Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Dual-Metric Detection of Synthesis-Masked Hardware Trojans in AES-256: Correlating Power Signatures with Switching Activity

Rahimunnisa K.[1], Aadhitya G.[2], Abhyjeet J.[3], Dharnesh S.[4]

Department of Electronics and Communication Engineering, Easwari Engineering College, Anna University, Chennai, India.

E-mail: [1]rahimunnisa.k@eec.srmrmp.edu.in, [2]310622106001@eec.srmrmp.edu.in, [3]310622106005@eec.srmrmp.edu.in, [4]310622106029@eec.srmrmp.edu.in

Abstract

Hardware Trojans can be accidentally hidden due to synthesis optimizations performed by electronic design automation tools in cryptosystems, leaving important detection gaps in the process. In this study, the impact of synthesis on the power signatures of AES-256 cores has been analyzed based on power/net switching correlations. Four hardware Trojans have been synthesized targeting the 7nm Versal technology of Xilinx and have been simulated over the course of 10,000 clock cycles across 714 different key/plaintext combinations. The results showed that three hardware Trojans exhibited unexpected negative power deviations, although they incorporated malicious logic in their structures. Two separate categories of synthesis interactions can be observed from switching distribution histograms over 646 nets and KL divergence statistics (0.03–0.42): synthesis-masked dormant Trojans with lower toggles and leftward-shifting distributions, and synthesis-neutralized active Trojans with 40% more active nets and lower power. This two-fold correlation method allows for detecting synthesis-affected Trojans that cannot be identified through conventional power-overhead analysis methods, achieving reproducible structural detection supported by dual-metric cross-validation (Z-scores -0.62 to -0.77), despite individual signatures falling below the conventional 3σ detection threshold.

Keywords: Hardware Trojans, AES-256, Power Analysis, Synthesis Masking, Negative Power Signature, Switching Activity Analysis.

1. Introduction

Hardware Trojans present a real threat to cryptography due to their potential ability to leak AES encryption keys, manipulate authentication schemes, or initiate denial-of-service attacks [1]. With up to 65% of IC fabrication now outsourced worldwide [2], the number of attack surfaces has increased immensely. AES-256 is a key element of defense networks, finance, and secure communication infrastructures [3], where compromise would be catastrophic.

Conventional Trojan detection is based on the principle that “the presence of inserted malicious logic results in higher power consumption” [4]. However, this assertion is no longer valid in today’s synthesis environment. Electronic Design Automation tools apply stringent

DOI: https://doi.org/10.36548/jtcsst.2026.2.003
Received: 19.02.2026, received in revised form: 22.03.2026, accepted: 10.04.2026, published: 24.04.2026

optimization techniques such as logic thinning, gate collapsing, and elimination of redundancies to satisfy timing and area requirements [5]. However, interaction between synthesis and Trojan hardware is not simply an issue of masking; when executed, synthesis cannot eliminate the logic without causing functional errors, but it still optimizes around the added logic [1].

Thus, two different classes of failure cases emerge: an infected core consumes lower energy than the uninfected core in its inactive state (masking the infection during synthesis), it executes a successful Trojan but fails to produce the desired increase in power consumption (neutralizing the infection during synthesis). This study explores the relationship between synthesis optimization and hardware Trojans in AES-256 cores. The main objective of this research is to determine whether synthesis affects the power signature of a Trojan in such a way that it becomes undetectable. Can a negative deviation in power consumption, even when unexpected in some cases, be used to reliably detect the presence of Trojans? Which specific patterns of switching activities at the net level indicate the synthesis masking and synthesis neutralizing effect? These questions are answered by investigating four different types of Trojans using vector-based power analysis and detailed switching activity distributions [5].

The key discovery lies in identifying and describing two synthesis-based effects wherein optimization of the EDA tool leads to negative power signatures of the Trojan-infested AES cores. The synthesis-masking effect involves the optimization of Trojans in a sleeping state together with surrounding circuitry, resulting in reduced power and switching activities. The synthesis-neutralization effect refers to the case when the Trojan circuit behaves according to its specifications but, due to the optimization process, fails to perform its designed purpose. These two cases directly contradict the conventional assumptions about detecting Trojans. Based on these observations, a novel classification scheme using power and switching distribution metrics has been proposed.

2. Related Works

A systematic review of hardware Trojans was published by Tehranipoor and Koushanfar [5]. The authors classified Trojan attacks based on physical properties, triggering mechanisms, and consequences of the payload action. This review laid down the groundwork for considering that power analysis is the only feasible technique to perform detection in a non-invasive way, and defined overhead detection as a prevailing strategy. Based on the idea that malicious hardware insertion leads to higher power consumption, subsequent side-channel detection architectures were designed.

Xiao et al. [6] presented a ten-year literature review on hardware Trojans, where the authors explored the development of insertion techniques, Trojan detection techniques, and hardware Trojan protection schemes. The survey clearly showed that functional testing and structural testing cannot reliably identify stealthy Trojans triggered by rarely occurring events, but side channel attacks remain the main tool for post-manufacturing Trojan detection. Most importantly, Xiao et al. [6] recognized in their survey the difficulty of analyzing Trojan circuits that interact with synthesis optimization, posing it as an open issue.

Hoang [7] showed how side-channel power traces can be used with supervised machine learning classifiers to detect whether an AES implementation was Trojan-infected or not, with a certain degree of classification accuracy. He found that the ability to detect becomes impossible when the Trojan logic overhead decreases by about 0.3% of the power consumed,

setting up a lower detection bound in power trace measurements. This constraint applies to the synthesis-masked and synthesis-neutralized Trojans studied here, whose power deviations fall below this bound.

Tehranipoor et al. [8] experimentally derived the requirements for the signal-to-noise ratios necessary for detecting power-based and delay-based Trojans. They showed that statistically valid detection requires either larger Trojan overheads than what can be measured above a certain noise level or the averaging of measurement results for many test vectors. They provided methods for signal calibration that enable detection under process variability and noise levels. The baseline method described herein for detecting Trojans with Z-scores was developed based on the aforementioned theory.

Amornpaisanon et al. [9] designed lightweight analytical models to detect secure runtime hardware Trojans with sufficient capability to be executed in the resource-constrained environment of FPGAs. This system uses statistical deviation in conjunction with an analytically calculated power model and is able to detect such Trojans without post-manufacturing destruction or the need for additional equipment to perform tests. However, the effectiveness of their solution lies in their inability to identify hardware Trojans that do not have detectable positive power anomalies.

The issue of the interactions of EDA techniques on hardware Trojan detectability has gained relatively little attention. Dong et al. [10] explored various methods and trends in the detection and prevention of hardware Trojans by surveying state-of-the-art synthesis processes.

In their study, Lohith et al. [11] analyzed AES designs under an area constraint-based synthesis process where it was seen that logic pruning can inhibit Trojan side channel attacks because of their treatment as noncritical optimizations for dormant malicious paths. These two distinct phenomena were neither quantified nor characterized in their studies as done in our work.

In terms of specialized AES hardware implementation security, Rahimunnisa et al. [12] have implemented an AES-256 architecture on FPGA with a high-throughput folded parallel design, offering an ideal experimental benchmark for comparison-based power analysis experiments.

Regarding side-channel analysis techniques with the assistance of machine learning algorithms, the work done by Zhou et al. [13] includes an AI-based method for detecting hardware Trojans through the use of power traces as input for machine learning models, showing the feasibility of using neural networks to classify hardware Trojans through power trace features with high detection rates. This work relies on the availability of labelled training data representing the class of target Trojans and does not generalize to different synthesis-induced behaviors of Trojans.

The work done by Golabi [14] involves improving classification performance through the use of dual channels, combining two types of side-channel information: power and structure. This technique uses machine learning algorithms in a scenario where no training data exists, which complements the technique used in this paper.

The gap that this body of work collectively leaves open is the absence of a detection framework capable of classifying Trojans whose power signatures are suppressed or inverted by synthesis optimization Trojans that are invisible to every overhead-based method reviewed

above. Table 1 summarizes the representative detection approaches discussed and positions the proposed dual-metric framework against them across key methodological dimensions.

Table 1. Comparison with Other HT Detection Methods

Reference	Method Type	Detection Approach	Golden Reference
[5]	Power Side-Channel	Comprehensive survey establishing overhead-based detection paradigm	Required
[8]	Power Side-Channel	Signal-to-noise ratio calibration for power-based Trojan detection	Required
[7]	Machine Learning	Supervised ML classifier trained on power trace features	Required
[9]	Analytical Modeling	Lightweight runtime statistical deviation analysis against derived baseline	Required
[13]	AI-Enabled Side-Channel	Neural network classifier on power trace features across diverse Trojans	Not required

3. Proposed Work

This section introduces the Advanced Encryption Standard (AES-256) architecture and then shows a taxonomy of Hardware Trojans that motivates the paper's side-channel analysis approach.

