Improving the Ability of Thermal Radiation Based Hardware Trojan Detection

Ting Su, Yaohua Wang, Shi Xu, Lusi Zhang, Simin Feng, Jialong Song, Yiming Liu, Yongkang Tang, Yang Zhang, Shaoqing Li, Yang Guo, and Hengzhu Liu,
National University of Defense Technology

https://www.usenix.org/conference/usenixsecurity24/presentation/su-ting

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Improving the Ability of Thermal Radiation Based Hardware Trojan Detection

Ting Su, Yaohua Wang*, Shi Xu, Lusi Zhang, Simin Feng, Jialong Song, Yiming Liu,
Yongkang Tang, Yang Zhang, Shaoqing Li, Yang Guo, Hengzhu Liu

National University of Defense Technology

Abstract

Hardware Trojans (HTs) pose a significant and growing threat to the field of hardware security. Several side-channel techniques, including power and electromagnetic radiation (EMR), have been proposed for HT detection, constrained by reliance on the golden chip or test vectors. In response, researchers advocate for the use of thermal radiation (TR) to identify HTs. However, existing TR-based methods are designed for the ideal HT that can fully occupy at least one pixel on the thermal radiation map (TRM). In reality, HTs may occupy multiple pixels, substantially diminishing occupancy in each pixel, thereby reducing the accuracy of existing detection methods. This challenge is exacerbated by the noise caused by the thermal camera. To this end, this paper introduces a countermeasure named noise based pixel occupation enhancement (NICE), aiming to improve the ability of TR-based HT detection. The key insight of NICE is that noise can vary the pixel occupation of HTs while disrupting HT detection. Consequently, the noise can be exploited to statistically find out the largest pixel occupation among the variations, thereby enhancing HT detection accuracy. Experimental results on a 0.13 μm Digital Signal Processing (DSP) show that the detection rate of NICE exceeds the existing TR-based method by more than 47%, reaching 91.81%, while maintaining a false alarm rate of less than 9%. Both metrics of NICE are comparable to the existing power-based and EMR-based methods, eliminating the need for the golden chip and test vectors.

1 Introduction

Hardware Trojans (HTs) are stealthy modifications to a circuit that can allow unauthorized access to and control over the content and communication of an integrated circuit (IC) [7, 14, 24, 28]. This emerging threat is compounded by the outsourcing of IC fabrication to third-party foundries due to economic and market forces. The attack poses a particularly powerful and stealthy risk because the unused spaces within the IC layout can be exploited by untrusted foundries to insert additional malicious circuits [49]. Addressing the fabrication stage attack is thus a crucial and urgent priority.

To mitigate this threat, side-channel techniques [2, 3, 4, 10, 19, 20, 22, 31, 39, 41] are studied due to their fast and cost-effective characteristics compared with destructive detection methods [27, 28]. Side-channel techniques rely on extracting the information of ICs through their power, electromagnetic radiation (EMR), and thermal radiation (TR). Dakshi et al. [2] initially proposed a countermeasure using power analysis. Subsequently, researchers have expanded upon similar concepts by utilizing EMR to enhance side-channel analysis [3, 10, 19, 20]. He et al. [19, 20] suggested that EMR traces can be used to detect HT with a considerable performance, identifying different types of HTs with an average accuracy rate of 89.2%. However, these methods rely on the IC fingerprinting from fabricated golden chips or test vectors to trigger the HT, both of which are challenging to obtain.

In 2014, Intel introduced a method that leveraged transient power consumption and TR, capable of detecting HTs with power consumption as low as $0.05\mu W/m^2$ [31]. Inspired by this method, TR-based methods were proposed to 1) identify regions with HT via statistical analysis [38, 44], and 2) detect HTs with power proportion as small as 0.14% in AES circuits through spatial projection transformation [42, 43]. The major limitation of these TR-based methods is that they still rely on the golden chip. In response to this limitation, Tang et al. [41] proposed a method to compare the Active Areas (AA) restored from Thermal Radiation Maps (TRM) with that of the IC design, so that HTs inserted during fabrication can be detected. Figure 1 presents the core processing flow of TRMs. Following the de-noising stage, incremental TRMs are utilized to distinguish between logic and vacant regions through statistical analysis, generating the actual AA shape of the fabricated IC. By comparing this AA shape with the design data, extra AA regions (i.e., possible HTs) can be successfully detected. Evaluations in [41] indicate that as long as the HT is larger than one pixel ($15\mu m * 15\mu m$) of the thermal camera, successful detection can be achieved.

[*] Corresponding author (e-mail: yaowangeth@gmail.com).

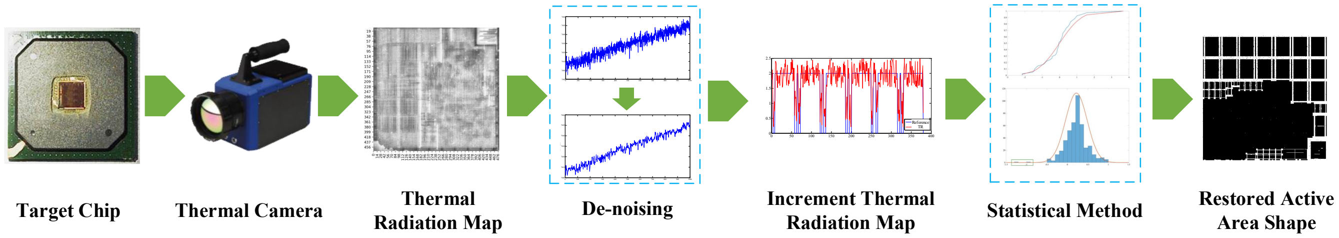
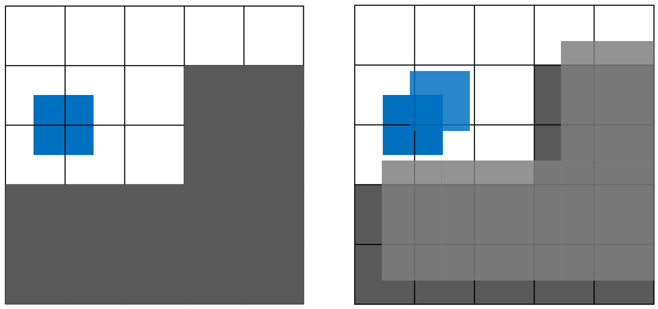


Figure 1: The core processing flow of TRM



(a) HT spreads into multiple pixels. The black and blue parts represent the HT and normal circuits, respectively.

(b) Mechanical vibration causes the image dithering. Dark and light colors represent scenes at different times.

Figure 2: Sub-pixel HTs and the mechanical vibration

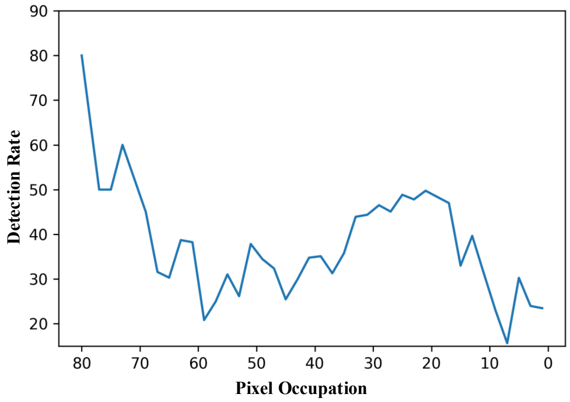


Figure 3: Performance deteriorates for the existing method

Despite the confirmed potential, existing TR-based methods are designed for the ideal HT that can fully occupy at least one pixel on the TRM. Such ideal HTs are not realistic due to the fact that HTs often extend across multiple pixels (shown in Figure 2(a)), occupying only sub-pixels in TRMs. Consequently, there is a significant reduction in occupancy within each pixel, leading to the ambiguous distinction of the TR for each sub-occupied pixel between logic or vacant areas, thereby rendering the current TR-based method impracticable. As illustrated in Figure 3, the performance of current methods deteriorates with decreasing pixel occupation. Notably, the detection rate drops below 50% when the HT occupies less than 70% of a pixel. Moreover, the noise resulting from the mechanical vibration of thermal cameras exacerbates the complexity of the issue. This disturbance induces image dithering in the TRMs sampled at different times (shown in Figure 2(b)), markedly diminishing the detection accuracy of sub-pixel HTs.

