted HT, and n is the total number of circuit inputs.

The experimental results and discussion advocate our claims that devising an HT using *TrojanZero* is a pragmatic approach, which can reasonably circumvent the existing state-of-the-art power-based HT detection techniques, and thereby requiring the investigation for new detection methodologies.

V. Conclusion

We proposed a novel concept of *TrojanZero* to design and embed undetectable HTs in the target circuit with absolutely no additional costs in terms of power and area. Our method leverages the knowledge of circuit configuration to secure meaningful cost in terms of power and area by decimating its redundant components. We used the salvaged resources to embed HTs in the circuit while adhering with the power, and area cap provided by the analysis of a given HT-free circuit. Our experimental results show that *TrojanZero* can successfully evade the available state-of-the-art HT detection techniques which have the baseline premise that HT insertion eventuates into notable increase of power and size of the circuit. Our methodology provides a foundation for devising new stealthier attacks. An attacker with a reasonable knowledge of circuit configuration can circumvent its security with potentially no risk of getting detected. This instigates a need of exploring more sophisticated and viable techniques for the post-silicon detection of HTs.

References

[1] K. Xiao, D. Forte, Y. Jin, R. Karri, and M. Tehranipoor, "Hardware Trojans: Lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, 2016.

[2] S. Bhasin and F. Regazzoni, "A survey on Hardware Trojan detection techniques," in *Circuits and Systems*. IEEE, 2015, pp. 2021–2024.

[3] S. Saha, R. S. Chakraborty, S. S. Nuthakki, D. Mukhopadhyay *et al.*, "Improved test pattern generation for HT detection using genetic algorithm and boolean satisfiability," in *Cryptographic Hardware and Embedded Systems*, ser. LNCS, vol. 9293. Springer, 2015, pp. 577–596.

[4] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. G. Wolff, C. A. Papachristou, K. Roy, and S. Bhunia, "Hardware Trojan detection by multiple-parameter side-channel analysis," *IEEE Transactions on computers*, vol. 62, no. 11, pp. 2183–2195, 2013.

[5] X. Zhang and M. Tehranipoor, "Detecting Hardware Trojans in Third-Party Digital IP Cores," in *Hardware-Oriented Security and Trust (HOST)*. IEEE, 2011, pp. 67–70.

[6] A. Bazzazi, M. T. M. Shalmani, and A. M. A. Hemmatyar, "Hardware Trojan detection based on logical testing," *Journal of Electronic Testing*, vol. 33, no. 4, pp. 381–395, 2017.

[7] V. Govindan and R. S. Chakraborty, "Logic testing for Hardware Trojan detection," in *The Hardware Trojan War*. Springer, 2018, pp. 149–182.

[8] R. S. Chakraborty, F. G. Wolff, S. Paul, C. A. Papachristou, and S. Bhunia, "MERO: A statistical approach for Hardware Trojan detection." in *Cryptographic Hardware and Embedded Systems (CHES)*, ser. LNCS, vol. 5747. Springer, 2009, pp. 396–410.

[9] G. Di Natale and Dupuis, "Is side-channel analysis really reliable for detecting Hardware Trojans?" in *Design of Circuits and Integrated Systems, (DCIS)*, 2012, pp. 238–242.

[10] R. Rad, J. Plusquellic, and M. Tehranipoor, "A sensitivity analysis of power signal methods for detecting Hardware Trojans under real process and environmental conditions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 12, pp. 1735–1744, 2010.

[11] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan horse detection using gate-level characterization," in *Design Automation Conference (DAC)*. IEEE, 2009, pp. 688–693.

[12] X. Chen, L. Wang, Y. Wang, Y. Liu, and H. Yang, "A general framework for hardware trojan detection in digital circuits by statistical learning algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 10, pp. 1633–1646, 2017.

[13] S. K. Haider, C. Jin, M. Ahmad, D. Shila, O. Khan, and M. van Dijk, "Advancing the state-of-the-art in Hardware Trojans detection," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[14] H. Liu, H. Luo, and L. Wang, "Design of Hardware Trojan Horse Based on Counter," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011*. IEEE, 2011, pp. 1007–1009.

[15] "ISCAS85:," http://web.eecs.umich.edu/ jhayes/iscas.restore/, 2018.

[16] M. Bushnell and V. Agrawal, *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Springer Science & Business Media, 2004, vol. 17.

[17] M. Yasin, O. Sinanoglu, and J. Rajendran, "Testing the trustworthiness of ic testing: An oracle-less attack on ic camouflaging," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 2668–2682, 2017.