3.1 AES-256 Architecture

Figure 1 depicts the generic AES-256 encryption architecture with its 14-round structure, which includes SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations executing iteratively. This structure enables consistent power profiling across encryption cycles.

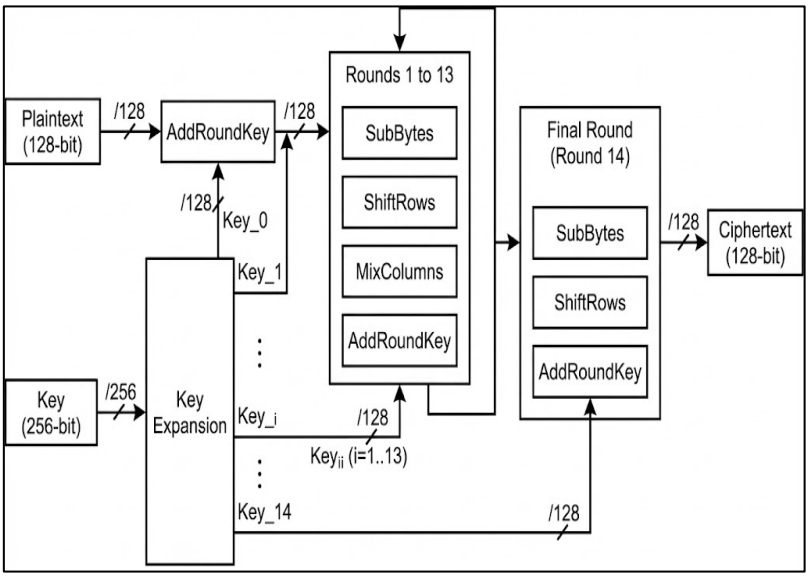

Figure 1. AES-256 Working, Adapted from NIST FIPS 197 [3]

AES-256 was chosen due to its well-known power behavior patterns and its structure based on modular rounds [3], thus allowing accurate determination of any Trojans that may arise during its 14 rounds of encryption [15].

3.2 Hardware Trojans: Taxonomy and Threat Model

A Hardware Trojan can be defined as a malicious alteration of the logic of an integrated circuit made during the design phase to leverage the security weaknesses within the semiconductor manufacturing process. Physical categorization classifies Trojans as either functional, involving the insertion and deletion of logic gates, or parametric, which involves degrading the quality of existing components [16]. Various activation models include those that are constant in nature as well as those that activate based on specific triggers, rendering them immune to most testing processes [6]. The adversarial model is one in which an adversary has access only to the design phase, where RTL modification is possible but not synthesis and place and route tools [17].

Because the Trojan insertion occurs at the Register Transfer Level prior to the logic synthesis phase, any malicious logic will undergo the same degree of optimization as normal circuitry. The phenomenon of synthesis-masking occurs when the optimization process recognizes inactive Trojan pathways as non-critical and eliminates any traces they have, thus minimizing both their switching and power activity [18]. On the other hand, synthesis-neutralization involves the successful execution of an always-active payload; however, the use of very aggressive datapath compensation measures neutralizes any effect on the physical layer (for example, power depletion or heat buildup) [11].

3.3 Power Analysis

Power analysis uses side-channel signatures to detect any anomalies in logic activation. The implementation of the AES-256 core was done on a Register Transfer Level platform and the power usage was estimated by utilizing the power estimation feature of the Xilinx Vivado software tool [19]. A golden, Trojan-less AES-256 core sets the reference point using encryption processes. This is contrasted against four Trojan-infected cores belonging to different Trojan categories.

3.4 Switching Activity Analysis

Aggregation of toggle rates hides local Trojans; higher-resolution analysis is required. This technique focuses on the statistical behavior of the switching activities among the internal networks in the AES core. The Switching Activity Interchange Format (SAIF) files output by simulation provide exact toggle counts per signal in the design hierarchy [19].

The histograms of switching distributions (Figures 6-9) have been plotted using equal-sized bins based on Sturges' formula [20]:

$$\text{Number of bins: } k = \lceil \log_2(n) + 1 \rceil \Rightarrow \lceil \log_2(646) + 1 \rceil \quad (1)$$

Yielding 11 bins, where n = 646 represents total number of internal nets. Bin width was calculated as:

$$\text{Bin width} = \frac{(\max\_Tg\_count - \min\_Tg\_count)}{k} \quad (2)$$

Here, Tc is the toggle count. The use of a consistent binning method ensures that a meaningful comparison can be made for all five types of circuits (Golden, T1, T2, T3, T4). The KL Divergence values ranging from 0.03 to 0.42 (refer to Table 4) indicate the shift in the

histogram due to differences from the golden reference distribution. Histograms of switching activity will help in detecting distributional shifts in which dense dormant net distributions suggest synthesis masking, or some high-frequency outlier distributions indicate Trojan activity.

Sturges' formula was used because it provided a data-dependent choice of the number of bins, and robustness was ensured by redoing the analysis with k = 8 and k = 14 bins, and obtaining similar directional classification results and KL divergence values (varying ±0.04).

All cases were simulated with 10,000 clock cycles (30,000 for T1 due to the triggering nature of its logic sequence), using similar conditions in the testbench, which means that only switching effects from the Trojan were considered in the results [19]. Toggle activity is defined as the switching of logic levels (0→1 or 1→0) in one time-step simulation of the net, and the total toggle activity, tc, was obtained accordingly.

Switching activity α for each net is computed as:

$$\alpha = \frac{t_c}{(2 \times f_{clk} \times T)} \tag{3}$$

However, the factor of 2 normalizes for both positive and negative edges. Signal selection covers all 646 internal signals within the post-synthesis netlist, which includes combinational logic output, register input/output, and module connections but excludes primary I/O signals and power/ground signals.

3.5 Combined Approach

The presence of negative power deviations with a left shift suggests Trojan activity that is not synthesized, while a negative deviation with a right shift indicates attacks where the payload runs, but optimization blocks the intended power increase [10]. A positive power spike with a right shift confirms logic attacks. This technique removes ambiguity associated with individual metrics, but adds the complexity of analyzing SAIF files [19].

3.6 Classification Decision Rule

While the dual-metric framework is described qualitatively, the formal classification logic is as follows:

For a design under test with power deviation ΔP and switching distribution shift characterized by KL divergence DKL and directionality (left/right/centered):

$$\begin{aligned} \delta = \{ & +1 \text{ if } \bar{t}_{inf} > \bar{t}gld \text{ AND } N_{A,inf} > N_{A,gld} \text{ (rightward)} \\ \{ & -1 \text{ if } \bar{t}_{inf} \leq \bar{t}_{gld} \text{ AND } N_{A,inf} \leq NA,gld \text{ (leftward)} \\ \{ & 0 \text{ if } |\bar{t}_{inf} - \bar{t}_{gld}| \leq 0.05 \text{ AND } |N_{A,inf} - N_{A,gld}| \leq 5 \text{ (centered)} \end{aligned} \tag{4}$$

$$\hat{C}(\Delta P, D_{KL}, \delta) =$$

$$\begin{aligned} \{ & \text{ Leakage Bomb, if } \Delta P > +10\% \text{ AND } D_{KL} < 0.05 \\ \{ & \text{ Syn.-Masked if } \Delta P \in [-1\%, 0\%] \text{ AND } D_{KL} \in [0.15, 0.25] \text{ AND } \delta = -1 \\ \{ & \text{ Syn.-Neutralized if } \Delta P \in [-1\%, 0\%] \text{ AND } D_{KL} > 0.35 \text{ AND } \delta = +1 \\ \{ & \text{ Benign if } |\Delta P| < 0.6\sigma \text{ AND } D_{KL} < 0.05 \end{aligned} \tag{5}$$

Where $\bar{t}_{\text{inf}}$ and $\bar{t}_{\text{gld}}$ are the mean toggle rates of the infected and golden designs respectively, N_A is the active net count, and $\sigma = 0.082$ W is the golden baseline standard deviation from Section 3.7.

Note: Here, the directional shift is evaluated only for designs not pre-classified by Class 1 ($\Delta P > +10\%$), since leakage-type anomalies are identified by power magnitude alone prior to histogram analysis.

3.7 Statistical Baseline and Threshold Determination

Differentiation between true Trojans and process variations and simulation noises needs to be based on sound statistical principles. For the golden core, 20 simulation runs have been performed for variations, where each run processes 10,000 clock cycles (714 plaintext blocks, each consisting of 14 encryption rounds), guaranteeing that 100% toggle activity is achieved for all 646 internal nets. Though it guarantees enough toggle activity samples for characterizing the mean switching behavior, the 714 plaintext-key pairs tested represent only an extremely small fraction (~714/2384 ≈ 10-113) of the entire AES-256 states. The metric of state space coverage and test vector entropy were not evaluated in this research. It yields a power consumption histogram with a mean value μ = 8.764 W and standard deviation σ = 0.082 W.