In order to improve the ability of TR-based methods, we propose a noise based pixel occupation enhancement (NICE) mechanism for HT detection. The primary observation of NICE is that noise can vary the pixel occupation of HTs while disrupting HT detection. As shown in Figure 4, a given HT position can result in different pixel occupation scenarios due to varying vibration directions. This observation motivates us to explore the optimal vibration direction, aiming to find out the largest pixel occupation scenario among the variations, which can prove instrumental in distinguishing HT pixels from vacant pixels, thereby significantly enhancing the accuracy of HT detection.

We implement NICE through a statistical manner, where TRMs are categorized into multiple sets based on the convergence of pixels dithering and the law of large numbers. This categorization is facilitated by estimating the dithering direction for each pixel, derived from the relationship between pixel occupation and TR increment. Subsequently, TRMs at each direction are processed independently for HT detection using Kolmogorov-Smirnov (K-S) statistic and the Pauta criterion. Finally, the detection results are statistically aggregated to derive the final conclusion. We carry out evaluations on a $0.13\mu m$ Digital Signal Processing (DSP) chip using a $15\mu m$ thermal camera to validate the efficacy of our approach.

The major contributions of this paper are:

1. We observe the impracticality inherent in the current TR-based method, designed for the ideal HT that fully occupies at least one pixel on the TRM. In real-world applications, where the HT typically occupies only a portion of one pixel, we demonstrate that the average detection rate of the current method falls below 45%.
2. We propose NICE, a noise based pixel occupation enhancement mechanism, aiming to improve the ability of TR-based methods. The key idea of NICE is to statistically find out the largest pixel occupations of the sub-pixel HT among variations caused by noise.

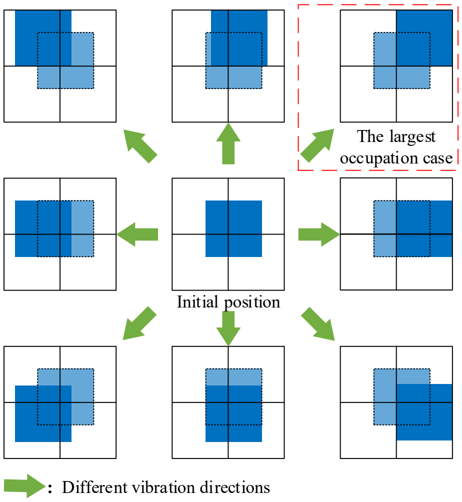


Figure 4: Different changes of pixel occupation caused by vibration directions

3. We demonstrate that NICE can improve the detection rate of sub-pixel HTs that exceeds the current TR-based method by more than 47%, reaching 91.81%, while maintaining a false alarm rate of less than 9%. Both metrics of NICE are comparable to the existing power-based and EMR-based methods, eliminating the need for the golden chip and test vectors.

2 Background

This section aims to provide a comprehensive understanding of the detection mechanism used in existing TR-based methods, along with an overview of the existing methods.

2.1 Threat Model

Adversaries inside the foundry possess complete access to the IC design layout, enabling them to carry out an HT attack during the IC fabrication. Various fabrication-time HTs have been studied recently, demonstrating the potential threat of this type of attack [26, 32, 33, 49]. These fabrication-time HTs can be divided into two categories: 1) additive HTs which implement HTs through insertion of additional logic cells to the original circuits; 2) modification HTs which modify the original logic cells to serve as HTs. Yang et al. [49] inserted analog HT for the attack, referred to as A2. Lin et al. [26] inserted Trojan side-channels (TSCs) which used side-channel leakage for HT implementations to leak exploitable information. Perez et al. [32, 33] demonstrated that attackers can insert a side-channel HT (SCT) into a finalized layout requiring minimal knowledge about the chip. There are also instances of modification HTs [5, 15]. However, such type of HTs remains largely unexplored [46] due to its tremendous effort [12], a-priori knowledge, and skill-dependent nature [48].

In our threat model, 1) we focus primarily on the additive HT, realized through the addition of transistors or gates [45, 46, 47], a more commonly employed approach in existing research [46]. This choice is practical as additive HTs are characterized by effective concealment and low implantation complexity. They minimally affect the original IC design, leveraging ample unused space within the IC for inserting additional gates and transistors [18, 41, 46, 48]. 2) We assume the victim cannot obtain a golden chip for HT detection [48], which is a reasonable assumption given the expensive and time-consuming nature of the reverse engineering process it entails. 3) We also assume the victim may lack test vectors capable of activating the HT. Detecting HTs poses inherent challenges due to the unknown function, structure, and location, leading to a lack of a-priori knowledge about their activation mechanism in real-world scenarios [17].

2.2 Physical Mechanism

The fundamental component of the IC is the metal-oxide-semiconductor (MOS) transistor. In its quiescent state, the MOS transistor's parasitic capacitance and intrinsic resistance convert electric energy into heat (Q), as elucidated by the specific heat capacity theory.

$$\Delta T = \frac{Q}{Cm}$$
$$Q = P_{leak} = i_{leak}V_{dd}$$

Where, i_{leak} is the leakage current, V_{dd} is the voltage, and C represents specific heat capacity. The temperature variation ΔT in each area of IC should be determined by the power consumption (P_{leak}) in that.

TR is a fundamental physical phenomenon exhibited by all objects at temperatures above absolute zero (-273.15^oC). Consequently, the incremental intensity of TR ($\Delta I(v, i_{leak})$) generated by the quiescent work of MOS transistors can be described by Planck law [6].

$$\begin{aligned}\Delta I(v, i_{leak}) &= \frac{2hv^3}{c^2}\frac{1}{e^{hv/kT}-1}\\ &= \frac{2hv^3}{c^2}\frac{1}{e^{hvCm/Ki_{leak}V_{dd}}-1}\end{aligned}$$

Where, v represents the frequency of TR, and h denotes the Planck constant.

The essence of TR-based detection lies in the impact of HTs on the TR within the infected region. When the HT is inserted into the vacant regions of the IC through additional circuits, it results in additional power consumption in the corresponding area. Consequently, by observing the power changes reflected in the TR signal, we can effectively identify these alterations in the TRMs of the IC and detect HTs.

More specifically, the $\Delta I(v, i_{leak})$ of the additional circuits will no longer be zero due to the presence of leakage current.

The detection and localization of inserted HTs can be achieved by verifying the incremental TR in the vacant regions of the target chip. Therefore, the larger the size of the HT, the more pronounced the change, facilitating easier detection.

Relevant studies [31, 41] have demonstrated numerous advantages of TR-based detection over traditional methods. The TR-based method offers non-contact detection with high resolution, making it faster and more cost-effective than destructive techniques [27, 28]. Additionally, TR-based methods exhibit relatively lower susceptibility to process variation [21, 29, 35], which is an inevitable deviation arising from the fabrication process of ICs. This is attributed to the fact that the $\Delta I(v, i_{leak})$ of the logic region is sufficiently large, rendering it hardly affected by the small variations in the manufacturing process.

$$\begin{aligned}\Delta I(v,R) &= \frac{2hv^3}{c^2}\frac{1}{e^{hvCm/Ki_{leak}^2 R(1-\Delta R)}-1}\\ &= \frac{2hv^3}{c^2}\frac{1}{(e^{\frac{1}{1-\Delta R}})^{hvCm/Ki_{leak}^2 R}-1}\end{aligned}$$

Where, R is the typical resistor under a certain process, ΔR is the deviation of R.

Obviously, $e^{\frac{1}{1-\Delta R}} \rightarrow \infty$ is a necessary condition for $\Delta I(v,R) \rightarrow 0$, because $2hv^3/c$ and $hvCmR/Ki_{leak}^2$ are larger than 0. In other words, the TR change caused by HTs cannot be merged by the process variation noises, only if the deviation of R approaches to 100%.