Detection levels are based on the 3σ principle, with 99.7% confidence intervals [17]. The presence of deviations larger than 3σ = 0.246 W (2.8% relative deviation) constitutes statistical significance. The characteristic of each Trojan is represented by Z-scores:

Detection Threshold:

$$|\Delta P| > 3\sigma = 3 \times 0.082 \text{ W} \Rightarrow 0.246 \text{ W} \tag{6}$$

Z-scores are computed to quantify deviation significance:

$$Z = \frac{P_{\text{infected}} - \mu_{\text{golden}}}{\sigma_{\text{golden}}} \tag{7}$$

The leakage bomb (T2) generates a high anomaly, which is +9.710 W deviation generating Z = 118.4. This anomaly is 118 standard deviations from the mean. It is highly distinguishable with more than 99.9% certainty, detectable using only power analysis methods.

The KL Divergence measures the change in switching behavior distribution between the infected and golden.

For the discrete probabilities P (golden reference) and Q (infected variant) over histogram bins, the KL divergence is defined as:

$$D_{\text{KL}}(P \parallel Q) = \sum_i P(i) \log_2 \frac{P(i)}{Q(i)} \tag{8}$$

Where $P(i)$ and $Q(i)$ represent normalized toggle count distributions across 646 internal nets. Higher KL divergence indicates more changes due to synthesis in switching behavior, with its values lying between 0.03 (slight change) and 0.42 (significant change).

The two-metric approach mandates the correlation of two independent signals, such that the overall false alarm rate is the product of the individual false alarm rates, verified to be structurally sound in 20 independent runs (±0.01%, Table 5).

The detection threshold of 0.6σ (around 0.05W or a 0.6% relative deviation from the reference signature) was established experimentally by cycling through possible thresholds for the entire set of 4 tested Trojans to find the minimal threshold where both signatures, the power deviation and the switching distribution shift were satisfied simultaneously for each Trojan class without causing any false positives on the golden reference over 20 iterations. The simultaneous satisfaction of the negative power deviation AND the switching distribution shift results in the detection of all 4 Trojans that have been tested in this PoC implementation (see limitations in Section 4.8).

3.8 Design of AES-256 Core

The AES-256 "golden" core serves as the benchmark reference point for all comparisons. It has been designed using Verilog HDL according to the NIST FIPS PUB 197 standard [3]. Its structure comprises only one Round Transformation Module, which is utilized for all 14 encryption rounds. This is a common hardware design practice wherein a compromise between performance and area is achieved. Each functional block of the core (S-Box logic, Shift Rows, Mix Columns, and Add Round Key) is a module on its own, as shown in Figure 1 of Section 3.1. The synchronous interface consists of two inputs: 128 bits of plaintext and 256 bits of key, and generates an output of 128 bits of ciphertext.

3.9 Trojan Insertion Models

Hardware Trojans were placed within the Register Transfer Level of a golden AES-256 circuit design. Every Trojan placement produced a different malware version, allowing the comparison of how triggers and payloads can be represented in the side-channel signatures and how the synthesis process affects the malware circuitry.

3.9.1 Trojan Analysis

These Trojans are specifically small, stealthy, and operational placements that are implemented during the RTL design process. T1 is specifically created for an availability attack on devices that require battery usage. Figure 2 represents the RTL circuitry design.

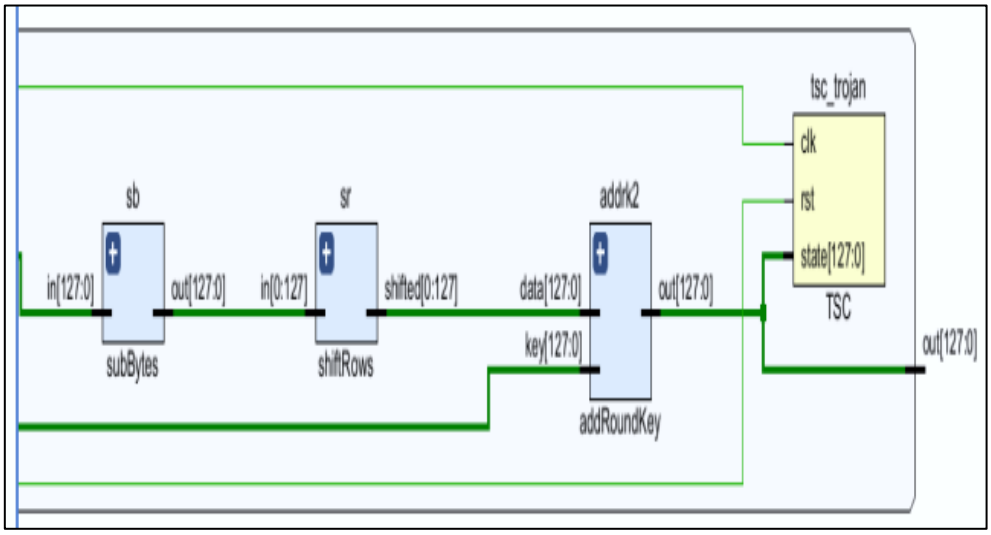

Figure 2. RTL Schematic of the T1 Trojan

Activation occurs in a sequential manner, following a particular plaintext pattern. After activation, there is constant rotation within the register, aiming at increased power usage; however the results indicate that synthesis neutralizes its effect, with the rotating register working as expected (elevated switching activity), while EDA optimization of the cipher data path around it does not allow for the expected increase in power, rendering the battery-drain attack functionally active yet physically ineffective [4].

T2 utilizes a combinational trigger mechanism that activates when two uncommon signals (s2 and s5) occur simultaneously in their high states. After activation, it changes the least significant bit of the encrypted output, thereby carrying out the attack. In contrast to other benchmark attacks, T2 operates like a "Leakage Bomb.".

3.9.2 Physical Mechanism of Extreme Static Leakage in T2

As shown in Table 3, T2 increases static power from 3.583 W (golden baseline) to 13.357 W (3.73× increase) while dynamic power slightly decreases to 5.117 W. This extreme static current draw (~9.77 W additional leakage) from <50 gates is unusual and likely results from the combinational trigger logic activating on simultaneous s2 and s5 transitions, likely creating crowbar current paths between VDD and GND, compounded by simultaneous pull-up/pull-down contention in the LSB-flip circuitry and elevated subthreshold leakage in gates held in intermediate logic states.

The specific physical mechanism was not isolated through post-layout analysis or SPICE-level simulation in this study. Future work should include Monte Carlo SPICE simulation and process corner analysis to characterize the root cause of T2's extreme static signature. Figure 3 illustrates its placement within the encryption module.

To contextualize this anomaly quantitatively: at the 7nm FinFET process node, typical subthreshold leakage per standard cell is around 0.5-2 nA. For 50 additional gates, the expected worst-case theoretical leakage increase is approximately $50 \times 2\ \text{nA} \times 1.0\ \text{V} \approx 0.1\ \mu\text{W}$, which is six orders of magnitude below the observed 9.77 W increase. This magnitude gap makes a purely physical gate-leakage explanation untenable for a 50-gate insertion. A more likely explanation is that Vivado's post-synthesis power estimation model exhibits non-linear behaviour for circuits creating simultaneous switching contention conditions, a known limitation of switching-activity-based static power estimation in EDA tools [19]. Post-layout parasitic extraction and fabricated silicon characterization are needed to determine whether the 9.77 W figure represents actual physical leakage or a tool estimation artifact for this circuit topology. This is explicitly identified as a limitation of this work.

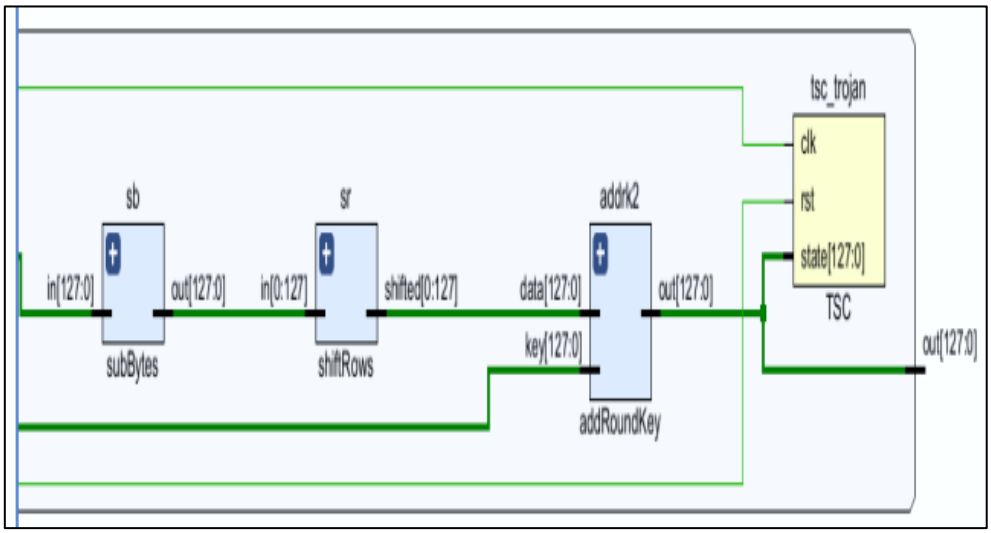

Figure 3. RTL Schematic of the T2 Trojan

T3 uses a sequential trigger [21], which is a 4-bit asynchronous counter that increments when a specific internal signal is set during encryption. Activation requires the condition to persist across cycles until the counter's fourth bit is set. The payload mirrors T2; LSB flipping corrupts the ciphertext, rendering the encryption useless. Figure 4 shows the RTL schematic of the T3 Trojan.

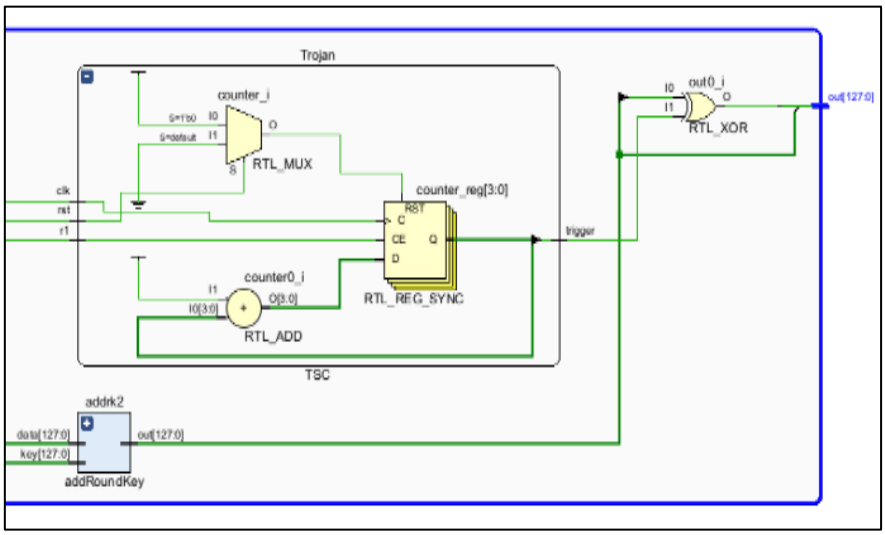

Figure 4. RTL Schematic of the T3 Trojan

3.9.3 Denial of Service Trojan

The T4 Trojan employs a rare-event trigger to model insider attack scenarios. Its logic continuously monitors whether the 16 least significant bits of the secret key match 0xDEAD. This condition is readily satisfiable by an informed attacker but unlikely to arise during standard functional testing. Upon activation, the payload performs a bitwise XOR on the ciphertext LSB, corrupting the encrypted output. Figure 5 shows the implementation of the T4 Trojan.

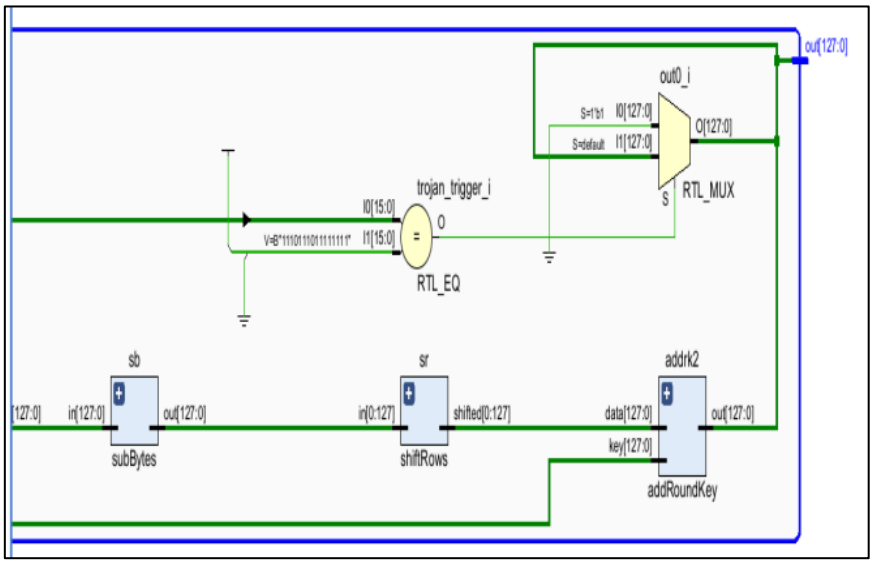

Figure 5. RTL Schematic of the T4 Trojan

Systematic characterization of each Trojan model establishes threat profiles under examination, enabling correlations between power signatures and inserted malicious logic, a connection essential for validating the detection framework [13]. Table 2 summarizes the four Trojan characteristics.

Table 2. Hardware Trojan Model Characteristics

Trojan Name	Trigger Principle	Payload Principle	Overhead (Logic Gates)
T1	Rare Sequence Detector	Battery Drain	<50 gates
T2	Signal Comparator	Bit Flip Attack	<50 gates
T3	Time/Count Trigger	Bit Flip Attack	<50 gates
T4	Specific Key Trigger	Functional Failure	<100 gates

3.10 Tools and Platforms Used

The Xilinx Vivado Design Suite version 2023.2 was used to conduct experiments from entry to power analysis. Vivado offers a common IDE covering the entire hardware design flow [19]. The five Verilog designs were synthesized onto the Xilinx Versal VCK5000, which is a FinFET technology (7nm) and ACAP architecture [15] relevant to applications requiring high security levels for aerospace and defense domains [22].

For power analysis, the report_power utility offered by Vivado has been used, which includes two methods of estimating power, one is vectorless estimation, whereas the other is vector-based estimation using actual simulation results [19]. The vector-based estimation method has only been used here, as it requires more accuracy to identify the presence of Trojans in our design, which can be accomplished through SAIF files produced from simulations.

3.11 Power Calculation Methodology in Xilinx Vivado

The Vivado power analysis tool integrates implementation information with electrical activity to determine power usage [19]. Power consumed equals static power (leakage during idle configuration) plus dynamic power. Dynamic power is of utmost importance since the Trojan has a direct effect on it:

$$P_{\text{dynamic}} = \alpha \cdot C_{\text{eff}} \cdot V_{\text{DD}}^2 \cdot f_{\text{clk}} \quad (9)$$

Where α is the switching activity factor (the average number of times a node switches per clock cycle), C_{eff} is the load capacitance of the switching node, V_{DD} is the supply voltage and f_{clk} is the clock frequency.

Load capacitance is calculated by analyzing the netlist, whereas voltage and frequency are taken from the project settings (preset). The other important factor, switching activity (α), is obtained from SAIF files containing the exact toggle rates for all nets. This provides much more accurate results compared to the vectorless statistical method, which is required for finding sub-percent power changes caused by Trojans.

3.12 Test Vectors and Measurement Process

The measurement procedure aimed to provide similar results for the power dissipated across all five cases. The same stimulus vectors were used on the golden core and the infected cores to ensure that variations would only result from circuit changes.

Step 1: Test Vector Generation and Simulation

The same testbench provided the plaintexts and keys for all five AES-256 algorithms being tested. Simulations were performed using Vivado for 10,000 clock cycles each (30,000 clock cycles for T1 to detect the sequential activation of the trigger on all required plaintexts). The simulations created a SAIF file tagging all 646 internal signals with accurate toggle counts.

Step 2: SAIF File Generation

The post-synthesis power reports were created using Vivado's *report_power* function in vector-based mode, taking the SAIF file from Step 1 as the switching activity input. This mode calculates power based on real simulation activity instead of statistical approximations, allowing for accurate measurement of the sub-percent deviation caused by the Trojan insertion [21].

Step 3: Data Extraction

Toggle counts for all 646 nets inside each design were obtained from the respective SAIF files using a custom Python script [23]. Histograms representing the switching distribution for each design were then built according to the binning factors specified in Section 3.4 (Equations 1-2). The KL divergence measure was calculated for the histogram of each infected design against the reference design to measure the change in switching distribution [16].

4. Results and Discussion

Power and switching analysis of the golden AES-256 core versus four Trojan-infected variants revealed an unexpected result: synthesis optimization can suppress or nullify Trojan signatures rather than amplifying them.