2.3 Existing Methods

In 2014, Intel [31] proposed that transient power consumption and TR can be utilized to detect HT. According to their research, the detection probability is greater than 50% when the power consumption of HTs is lower than $0.05\mu W/m^2$ with effective process variation mitigation. While showing promise, this method relies on stronger simulation tools, currently limiting its practical implementation to theoretical considerations.

To enhance detection performance, Chen et al. [9] proposed a method that focuses on enhancing the spatial detail of TRMs to effectively eliminate noise. This was achieved by combining adaptive filtering in the time domain and guided filtering in the space domain, resulting in the improvement of 6.14dB in the signal-to-noise ratio of TRMs. Additionally, drawing inspiration from the concept of spatial projection transformation, principal component analysis (PCA) was employed to identify the key factors within the TR information. This enabled the detection of HTs comprising fewer than 20 gates [43] or with power proportions as low as 0.14% in AES circuits [42]. However, it is important to note that these TR-based methods encounter significant limitation. They heavily rely on the availability of the golden-chip, which is challenging to obtain in practice. As a result, the deployment and practical implementation of these methods is hindered.

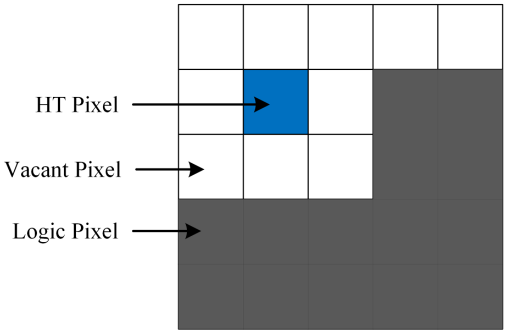


Figure 5: The HT fully occupies one pixel

In order to tackle the challenge posed by the requirement for a golden-chip, researchers have been exploring golden-chip-free methods for HT detection. Su et al. [40] proposed a method involving the pre-placement of Ring Oscillators (ROs) with different stages in the vulnerable areas of ICs. These ROs are designed to emit special TR signals. Consequently, these areas exhibit significant changes in the corresponding TRMs when an HT is inserted. However, a persistent challenge remains in that vulnerable areas within the IC are often difficult to fully occupy, rendering the practical application of this approach challenging, particularly in the context of ASICs.

Tang et al. [41] proposed another solution utilizing AA shapes obtained from the GDS II file as a reliable reference for golden-chip-free HT detection. In their suggested detection scenario, the pixels in TRMs are categorized into two groups: vacant pixels and fully occupied pixels, as illustrated in Figure 5. The fundamental concept of this method is that the additional HT pixel can be considered as the normal logic pixel in TRMs, with the distinction that its TR information differs from that of the vacant pixel. They employed statistical methods to generate the actual AA shape through the differentiation of logic and vacant regions, as illustrated in Figure 1. The procedure involves verifying the normal distribution of the TR of the logic region through a K-S test. Subsequently, the Pauta criterion is applied to identify significant differences between the TR of the vacant region and this established normal distribution. Their experiments demonstrated the effectiveness of this method in detecting HTs that occupy larger than one pixel ($15\mu m * 15\mu m$) on a $130nm$ DSP. However, despite the confirmed potential, current methods assume ideal HTs that fully occupy a pixel, making them far from practical use. Therefore, further exploration and development of TR-based countermeasures are still required.

3 Motivation

In this section, we explore the potential of utilizing mechanical vibrations to enhance sub-pixel HT detection, preceded by a detailed overview of the impact of sub-pixel HTs and mechanical vibrations on TR-based detection.

3.1 Sub-Pixel HTs Analysis

In this paper, our focus is on sub-pixel HTs distributed across multiple pixels, as depicted in Figure 2(a). This situation arises from the transparent nature of HTs when inserted into a chip, posing challenges in ensuring precise alignment of the HT boundaries with the pixels. In such cases, each infected pixel could be easily blurred as either a logic or vacant area.

This issue can be analyzed by considering the detection mechanism discussed in § 2.2. When the IC is operating statically, the incremental TR intensity $\Delta I(v, i_{leak})$ is correlated with the number of MOS transistors in the corresponding region. In other words, the TR distinction between sub-occupied and vacant pixels depends on the pixel occupation.

Figure 6 illustrates possible scenarios where a sub-occupied pixel might be misidentified as a vacant area. The TR distinction between fully occupied pixels and vacant ones is significant, as indicated by solid lines in the figure. However, some low-occupied pixels (represented by dashed lines) tend to closely resemble the vacant pixel (shown by the blue line). This indicates that certain sub-pixel HTs may evade detection using existing methods, as the occupancy in each pixel is significantly reduced. Moreover, the pink interval in the figure represents an area where HT may be missed, a concern that could be exacerbated by white noise. In this study, our aim is to investigate approaches to enhance the detection performance of sub-pixel HTs.

3.2 Noise Caused by Mechanical Vibration

The performance of sub-pixel HT detection is also influenced by the noise resulting from mechanical vibration. In many high-end thermal cameras, such as modern "Focal Plane Array" (FPA) cameras with cooled detectors, the cooling of the detector is achieved using a Stirling cooler [11]. The Stirling cooler consists of a compressor and an expander, where the gas is compressed to a high pressure in the compressor and then undergoes a pressure drop in the expander to generate refrigeration. The Stirling cooler is essential for maintaining a constant operating temperature of the detector, thereby improving the quality of TRMs. However, due to the nature of refrigeration, mechanical vibration becomes inevitable, and the noise level of the cooler may slightly increase over time. While the camera is in operation, the noise caused by the cooler can be both visually and audibly perceptible.

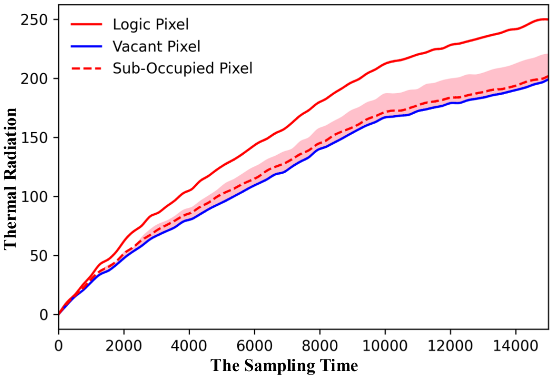


Figure 6: The TR distinction among fully occupied, sub-occupied and vacant pixels

Mechanical vibration refers to the oscillatory motion of a particle or a body around its equilibrium position [50]. Simple harmonic motion is an example of mechanical vibration that exhibits certain key characteristics such as frequency (ω) and amplitude (A). Simple harmonic motion is mathematically represented by the following formula:

$$x(t) = A\cos\omega t$$

In TR-based HT detection, the mechanical vibration of the thermal camera can introduce image dithering in the TRMs. Figure 7 illustrates how mechanical vibration can alter the TR distinction between sub-occupied and vacant pixels. The pink area in the figure indicates that vibrations in certain directions may enhance this TR distinction, while in other directions, the opposite effect occurs. However, the stack of different vibration directions complicates the differentiation between sub-occupied and vacant pixels, thereby posing challenges for accurately detecting sub-pixel HTs with existing methods.

Overall, it is crucial to investigate mechanisms for mitigating this noise in TR-based methods. Typically, noise reduction techniques can be implemented through hardware or software approaches. However, the vibration is coming from the internal detector, which is not tightly coupled to the lens or casing. This precludes possible hardware approaches to measure or approximate the vibration data, due to its attenuation and delay. A possible way to address the vibration is through image dithering elimination, such as the method involving the extraction and alignment of prominent features of the TRMs. However, it is widely recognized that noise can be suppressed to a certain extent but not entirely eliminated. Consequently, existing software methods may not effectively handle imperceptible dithering within the pixels, as depicted in Figure 2(b). This motivates us to take advantage of the vibration.