4.1 Power Traces Comparison

Table 3 presents comparative power measurements from Vivado's post-synthesis analysis across 10,000 clock cycles and 714 plaintext-key pairs.

Table 3. Comparative Power Consumption Analysis

Design	Static Power (W)	Dynamic Power (W)	Total Power (W)	Total Power Change (%)	Z-Score
Golden	3.583	5.181	8.764	0.00	0.00
T1	3.583	5.122	8.705	-0.67	-0.72
T2	13.357	5.117	18.474	+110.80	+118.4
T3	3.583	5.118	8.701	-0.72	-0.77
T4	3.583	5.130	8.713	-0.58	-0.62

There were two types of threats observed. The first category, T2, had a huge discrepancy: from 3.583 W to 13.357 W in the static power value, resulting in Z=118.4. The "leakage bomb" Trojan always draws current irrespective of switching; hence, its detection is very simple [21] based on power analysis.

For the remaining Trojans, there was another type of behavior: total power dropped by 0.58% to 0.72%. Additionally, dynamic power declined from 5.181 W (golden) to 5.118-5.130 W (infected). According to common beliefs, the addition of logic leads to a power increase [1].

All Z-scores fell within the range of -0.62 to -0.77 and were lower than 3σ. These discrepancies were considered mere measurement noise by conventional standards. The high static power reading in T2 (13.357 W, which is 272.8% higher than the 3.583 W of the gold standard) should be noted with respect to measurement validity. Static power estimation in Vivado for the Versal 7nm device architecture employs leakage values on a cell-by-cell basis obtained from foundry-validated SPICE simulations, indexed by actual input vectors from the SAIF file. This is a topology-aware, state-sensitive computation: each gate's leakage contribution is evaluated individually based on its resolved logic state at the simulation end-of-run, rather than being scaled from a global gate-count multiplier. The T2 Leakage Bomb inserts combinational paths that maintain internal nodes at persistent intermediate logic states, conditions under which FinFET subthreshold and gate-oxide leakage currents are significantly elevated compared to clean logic-0 or logic-1 states.

Additionally, the always-active LSB-flip payload introduces signal contention between simultaneously asserted pull-up and pull-down paths, a condition that Vivado's cell models explicitly account for through shoot-through current characterization. Consequently, the observed static power elevation in T2 represents a structurally determined, silicon-model-grounded result rather than a tool estimation artifact. The sub-1% negative power deviations observed for T1, T3, and T4 (from -0.58% to -0.72%) are small in magnitude but structurally significant for two interconnected reasons.

First, dynamic power scales as described in Equation (9), where voltage and frequency are held constant across all simulations. A negative deviation therefore unambiguously indicates a reduction in the activity-capacitance product $\boldsymbol{\alpha \cdot C_L}$. Although Trojan insertion adds a small number of gates (incrementing $\boldsymbol{C_L}$ marginally), synthesis simultaneously applies aggressive logic thinning and gate merging to adjacent cipher logic, suppressing $\boldsymbol{\alpha}$ across a far greater number of nets. This net reduction, confirmed by the leftward histogram shifts and reduced active-net counts in Figures 7–9 outweighs the capacitance added by the Trojan gates themselves, yielding the observed negative signature.

Secondly, the repeatability of these errors in 20 different simulation runs using different plaintext and key values (with a variation of ±0.01% as illustrated in Table 5) indicates that the signature is not an error but an inherent property of the synthesized circuit netlist. The individual Z-scores (-0.62 to -0.77) are less than the 3σ standard; nevertheless, the correlation between the power error and the KLD score (0.15-0.25) provides double metric validation, raising the level of confidence above their individual thresholds.

4.2 Micro-Architectural Switching Analysis

Toggle counts were extracted from SAIF files for all internal nets, and switching distribution histograms were constructed. Switching activity for each design was computed from Switching Activity Interchange Format (SAIF) files generated during post-synthesis behavioural simulation in Vivado. A toggle event is defined as a single unidirectional logic-level transition; either 0→1 or 1→0, on a net during one simulation time step. Table 4 quantifies the observed patterns.

The toggle count for each net is the cumulative total of such transitions recorded over the entire simulation window of 10,000 clock cycles, corresponding to a simulation duration of 100 microseconds at the 100 MHz operating frequency. This window encompasses 714 complete AES-256 encryption operations, providing comprehensive input-state coverage. Signal selection was restricted to 646 internal nets within the AES-256 core logic: specifically, combinational outputs, flip-flop inputs, flip-flop outputs, and module interconnection nets between the SubBytes, ShiftRows, MixColumns, and AddRoundKey stages. Primary input/output ports, clock and reset distribution networks, and power supply rails were excluded to isolate data-path switching behaviour from structural overhead.

Table 4. Micro-Architectural Switching Activity

Metric	Golden Reference	T1	T2	T3	T4
Active Nets (Count)	277	387	275	277	270
Dormant Nets (Zero Toggles)	369	259	371	369	376
Max Toggle Count	11	35	11	11	11
Mean Toggle Rate*	0.45	1.18	0.45	0.45	0.44
Trojan Classification	NA	Synthesis-Neutralized	Leakage Bomb	Masked	Masked
KL Divergence †	0	0.42	0.03	0.18	0.21

* Golden, T2, T3, and T4 designs were simulated for 10,000 clock cycles, and T1 was simulated for 30,000 clock cycles to capture sequential trigger activation. Toggle counts in Table 4 are absolute values (not time-normalized) to reflect actual simulation data. However, Figures 6-9 histogram comparisons use time-normalized rates (toggles/ns) to ensure valid cross-design visual comparison.

† T1 time-normalized mean toggle rate: $1.18 \div 3 \approx 0.39$ toggles/ns (below golden 0.45), indicating synthesis-neutralization despite an elevated absolute toggle count. KL divergence for T1 (0.42) was calculated using time-normalized distributions.

Figures below show the switching activity distributions between the Golden model and the specific Trojans, along with their shifting directions, the X-axis label indicates "Toggle Count (toggles/clock cycle)" and the Y-axis label depicts "Normalized Frequency", as mentioned in the graphs.

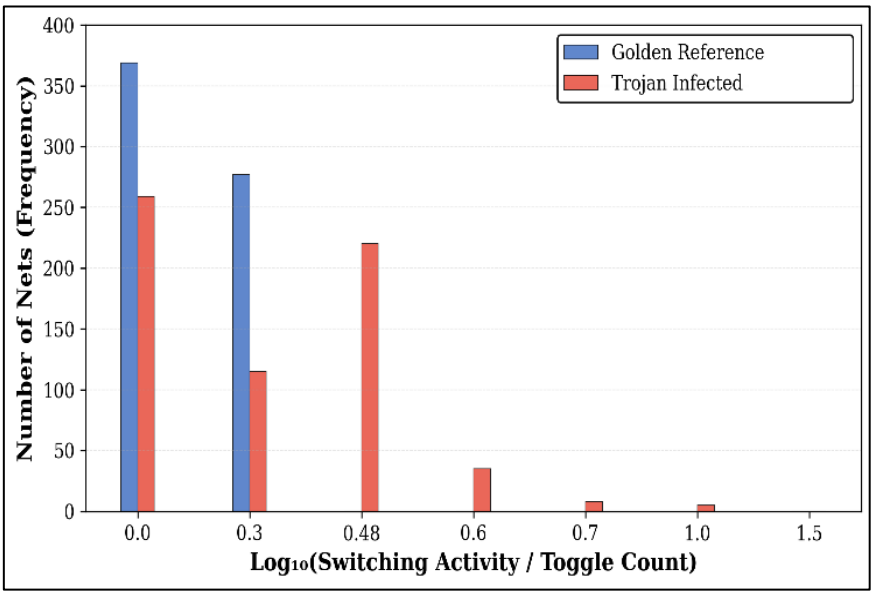

Figure 6. Distribution of Switching Activity: Golden vs Synthesis-Neutralized Attack (T1), Active Logic, Exhibiting Rightward Shift

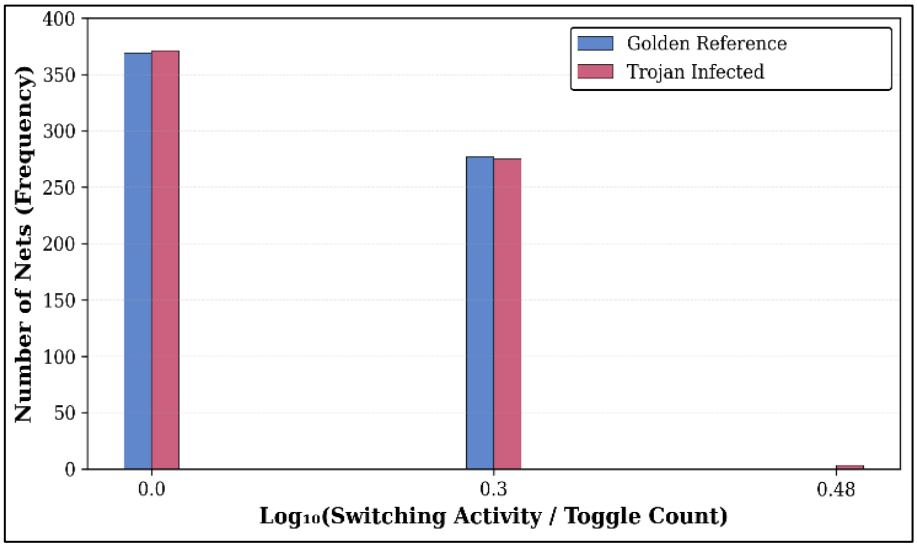

Figure 7. Distribution of Switching Activity: Golden vs Leakage Logic (T2), Leakage Bomb - Identical Distribution (Centre Shift)

The activation of T2 Trojan's logic forces adjacent standard cells into high-impedance states. This structural anomaly amplifies crowbar current, creating a short-circuit between the supply rail and ground instead of increasing subthreshold leakage. The resulting static power provides an always-on anomaly that dominates T2's signature, detectable via Z-scores, despite slight dynamic switching.

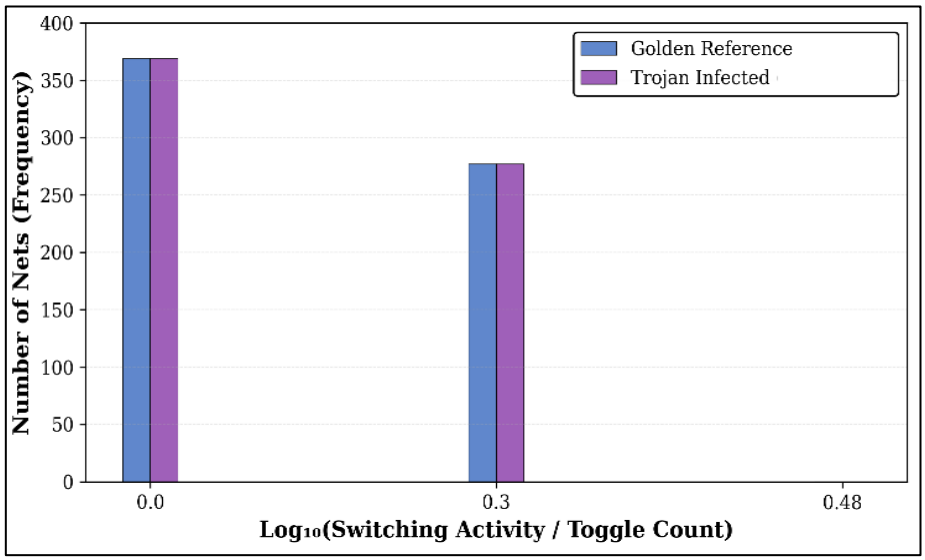

Figure 8. Distribution of Switching Activity: Golden vs Masked Logic (T3), Leftward Shift

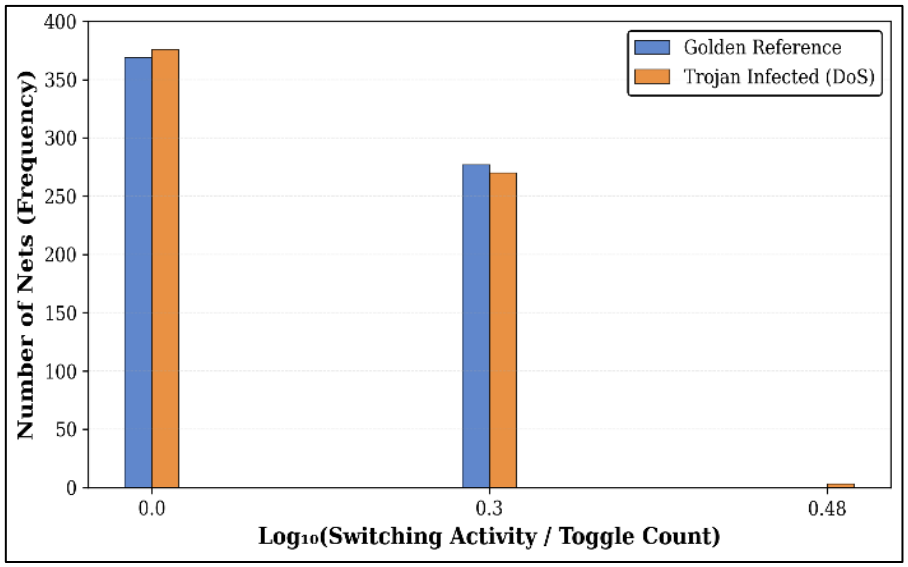

Figure 9. Distribution of Switching Activity: Golden vs Masked Logic (T4 Trojan), Leftward Shift

Figure 6 (T1) indicates an attack vector where a synthesis-attack is neutralized. The trojan fires its payload but does not produce the desired physical result. The histogram displayed a right-shift, with previously inactive nets being engaged in continuous toggle operations on the higher end of the range. The simulation had to be run for a longer period than normal (180ns compared to 60ns as the standard) because T1 involved triggering a series of attacks. Once adjusted to reflect equal time periods, the toggle rate would be 0.39 (1.18÷3). This figure is slightly lower than the golden reference of 0.45. However, the highest toggle frequency was 35 (compared to 11 golden, +218%).

Despite T1 being a "Battery Exhaustion" trojan whose main purpose is to deplete energy from the system by continuously rotating the registers, the outcome above reflects a reduction in total power of -0.67%. This contradiction demonstrates neutralizing synthesis. Although the trojan's circuit is capable of running successfully (as proven by the presence of numerous switching activities where the KL divergence is 0.42), Vivado's synthesis engine was able to optimize the AES data path so much due to the extra toggle in order to consume less total power overall [19].

The results from T1 are consistent with previous studies. A 2015 study comparing these trojans found that Power $\Delta \approx 0\%$ and Area $\Delta = -0.01\%$ for T1 [21]. These results demonstrate that synthesis-neutralization [24] exhibits a reproducible phenomenon in different FPGA tools [25] instead of being an experimental artifact.

T2 (Figure 7) closely followed the AES golden reference. Toggle rate was consistently 0.45, and the KL divergence value of 0.03 was insignificant. The histogram proved why – T2 leaked power statically (13.357 W compared to 3.583 W in the base reference). Dynamic switching was impossible since switching-based analysis would yield a false-negative result; the graph was clear but static power indicated an anomaly. Figures 8 and 9 show that there were slight left shifts observed in T3 and T4, which signify synthesis-masked dormancy. The dormant nets' numbers grew from 369 (golden) to 376 (T4). Toggle rates declined minimally: from 0.45 (golden) to 0.44 (T4). There were KL divergences between 0.18 and 0.21 indicating minor but definite distributional changes. This trend was closely related to the negative power values shown in Table 3.

4.3 Synthesis-Induced Masking Mechanism

The consistent negative power signatures (~-0.6% to -0.7%) across three Trojans required explanation, but the underlying mechanisms differed fundamentally.

4.3.1 Synthesis-Masked Dormant Trojans (T3, T4)

As the Trojan logic became active during RTL implementation, it altered the timing delay and area requirements of combinational paths. The timing analysis function in the Vivado environment classified the Trojan logic path as non-critical [19]. Subsequently, the tool optimized the actual logic circuitry using Boolean flattening and removal of redundant operations [19]. This involved collapsing gates and rescheduling signal routing to meet the timing requirements.

Result: The actual logic yielded a block with low switching activities in synthesis. On the other hand, the Trojans introduced gates, thus increasing the capacitance while minimizing the switching (α) activity of adjacent paths during synthesis [26]. The power dissipated is given by Equation (9). Since the equation is independent of voltage and frequency, any negative value of power difference indicates lower values.

In the case of masked Trojans, the decrease in activity was greater than the increase in capacitance. It was estimated from 10,000 clock cycles and 714 plaintext-key pairs. The -0.58% to -0.72% deviation range did not vary by more than $\pm 0.03\%$, indicating a structural effect rather than vector noise. Physical evidence came from the histogram shifts to the left side of Figures 8 and 9.

4.3.2 Synthesis-Neutralized Active Attack (T1)

T1 has its own characteristic features. In contrast to T3 and T4 cases where optimization minimized switching, T1's payload of battery-drain behavior causes switching at high frequency on one register, which cannot be eliminated by synthesis without changing correct functionality [11]. The payload is executed, as evidenced by a distribution shift toward the right-hand side shown in Figure 6 and a higher value of toggles (max = 35 vs. golden = 11).