3.3 Exploiting the Potential of Noise

In our investigation, we made an intriguing observation regarding the influence of mechanical vibration on the position of HTs within TRMs. Mechanical vibration induces shifts in HT positions, broadening the range of potential position cases. As depicted by the dashed line in Figure 7, this heightened variability can enable HTs to reach the largest pixel occupation in specific directions, thereby augmenting the TR distinction with vacant areas compared to the vibration-free case (represented by purple lines). This motivates us to find out the vibration direction that can enhance this distinction, thereby improving the accuracy of sub-pixel HT detection.

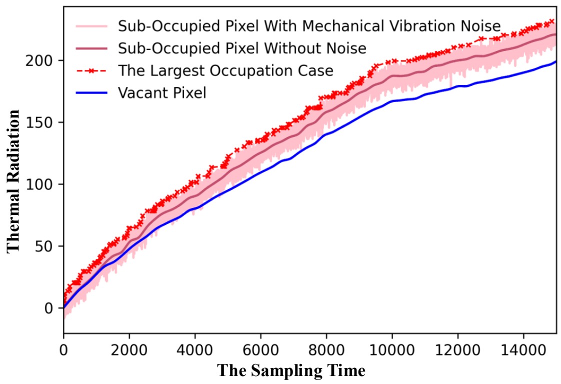


Figure 7: The effect of mechanical vibration on TR distinction

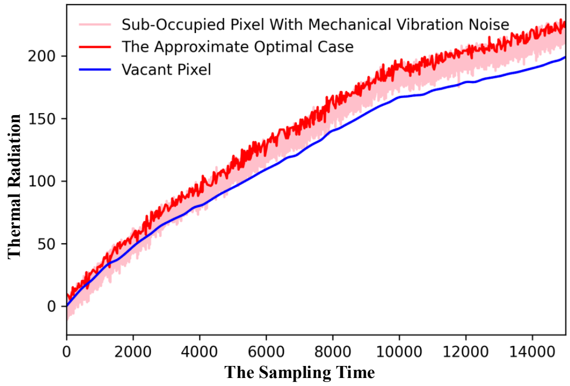


Figure 8: The TR distinction between sub-occupied and vacant pixels in the approximate optimal case after direction-based TRMs classification

However, this is a non-trivial task for a given HT, due to the presence of diverse biases in vibration directions among different HTs. Figure 9 shows that HTs with different initial positions may undergo entirely opposite changes in occupancy, even when subjected to the same vibration direction. Consequently, identifying a direction that uniformly optimizes detection across all HTs within the IC is impractical.

An alternative approach involves the application of a divide-and-conquer strategy, transforming the problem into a direction classification task for the TRMs dataset. The key insight of this approach lies in the imperative to identify any HT with corresponding bias in each direction, due to the transparency of HTs' distribution to the tester. More specifically, this strategy encompasses two considerations: 1) Direction-based detection involves identifying each HT at every individual direction, ensuring the inclusion of the approximate largest pixel occupation. 2) Furthermore, by aggregating the detection results from other directions characterized by a high pixel occupation, as illustrated in "The high pixel occupation case" within Figure 9, detection accuracy can be further enhanced. Clearly, direction-based detection also serves to alleviate the impact of noise caused by mechanical vibrations, reducing the difficulty of differentiation.

According to this strategy, the procedure entails identifying the dithering direction within the pixel at each sampling time, subsequently classifying TRMs based on the direction. Despite the challenges in precisely measuring shifts within pixels, hindering the accurate determination of their direction and magnitude, the observed correlation between variations in pixel occupation and incremental TR changes suggests an approximate method to identify the possible dithering directions of pixels. As illustrated in Figure 10, through the analysis of incremental TR for each pixel, we can determine whether the pixel occupation is exhibiting an increase (depicted in red) or a decrease (depicted in blue) relative to the previous sampling time, consequently revealing the corresponding dithering direction. By considering the possible directions of the majority of pixels within the TRM, we can determine the most likely dithering tendency (P_{trend}) of the entire image using statistical analysis methods, as depicted in the following formula.

$$p_{di} = \sum_{k=1}^{N} p_{di}^{k}$$

$$P_{trend} = \max_{1\leq i\leq n} \{p_{di}\}$$

Where, the probability p_{di} of different dithering directions (di) can be calculated as the sum of p_{di}^{k} for each pixel k. n and N respectively represent the number of vibration directions and pixels within the TRM.

In Figure 10, all four pixels have the possibility to move up. According to the law of large numbers, as the number of pixel samples increases, the determination of the dithering tendency approaches its true situation.

$$\lim_{N\to\infty} P\{|\frac{\mu_N}{N} - P_{trend}| \geq \varepsilon\} = 0$$

After classifying TRMs at each sampling time, we can approximate the largest occupation case for sub-pixel HTs with corresponding biases from the direction-classified TRMs, as shown in Figure 8. The accuracy of this approximation is determined by the number of directions and the associated step size, as depicted in the following formula.

$$\{TRMs_{opt}\} \subseteq \lim_{n\to\infty}(\{TRMs_{d1}\},\{TRMs_{d2}\},\ldots,\{TRMs_{dn}\})$$

Where, the $\{TRMs_{opt}\}$ represents the TRMs set with the largest occupation case for sub-pixel HTs and the $\{TRMs_{dn}\}$ represents the TRMs set at the dithering directions dn.

4 Noise Based Pixel Occupation Enhancement for HT Detection

In this section, we introduce the NICE mechanism to enhance the accuracy for sub-pixel HT detection, including the detection framework and core detection algorithms.

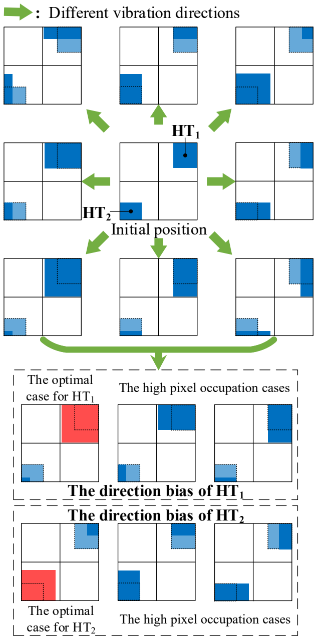

: Different vibration directions
HT$_1$
HT$_2$
Initial position
The optimal case for HT$_1$
The high pixel occupation cases
The direction bias of HT$_1$
The direction bias of HT$_2$
The optimal case for HT$_2$
The high pixel occupation cases

Figure 9: The biases of HTs towards the direction of vibration

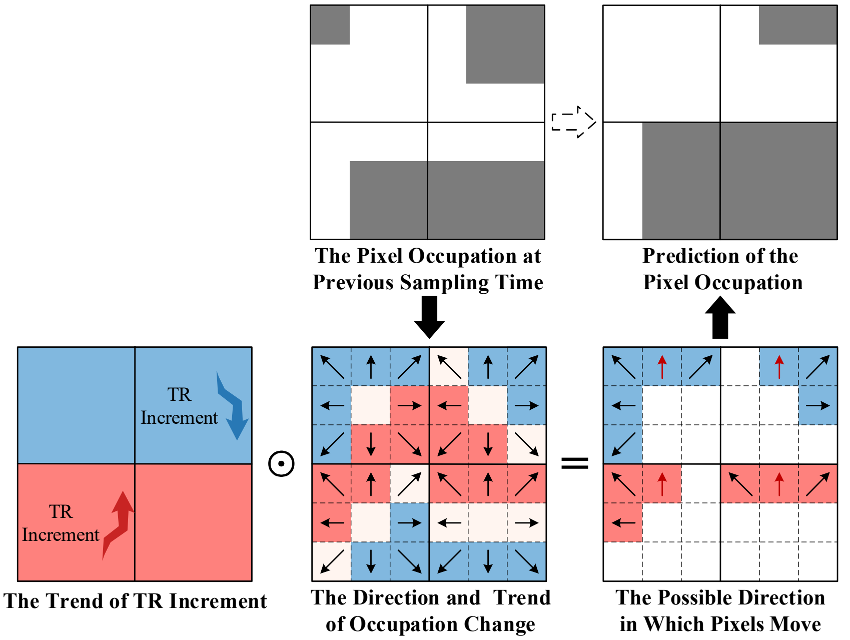

The Pixel Occupation at Previous Sampling Time
Prediction of the Pixel Occupation
TR Increment
TR Increment
⊙
=
The Trend of TR Increment
The Direction and Trend of Occupation Change
The Possible Direction in Which Pixels Move

Figure 10: The tendency determination of image dithering

4.1 Detection Framework

Inspired by the analysis presented in § 3.3, we propose a novel HT detection framework called NICE. The primary objective of NICE is to approximate the HT's largest pixel occupations among the variations caused by noise, achieved through the application of direction-based classification. Figure 12 shows the detection framework employed by NICE, which comprises two major components: TRMs classification and detection result aggregation.

Specifically, TRMs are categorized into several sets based on the direction of image dithering, and corresponding golden references are extracted from the design data of the target IC. Subsequently, each set of TRMs is processed independently for HT detection using the K-S statistic and the Pauta criterion. By comparing with golden references in the corresponding direction, multiple results indicating possible HT pixels are obtained. Finally, the detection results are statistically aggregated to produce the final result, thereby significantly enhancing the detection accuracy of sub-pixel HTs.

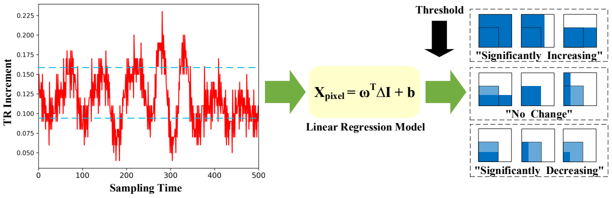


Figure 11: The trend determination based on TR increment

4.2 Direction-based TRMs Classification

The primary principle behind TRMs classification involves the convergence of pixel dithering, where the dithering trend of each pixel can be estimated by the variation in TR increment. A detailed implementation of the algorithm is provided in Appendix A.1.

1) Estimating possible dithering directions for each pixel. For a single pixel, its TR increment fluctuates over time. The magnitude of change in the TR increment depends on whether the pixel occupation increases or decreases. To ascertain the trend of pixel occupation, we have formulated a backward linear regression model to delineate the relationship between pixel occupation and TR increment, as shown in the following equation.

$$\mathbf{X}_{\mathbf{pixel}} = \omega^T \Delta\mathbf{I} + b$$

Where, $\Delta\mathbf{I}$ is TR increment, $\mathbf{X}_{\mathbf{pixel}}$ is pixel occupation.

Determining trends of pixel occupation over time. The TR increment data of all pixels is employed to facilitate the fitting of model parameters, with the TR increment of each pixel evenly corresponding to approximate occupation values based on the occupation variation interval specific to each pixel. This interval is calculated from the golden references containing occupation information for each pixel. Subsequently, the model utilizes the TR increment of each pixel to reevaluate its occupation at every sampling time, enabling the determination of whether the pixel exhibits an increase or decrease in occupied regions compared to previous time.

Estimating possible dithering directions. Pixel dithering

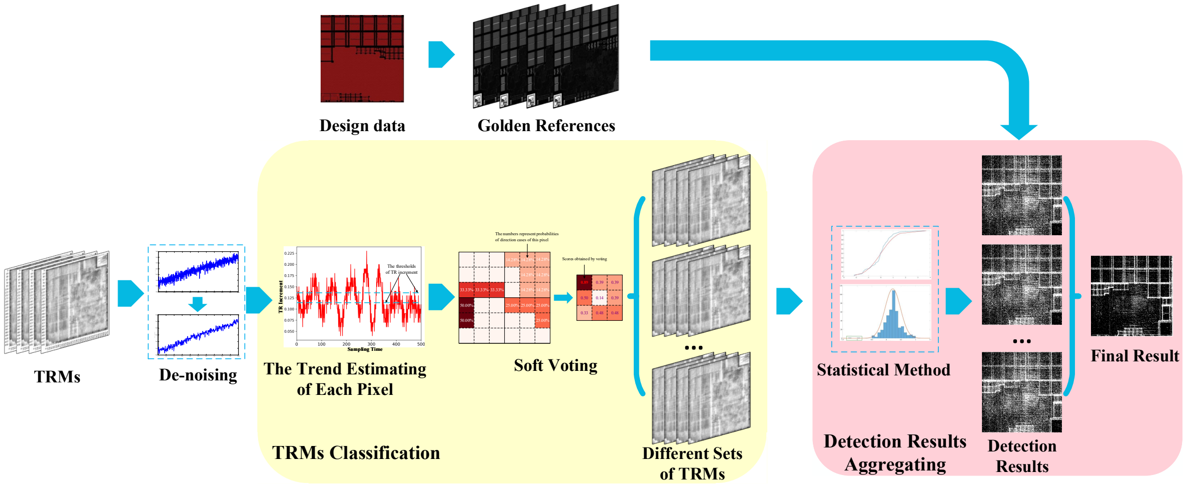


Figure 12: The detection framework for NICE

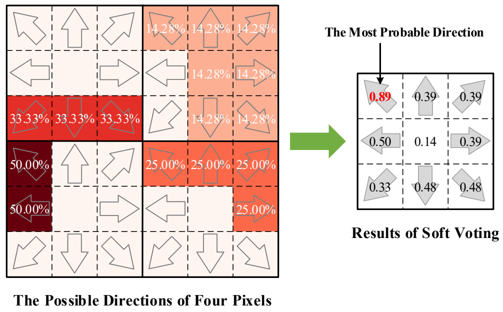


Figure 13: TRMs classification based on soft voting. The percentages represent probabilities of the pixel dithering in different directions.

is estimated through approximate matching, given the challenge of accurately determining the dithering direction and position based on occupation changes, as depicted in Figure 11. The occupation change is classified into three categories using a threshold: "significantly increasing", "significantly decreasing", or "no change". These categories correspond to different dithering direction sets, typically covering nine fundamental directions. In this way, multiple possible dithering directions for the pixel at each time are identified, enhancing the error tolerance of the classification. Moreover, we set the threshold on the occupation change instead of the TR increment, enabling its application to the detection of various ICs.

2) Classifying TRMs into different direction sets. The dithering directions of the entire TRM can be ascertained by the convergence of all pixels at the same sampling time. It is well known that image dithering affects every pixel equally, so the correct dithering direction can be obtained by the statistical analysis of classification results of all pixels according to the law of large numbers. Figure 13 shows the Soft Voting method employed for the global statistical analysis of pixels exhibiting multiple potential dithering directions. We assign equal weights to all possible directions of the pixel, and calculate the probabilities of various pixel directions, facilitating the determination of the most probable dithering trend through a weighted average. The equation for Soft Voting is provided in Appendix A.1. As a result, the TRM at each sampling time can be classified into different direction sets, matching the corresponding golden reference.

4.3 HT Detecting and Results Aggregating

Conducting HT detection by traversing all directions. The TRMs set in each direction is processed independently to distinguish between logic and vacant regions through statistical analysis. Specifically, the statistical parameters of logic regions are extracted from the TR increment data, after verifying whether this data follows a normal distribution using the K-S statistic. The *mean* and *standard deviation* of the distribution are used to identify the vacant regions in the TRMs. According to the Pauta criterion, when the TR increment of a certain pixel deviates from the *mean* of logic regions by more than three *standard deviations*, it can be considered a vacant pixel with significant probability. By comparing these results with the golden references, several possible sets of HT pixels can be identified.

Aggregating results for possible HT pixels. Multiple results are obtained after traversing detection in all directions, facilitating the accurate detection of possible HT pixels. The target HT is hard to be detected in every direction, as sub-occupied pixels can be easily misclassified as either logic or vacant areas. Typically, sub-pixel HTs are detected in only a few results, especially in cases with the largest pixel occupation. In such scenarios, suspicious pixels are considered as potentially indicating the presence of an HT inserted during fabrication, if an extra AA is detected.