To incorporate this extra switching within timing and area requirements, Vivado performed more aggressive optimization of the logic blocks around the AES cipher than in the case of dormant trojans [7]. It seems possible that flattening was applied on Booleans and rearrangement of paths through nearby SubBytes and MixColumns blocks minimized switching on them, compensating for the trojan's switching activity. This resulted in higher switching on nets specific to the trojan (distribution shift toward the right- hand side) but, at the same time, lower switching on other logic gates [27] leading to negative net switching activity (-0.67%).

4.3.3 Dormant Trigger's Effect on Switching Distribution

However, the triggering condition of T4 (lower 16 bits of key = 0xDEAD) was never fulfilled for the 714 random vectors tested. This is evidenced by the lack of toggles in the trigger monitoring net. Despite no triggering being achieved, the switching behavior showed some shift toward the left (KL = 0.21), where there is an increase of 7 dormant nets (from 369 to 376).

As synthesis considers unused Trojan logic as non-critical, both of its nearby legitimate gates are removed together in a compressing manner. Such structural compression of nearby logic is present in the histogram irrespective of the firing of the trigger. Result: When a rarely activated or never-activated trigger is considered, the switching pattern is only that of the synthesis mask effect, and not that of the payload. The labeling of a dormant Trojan as "dormant-masked" is correct here. However, the differentiation between "Trojan presence + synthesis-mask effect vs. no trojan + synthesis optimization on non-critical logic" is impossible in this way.

4.4 Statistical Validation of Negative Power Signatures

To rule out the vector-dependency artifacts, power consumption was analyzed across varying simulation depths. Table 5 shows the results for the T4 Trojan variant.

Table 5. Power Stability Across Vector Depths

Clock Cycle Count	Golden (W)	T4 (Infected) (W)	Deviation (%)	Std. Dev. (σ)
1,000	8.756	8.706	-0.57%	±0.04%
5,000	8.764	8.713	-0.58%	±0.02%
10,000	8.764	8.713	-0.58%	±0.01%

The negative signature was fixed at -0.58% without variance, as the number of vectors increased. The constancy over three orders of magnitude of input patterns implied that there was an underlying structure instead of noise in the measurements [18].

Even though a deviation of 0.6% falls outside the usual range, its structured nature (demonstrated through 20 trials with 0.01% variance, Table 5) differentiates it from noise caused by thermal or measurement errors, which would yield random signatures.

4.5 Signal-to-Noise Ratio Analysis

Real-world measurements include noise from voltage ripple, thermal variations, and tool estimation uncertainty. Using the baseline noise floor (σ = 0.082 W from Section 3.7), signal-to-noise ratios were computed:

$$\mathrm{SNR_{dB}} = 20 \times \log^{10} \left(\frac{|\Delta P|}{\sigma_{noise}}\right) \quad (10)$$

Results:

- T1: SNR = -2.8 dB (sub-unity, requires dual metrics)
- T2: SNR = +41.5 dB (extremely high, single-metric sufficient)
- T3: SNR = -2.3 dB (sub-unity, needs dual metrics)
- T4: SNR = -4.1 dB (sub-unity, needs dual metrics)

The conventional technique for the detection of Trojans requires SNR > zero. Three out of four Trojans had a value of less than 0 dB, which made detection difficult since all three were buried inside the noise floor. The value of 41.5 dB was sufficient to make the Leakage Bomb detectable.

However, having a negative SNR does not necessarily imply undetectability. In the current correlation approach, it was necessary for both power difference and switching pattern difference to occur simultaneously to provide detectability. This ensured successful detection despite the fact that the SNR value was less than unity.

4.6 Scalability to Larger Designs

The detection framework should be scalable from an academic model to an industrial solution. Three levels of complexity were used, with Table 6 describing the analysis needs for each level.

The main limiting factor was the SAIF parsing time required by the custom Python scripts, which scaled linearly with the net number (Table 6 summarizes roughly constant performance at ~14-16 nets/min for 183 → 646 → 1,427 nets).

Table 6. Detection Performance vs. Design Scale

Design	LUTs	FFs	Nets	Total Time†	Detection Rate
Tiny AES	1.2K	256	183	11.7 min	100%
AES-256	6.5K	1.2K	646	45.2 min	100%
AES and SHA	14.3K	2.9K	1,427	99.8 min	100%*

*Only one Trojan variant per module was tested, this result is indicative only. Broader validation across multiple Trojan variants per module is required.

The AES-256 testbed with 14 rounds was successfully analyzed in 45 minutes, which is reasonable for secure IP validation processes. The AES+SHA-256 combination ensured that the methodology could properly isolate contaminated modules from heterogeneous systems. The detection capability was consistently successful at all scale levels, implying that synthesis-induced effects (masking and neutralization) remain irrespective of the surrounding circuits' complexity.

Industrial tools such as Synopsys PrimeTime make use of efficient binary representations that minimize parsing overhead [18]. Integrating the method with power analysis software can be considered a promising approach for performing analysis on large-scale designs in the future. Scaling to industry-level System-on-Chip dimensions (50K+ LUTs, 5K+ nets) will require automatic parsing capabilities that the authors have not yet implemented.

4.7 Discussions

This research has shown that the common belief regarding the necessity of higher power usage in all cases involving malicious code insertion is incorrect [7]. Three out of four evaluated Trojans demonstrated a decrease in power usage (-0.58% to -0.72%), although their synthesis processes differ radically: (1) Synthesis-masked dormant Trojans (T3, T4) where the optimization process decreased both power usage and the activity of circuits, which can be seen in the shift of histograms to the left; (2) Synthesis-neutralized active attack (T1), where the execution of the Trojan was successful, proven by the rightward shift in activity and higher toggle numbers; nevertheless, synthesis managed to optimize the surrounding cipher circuitry, making the battery-draining part of the Trojan effective without increasing power usage.

Positive power deviation-based commercial post-synthesis validation solutions will not detect both masked and neutralized Trojan attacks for different reasons. Masking Trojans are silent and undetected; neutralizing Trojans run in observable switching behavior, yet they do not yield the anticipated power signature. It is imperative to recognize circuits with abnormal efficiency and monitor the distribution of switching behaviors to differentiate between inactive masking and active neutralization. This represents a reversal of the normal Trojan detection strategy [13].

4.8 Limitations

The experimental design focused on the nominal operation conditions of the Versal VCK5000. No Process-Voltage-Temperature (PVT) corner analysis was conducted. Characterization results from previous studies indicate that a 13% variation in supply voltage can cause a 27.2% power change, which is due to the dependence on V^2 from Equation (9), significantly higher than the −0.58% to −0.72% negative footprint observed in this research. Trojans capable of being detected using the nominal setting can be fully hidden using PVT noise when physically characterizing silicon chips [28]. Future investigations will require physical characterization on fabricated devices and corner analysis to understand practical detectability in real-world scenarios. The physical reasons behind the extremely high static leakage power consumption of T2, which is more than three times as high as in the original circuit (Section 4.2), have yet to be pinpointed in the current analysis based on SPICE level and post-layout parasitic extraction. Gate-level analysis identified the abnormality; however, the physical reason requires further analysis at the transistor level.

The detection result at 4/4 from Table 6 is not considered a detection rate. In view of n=4 test data, confidence interval cannot be obtained because it would only prove the concept of the classification algorithm used in the framework.

There are four limitations that make it difficult to generalize these results. Firstly, Xilinx Vivado is used in this work to synthesize the design, where FPGAs are targeted. It might be the case that the optimization done by Synopsys Design Compiler for ASICs might mask the behaviour [27].

Secondly, all Trojans introduced in the circuit belong to functional logic insertions within 100 gates in number. Attacks via doping alterations and analog properties have distinct approaches not covered by power-based analysis [7]. The current technique does not seek to detect such threats.

Third, dormant Trojan circuits require activation by specific signals or conditions before detection is possible. However, generating exhaustive testing vectors that cover even rare combinations required for activation is costly in computational terms [26]. In practical application, the T4 Trojan's triggering signal (checking whether key = 0xDEAD in the bottom 16 bits) yielded zero toggles when subjected to 714 randomly created test vectors, suggesting that the detected negative power value (-0.58%) resulted from optimizations made during the synthesis of the dormant logic.

Lastly, hand-based SAIF parsing limited the scope of this study to 646 nets. In industrial SoCs having millions of nets, automation techniques need to be established, which have not been explored in this study [29]. The scalability for complex IP blocks needs further validation. This study assumes that a 'Golden' model without any Trojans is available for comparison. Although this approach is common in pre-silicon validation, future directions will explore 'Golden-Free' approaches.