However, TRMs classification is challenging to ensure complete accuracy. A few TRMs may be misclassified due to white noise or unreasonable classification thresholds, which may result in a false alarm. To mitigate this, the information from suspicious pixels in other directions can be utilized to further determine whether the result should be corrected. Specifically, if an extra AA is detected in a certain direction while the suspicious pixel corresponds to a logic region in most other references, the result should be corrected, thereby reducing the false alarm. More details on the HT detection are presented in Appendix A.2.

4.4 NICE System Implementation

To implement the proposed method, we construct the corresponding software and hardware systems. Figure 14 shows our hardware acquisition system which consists of three major parts: a thermal camera, a support and isolation platform, and a low-noise module. The thermal camera works continuously during the data acquisition stage to capture high-quality TRMs, while the support platform ensures that the target IC can be positioned on the focal plane of the thermal camera, and is insulated from external vibrations using an air floating system. The heat conduction and environment noises can be mitigated during the acquisition through our low noise module. The heat dissipation equipment can form a low temperature environment in the chamber specially designed for shading, which can spray the low temperature nitrogen into the chamber from a dewar bottle. Then, the TRMs processing is completed by the computer equipped with an Intel(R) Core(TM) i7-9700 CPU, operating Python-based software.

This modular design of NICE system facilitates integration into real-world post-silicon HT detection scenarios and enables adaptability for detecting various ICs. The isolation platform and the low-noise module effectively mitigate the environmental interference, making NICE suitable for deployment across various environments. The support platform allows for adjustments to accommodate different thermal cameras through simple replacement of connection flanges, ensuring compatibility with different chip technologies. Moreover, the NICE system is not affected by larger ICs, as TRMs can be acquired and processed region-by-region for a larger IC using the X-Y Horizontal platform. While this may increase processing time, parallel computation offers a viable solution.

5 Experiment and Evaluation

In this section, we conduct experiments to evaluate the performance of NICE and further investigate the impacting factors for the effectiveness of NICE. The performance evaluation covers the following five vectors:

- **Evaluation for sub-pixel HT detection** (§ 5.2). We evaluate the performance of NICE in comparison with existing methods for sub-pixel HTs detection.
- **Performance analysis across different HTs** (§ 5.3). We analyze the detailed detection performance for various HT samples with different sizes.
- **Sensitivity to the number of TRMs samples** (§ 5.4). We assess the effectiveness of NICE based on several groups of TRMs with different sample sizes.
- **Sensitivity to classification thresholds** (§ 5.5). We explore six classification thresholds to guide the selection of a classification threshold.
- **Sensitivity to the white noise** (§ 5.6). We analyze the impact of different white noise levels on the detection performance of NICE.

5.1 Experiment Scheme

In this experiment, the target IC is a high-end universal DSP under the $0.13\mu m$ process, which is depackaged before the test. The TRMs acquisition system is equipped with a thermal camera with a spatial resolution of $15\mu m * 15\mu m$ and Noise Equivalent Temperature Difference (NETD) of $30mK$.

Figure 15 presents our experiment scheme, designed to assess the constraints of the previous method and the effectiveness of the NICE mechanism. The proposed scheme consists of three major components: an equivalently approach to implement "HT", generating actual AA shapes, and statistical analysis.

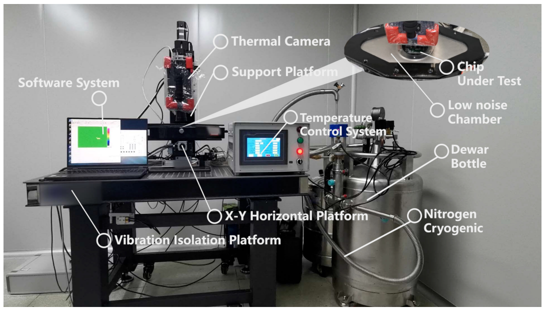


Figure 14: The platform of NICE

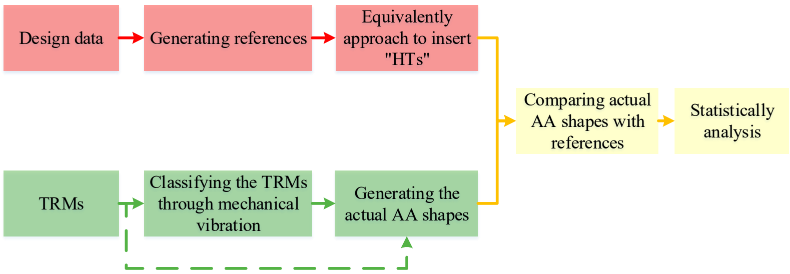

Design data
Generating references
Equivalently approach to insert "HTs"
TRMs
Classifying the TRMs through mechanical vibration
Generating the actual AA shapes
Comparing actual AA shapes with references
Statistically analysis

Figure 15: Experiment scheme in this paper

1) "HT" insertion: The equivalently approach is employed to implement "HT" which is challenging to implement in large numbers within the real IC. This method is similar with [41] and [34], and extend both HT size and insertion scopes. Upon extracting the golden reference from the IC design, we randomly remove certain logic regions of the IC layout. These regions appear as vacant in the golden references but remain logic in the actual IC. In our experiment, all logic pixels can be regarded as "HT", and each pixel should be iteratively detected. This process provides sufficient "HT" samples, exhibiting varying sizes and arbitrary locations within the IC.

2) Actual AA shapes generation: In this experiment, 15000 consecutive sample time TRMs are selected to generate AA shapes, after suppressing the white noise with high frequency through an eight-level wavelet filtering with the "Sym6" base. Following our NICE mechanism, TRMs are categorized into nine direction sets, as illustrated in Figure 22. Consequently, nine different AA shapes will be obtained by the K-S statistic and the Pauta criterion, then results are aggregated to form the final result. However, the classification process is omitted for the previous method, which directly generates the AA shape through the subsequent processing of unclassified TRMs affected by mechanical vibrations.

3) Performance evaluation and sensitivity analysis: The statistical method is utilized to analyze detection results, encompassing performance evaluation and sensitivity analysis. The performance of the previous method and NICE is compared by the detection rate and the false alarm rate for detecting all "HT" samples, which respectively represent the recall rate of extra AA pixels and the error rate in the vacant region of golden reference. Additionally, the robustness of NICE is assessed through the adjustment of the number of TRMs, the classification threshold, and the level of white noise.

5.2 Evaluation for Sub-pixel HT Detection

Overall performance. Our results reveal a significant enhancement in performance compared to previous methods. We successfully reproduce the results of [41], using the same process IC and a similar thermal camera with the same resolution. The experimental result shows that the detection rate of this method can reach up to 99% for the "HT" that fully occupies one pixel. However, there is a substantial decrease in accuracy when this method is applied to detect sub-pixel "HTs". In contrast, as presented in Table 1, NICE can detect sub-pixel "HT" samples with a detection rate of up to 91.82% and a false alarm rate below 9%, representing a performance improvement of more than 47% over the previous method.

We can visually compare the performance between them in Figure 16, where the logic and vacant regions are respectively depicted by black and white pixels. Evidently, NICE exhibits superior capability in recovering the AA shape of the IC.

Detailed analysis. Figure 17 provides a detailed illustration of the performance improvement achieved by NICE, presenting the detection results for different pixel occupation cases. It can be observed that when the "HT" occupies more than 70% of a pixel, NICE can achieve great detection accuracy with a detection rate reaching up to 97.5%. This can be attributed to the NICE can capture samples near the detection threshold when they are slightly affected by vibrations. Moreover, NICE also demonstrates the substantial enhancement when "HTs" occupy less than 10% of a pixel, despite they may be notably challenging to detect. This can be attributed to the ability to approximate the largest pixel occupation case among the variations, even when "HT" is minute in the current scenario.

The case of the occupation within the range of 10%~60% of a pixel presents the most complex scenario, imposing limitations on the overall performance of NICE. However, NICE improves the detection rate of these samples by over 40%.

Impact of the variability of pixel occupation. To delve deeper into the mechanics of NICE, the statistical data for the occupation change of all "HT" samples can be leveraged to illustrate the functioning of NICE. We counted the proportion of all pixels that changed by more than 10% and analyzed the detection results. Figure 18 depicts the statistical results of the occupation change aligned with the observed trend of NICE performance enhancement.

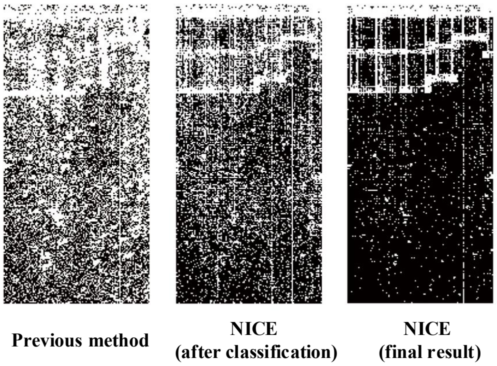


Figure 16: The AA shapes restored from parts of the IC

Table 1: Detection results of NICE and the previous method

	Previous method	NICE (single set)	NICE (final result)						
			Thresholds:	1%	2.5%	5%	10%	15%	20%
Detection rate	44.36%	67.48%		45.26%	87.77%	91.81%	90.53%	83.20%	84.30%
False alarm rate	15.90%	13.42%		12.56%	16.18%	8.44%	9.85%	13.13%	10.09%

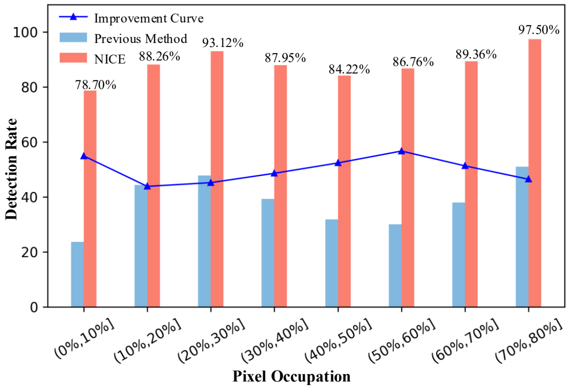


Figure 17: The detection rate of NICE and the previous method

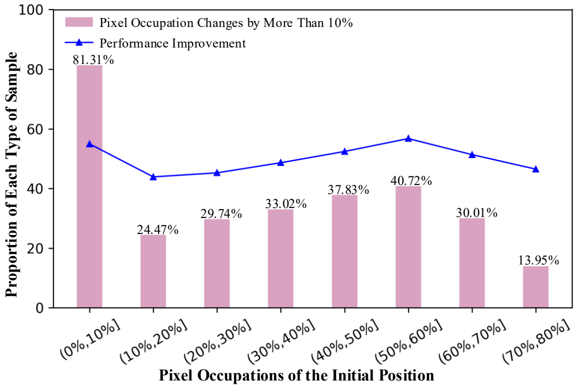


Figure 18: The statistics of the occupation change for "HTs"

The statistical result for "$\leqslant 10\%$" samples with significant occupation changes is 81.31%, closely matching the detection rate of the corresponding samples in Figure 17 (78.70%). This observation underscores the fundamental efficacy of NICE in detecting a sub-pixel HT that spreads across multiple pixels.

The improved performance for "10%~60%" samples can be attributed to the effectiveness of TRMs classification. This approach not only enhances the detection performance of a single set of TRMs but also consolidates multiple detection results from the same sample. As outlined in Table 1, the results of detecting a single set of TRMs indicate a maximum performance increase of up to 23.12%.

The detection rate of the "more than 60%" samples is the highest (more than 93%), while their pixel occupations change little. The reason is that the stable TR change makes detection easier for NICE, when HT is large enough.

5.3 Performance Across Different HTs

Overview. Unlike other side-channel detection methods sensitive to HT types, TR-based methods are mainly influenced by HT size, as it affects the TR distinction between sub-occupied and vacant pixels. Therefore, we conduct experiments with a range of HT sizes to evaluate the detection capabilities of NICE, including HTs that occupy multiple pixels and those smaller than a single pixel. Figure 19 provides detection results for HTs with different sizes. We can observe that NICE can effectively identify HTs occupying more than 70% of a pixel with a 98% detection rate, whereas the previous method only performs well when HTs are more than two pixels.

Detailed analysis. It can also be observed that as HT size increases, the detection rate of NICE converges rapidly. This is because larger HTs are more likely to achieve higher occupation within certain pixels through the vibration, making them easier to identify. In contrast, the previous method exhibits limited improvement in detection rates even when the HT size exceeds a pixel. This limitation arises because HTs typically partially occupy multiple pixels. Overall, the result indicates that NICE can push the detection boundary of TR-based methods from more than two pixels to only 0.7 pixels.

5.4 Sensitivity to the Number of TRMs

Overview. In a practical detection scenario, the smaller TRM samples may be acquired for HT detection to further control the cost and overhead. Therefore, we try to analyze the impact of the number of TRMs on NICE. Figure 20 presents the performance trend for detecting HT with decreasing TRM samples. We can observe that NICE achieves steady performance, with a detection rate of more than 91% even when the number of samples is decreased to 50% (7500) of TRM samples in § 5.2. However, the performance will deteriorate when the sample size is reduced to a third (5000).

Detailed analysis. This result demonstrates that the NICE is effective in the smaller-samples scenario with a high detection rate, which indicates its robustness. However, NICE is implemented based on statistical methods, which can be completely effective when the TR growth of the IC is sufficient. Therefore, its performance inevitably deteriorates with decreasing the number of TRMs, when parts of the TR information are lost. To address this, we further conduct experiments based on the interval sampling method to simulate the reduction of sampling frequency, which shows a more stable performance in the red curves in Figure 20.

False alarm. We can also observe that the false alarm rate is more affected by the sample reduction than the detection rate. Fortunately, interval sampling is also effective in reducing false alarms. Therefore, sufficient TRM samples are recommended for NICE. Even in certain cases where the sample size must be reduced, it is more reasonable to reduce the sampling frequency or interval sampling in sufficient TRMs.

5.5 Sensitivity to Classification Thresholds

Overview. The sensitivity of NICE to different classification thresholds is explored to guide the selection of optimal

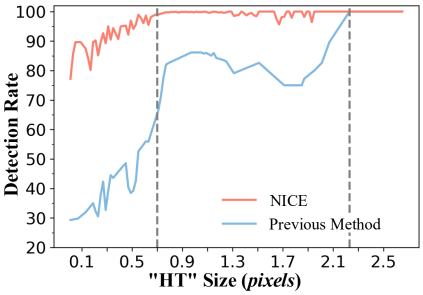


Figure 19: Detection results for HTs with different sizes

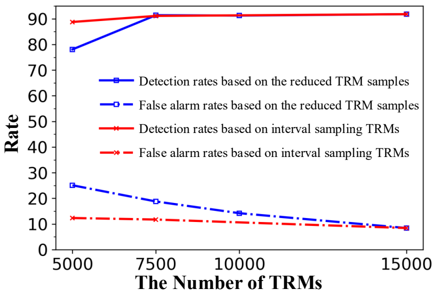


Figure 20: Detection performance based on different TRM samples

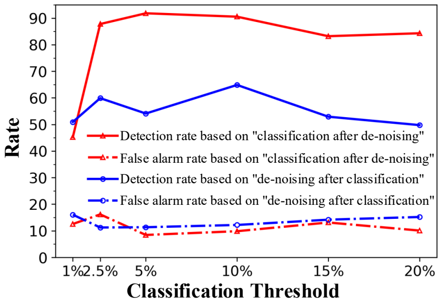


Figure 21: The detection performance for sensitivity analysis of classification thresholds and white noise

Table 2: Performance based on different noise levels

Noise leve (dB)	7.80	-5.57	-12.85	-20.33
Detection rate	91.81%	91.07%	87.18%	64.86%
False alarm rate	8.44%	9.31%	10.09%	12.20%

thresholds. We classify TRMs based on occupation change thresholds of more than 1%, 2.5%, 5%, 10%, 15% and 20%. The detection results in Table 1 indicate that a threshold of 5% provides the best performance, as also illustrated by the curves in Figure 21. Figure 22(a) presents the classification results of TRMs based on the 5% threshold, indicating the direction determination of image dithering within the pixel. It can be observed that the dithering tends to be more pronounced in the horizontal direction, which aligns with the structural characteristics of our system.