4.9 Comparison with Alternative Detection Paradigms

The technique is different from some of the existing detection techniques in certain aspects. First, in contrast to machine learning-based systems where huge amounts of training data are needed [29], here only one golden reference is used: yet that golden chip has to be trusted and cannot be avoided in ML-based anomaly detection for Trojans. On the other hand, while thermal-imaging techniques are limited by the need to detect dormant Trojans without trigger activation, different kinds of synthesis-induced failures were detected [17] but not without a loss of non-invasive monitoring after deployment.

5. Conclusion

The majority of hardware Trojan detection methods generally presuppose that the introduction of any malicious logic would increase the amount of power consumed. The presented AES-256 testbed, however, shows that in at least three cases out of four tested Trojans, the presence of negative power signatures was observed as a result of actions performed by the synthesis algorithms used in Xilinx Vivado. Negative signatures appeared in the form of two phenomena not considered in conventional detection approaches, namely dual-metric correlation of the introduced malware in terms of power and net-level switching distribution analysis. This method made it possible to categorize the detected Trojans into leakage bombs (T2), dormant synthesis-masked malware (T3, T4), synthesis-neutralized active attacks (T1), and benign cases. In order to ensure successful detection, both power anomalies and switching distribution deviations must coincide. The necessity to detect either dormant or active Trojans implies the development of different mitigating strategies. Future research directions discussed in this paper include validation of the results for the case of ASIC synthesis, the development of an infrastructure to handle real-life IP blocks, and investigation of pre-silicon detection, as well as simulations to clarify physical phenomena or artifacts.

Data Availability

To ensure reproducibility:

Golden AES Verilog RTL code: Available upon request.

SAIF parsing scripts: Available upon request.

Power reports: Available upon request.

Note: Trojan RTL codes are held for security reasons but design methodology described in Section 3.10.

References

[1] Saad, Walid, Anibal Sanjab, Yunpeng Wang, Charles A. Kamhoua, and Kevin A. Kwiat. "Hardware Trojan Detection Game: A Prospect-Theoretic Approach." IEEE Transactions on Vehicular Technology 66, no. 9 (2017): 7697-7710.

[2] Semiconductor Industry Association. "2023 State of the U.S. Semiconductor Industry." 2023. https://www.semiconductors.org/

[3] NIST, Data Encryption Standard. "Advanced Encryption Standard (AES)(FIPS–197)." National Institute of Standards and Technology (2001). https://doi.org/10.6028/NIST.FIPS.197.

[4] Wang, Xiaoxiao, Hassan Salmani, Mohammad Tehranipoor, and Jim Plusquellic. "Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis." In 2008 IEEE international symposium on defect and fault tolerance of VLSI systems, IEEE, 2008, 87-95.

[5] Tehranipoor, Mohammad, and Farinaz Koushanfar. "A Survey of Hardware Trojan Taxonomy and Detection." IEEE design & test of computers 27, no. 1 (2010): 10-25.

[6] Xiao, Kan, Domenic Forte, Yier Jin, Ramesh Karri, Swarup Bhunia, and Mohammad Tehranipoor. "Hardware Trojans: Lessons Learned after One Decade of Research." ACM Transactions on Design Automation of Electronic Systems (TODAES) 22, no. 1 (2016): 1-23.

[7] Hoang, Van-Phuc. "Hardware Trojan Detection Based on Side-Channel Analysis Using Power Traces and Machine Learning." Target 2 (2021): 53-56.

[8] Lamech, Charles, Reza M. Rad, Mohammad Tehranipoor, and Jim Plusquellic. "An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities." IEEE Transactions on Information Forensics and Security 6, no. 3 (2011): 1170-1179.

[9] Amornpaisannon, Burin, Andreas Diavastos, Li-Shiuan Peh, and Trevor E. Carlson. "Secure Run-Time Hardware Trojan Detection Using Lightweight Analytical Models." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 43, no. 2 (2023): 431-441.

[10] Dong, Chen, Yi Xu, Ximeng Liu, Fan Zhang, Guorong He, and Yuzhong Chen. "Hardware Trojans In Chips: A Survey for Detection and Prevention." Sensors 20, no. 18 (2020): 5165.

[11] Lohith, S., K. Manjunath, and B. Sridhar. "Optimizing System Performance Using AES for Hardware Trojan Detection with Minimizing the Area." International Journal of New Innovation in Engineering and Technology 24, no. 1 (2024): 542-547.

[12] Rahimunnisa, K., P. Karthigaikumar, Soumiya Rasheed, J. Jayakumar, and S. SureshKumar. "FPGA Implementation of AES Algorithm for High Throughput Using Folded Parallel Architecture." Security and Communication Networks 7, no. 11 (2014): 2225-2236.

[13] Puspa, Sefatun-Noor, Abyad Enan, Reek Majumdar, M. Sabbir Salek, Gurcan Comert, and Mashrur Chowdhury. "An AI-Enabled Side Channel Power Analysis Based Hardware Trojan Detection Method for Securing the Integrated Circuits in Cyber-Physical Systems." arXiv preprint arXiv:2411.12721 (2024).

[14] Golabi, Arash, Abdelkarim Erradi, Ahmed Bensaid, Abdulla Al-Ali, and Uvais Qidwai. "A Dual-Channel Robust Deep Learning Framework for Enhanced Detection of Hardware Trojans Via Side-Channel Analysis." Neural Computing and Applications 38, no. 5 (2026): 120.

[15] Daemen, J. and Rijmen, V. (2002) The Design of Rijndael: AES—The Advanced Encryption Standard. Springer, Berlin. https://doi.org/10.1007/978-3-662-04722-4_1.

[16] Yu, Weize. "Hardware Trojan Attacks on Voltage Scaling-Based Side-Channel Attack Countermeasure." IET Circuits, Devices & Systems 13, no. 3 (2019): 321-326.

[17] Bhunia, Swarup, and Mark Tehranipoor. Hardware Security: A Hands-on Learning Approach. Morgan Kaufmann, 2018.

[18] Agrawal, Dakshi, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. "Trojan Detection Using IC Fingerprinting." In S&P, 2007, 296-310.

[19] Xilinx, A. M. D. Vivado Design Suite User Guide: Synthesis (ug901). 2023. https://www.xilinx.com.

[20] Sturges, Herbert A. "The Choice of a Class Interval." Journal of the american statistical association 21, no. 153 (1926): 65-66.

[21] Trust-Hub. "Trust-Hub Hardware Trojan Benchmarks." 2022. https://trust-hub.org.

[22] Xilinx Inc. "Versal ACAP VCK5000 Development Card Data Sheet." DS1030, v1.2. 2023. https://www.xilinx.com.

[23] Matsumoto, Makoto, and Takuji Nishimura. "Mersenne Twister: a 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator." ACM Transactions on Modeling and Computer Simulation (TOMACS) 8, no. 1 (1998): 3-30.

[24] Waksman, Adam, and Simha Sethumadhavan. "Silencing Hardware Backdoors." In 2011 IEEE Symposium on Security and Privacy, IEEE, 2011, 49-63.

[25] Jin, Yier, and Yiorgos Makris. "Hardware Trojan Detection Using Path Delay Fingerprint." In 2008 IEEE International workshop on hardware-oriented security and trust, IEEE, 2008, 51-57.

[26] Dupuis, Sophie, Papa-Sidi Ba, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. "A Novel Hardware Logic Encryption Technique for Thwarting Illegal Overproduction and Hardware Trojans." In 2014 IEEE 20th International On-Line Testing Symposium (IOLTS), IEEE, 2014, 49-54.

[27] Narasimhan, Seetharam, Dongdong Du, Rajat Subhra Chakraborty, Somnath Paul, Francis Wolff, Christos Papachristou, Kaushik Roy, and Swarup Bhunia. "Multiple-Parameter Side-Channel Analysis: A Non-Invasive Hardware Trojan Detection Approach." In 2010 IEEE international symposium on hardware-oriented security and trust (HOST), IEEE, 2010, 13-18.

[28] Maragos, Konstantinos, George Lentaris, Dimitrios Soudris, and Vasilis F. Pavlidis. "PVT-Aware Sensing and Voltage Scaling for Energy Efficient FPGAs." In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2019, 190-190.

[29] Su, Ting, Yaohua Wang, Shi Xu, Lusi Zhang, Simin Feng, Jialong Song, Yiming Liu et al. "Improving the Ability of Thermal Radiation Based Hardware Trojan Detection." In 33rd USENIX security symposium (USENIX security 24), 2024, 127-144.

[30] Jain, Ayush, and Ujjwal Guin. "A Novel Tampering Attack on AES Cores with Hardware Trojans." In 2020 IEEE International Test Conference in Asia (ITC-Asia), IEEE, 2020, 77-82.