Detailed analysis. The reasonable threshold can be further verified by the results shown in Figures 22(b) and 24. As illustrated in Figure 22(b), classification with thresholds of 1% and 2.5% results in fewer categories compared to other thresholds, because the tiny occupation changes can not be recognized by the thermal camera. Furthermore, Figure 24 shows the dithering over time in the horizontal direction, demonstrating that the frequency of image dithering is faster with the decrease of threshold. The result indicates that if the threshold is too large, the occupation changes caused by different dithering directions would be confused, which reduces the precision of classification.

False alarm. We can observe in Figure 21 that the false alarm rate can be affected by the unreasonable threshold. Comparing among the cases with thresholds of 2.5%, 5%, and 10%, where their detection rates are close, the unreasonable threshold (2.5%) results in a significant increase in the false alarm rate. Fortunately, the law of large numbers ensures the accuracy of our method, as long as the threshold value is not excessively large or small, such as from 5% to 10% in this experiment, which is robust enough for different thresholds.

5.6 Sensitivity to White Noise

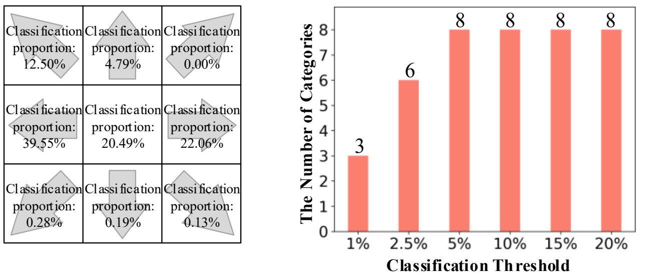
Overview. To investigate the potential impact of white noise on the accuracy of dithering trend determination, we conduct an experiment comparing two different TRMs processing flows: "classification after de-noising" and "de-noising after classification". As shown in Figure 23, the presence of white noise may limit the classification of TRMs to three dithering directions, even resulting in inconsistencies (as depicted in Figure 23(a)) with the correct dithering trend. Figure 21 illustrates that the combined effect of white noise and threshold settings can significantly reduce the detection performance of NICE. Nevertheless, with a well-selected threshold of 10%, NICE also outperforms previous methods, achieving a detection rate of 64.86% and a false alarm rate of 12.20%.

Detailed analysis. To evaluate the impact of different noise levels on detection performance, we gradually reduce the noise levels in the initial data, estimating these levels using the Signal-to-Noise Ratio calculation in Python combined with the Kaiser window. As shown in Table 2, while white noise can affect the performance of NICE, its impact is manageable. For example, utilizing the simple four-layer wavelet filtering, the detection rate can reach 87%, and with wavelet filtering of more than six layers, the detection rate can increase to over 91%. This suggests that NICE demonstrates a certain robustness against white noise.

False alarm. The experimental results indicate that white noise can lead to TRM classification errors, resulting in the increased false alarm rate. One effective solution is to eliminate white noise before TRM classification. Moreover, according to our experiments, as the effects of classification thresholds and white noise are combined, a well-selected threshold can suppress the false alarms caused by white noise.

6 Discussion

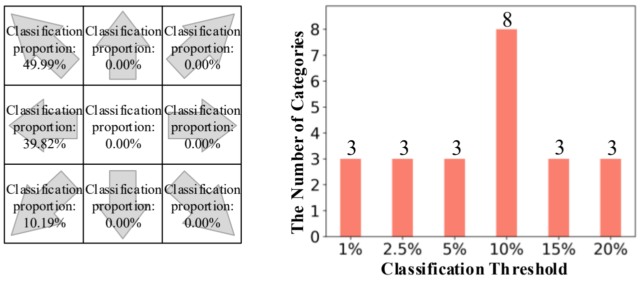
Investigating the potential of thermal cameras across chip technologies. As semiconductor technology continues to scale down, the detection of sub-pixel HTs becomes increas-



(a) The result of TRM classification with thresholds of 5%

(b) The number of classification categories with different thresholds

Figure 22: TRMs classification results with various thresholds



(a) The result of TRM classification with thresholds of 20%

(b) The number of classification categories with different thresholds

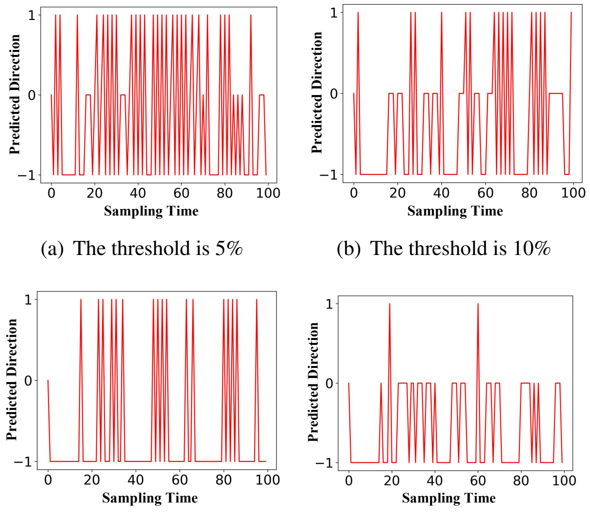
Figure 23: TRMs classification results without noise reduction

ingly challenging. A potential strategy to address this issue is to employ thermal cameras with ultra-high resolution. However, while more advanced cameras offer higher capabilities, there are technical limitations that restrict the continuous improvement of resolution. Moreover, using the high-resolution camera for HT detection in relatively large technology nodes incurs increased economic costs and computational overhead. NICE focuses on enhancing the capabilities of given thermal cameras, pushing the detection boundary from more than two pixels to only 0.7 pixels, as shown in Table 3.

Taking the proposed hardware system as an example, NICE can detect HTs with 40 gates, compared to the original 110 gates, under a 130nm process using a 15μm thermal camera. As shown in Table 4, most of HTs in Trust-Hub are smaller than 100 gates. Moreover, the A2 HT implemented under the 130nm process consists of around 170 standard MOS transistors, which corresponds to approximately 43 gates, based on the circuit structure in [13]. In these scenarios, existing TR-based methods require a 12μm thermal camera to achieve detection under 130nm process, while NICE can cover these HTs with a 15μm thermal camera. Overall, NICE enables a more flexible and cost-effective selection of thermal cameras.

Comparing detection performance of NICE with other techniques. Table 5 summarizes the detection performance and characteristics of NICE alongside other side-channel techniques, which indicates that the performance of NICE reaches the average level of other side-channel methods, and even shows slight improvement as the IC size increases. Although some methods claim to achieve extremely high detection rates, they are often constrained by stringent conditions, requiring either a golden chip or specific testing vectors to trigger the HT. In contrast, NICE stands closer to practical use due to its high performance and reasonable detection conditions.

Limitation and evasion techniques. There are still some limitations in the current implementation of NICE: *1) Smaller HTs.* Adversaries can potentially evade our detection by designing increasingly smaller HTs. In fact, smaller HTs present significant challenges to all existing detection methods. We anticipate that using the most advanced thermal camera (1.5μm), NICE can extend the detection capabilities to identify HTs consisting of approximately 26 gates and 35 gates under 7nm and 5nm chip technologies, respectively [1]. A review of recent HT designs [8, 13, 14, 23, 25, 26, 46] and those documented in Trust-Hub (as illustrated in Table 4) reveals that existing HTs exceed 35 gates, typically around 100 gates. This trend likely stems from the consideration that extremely small HTs may restrict their malicious functionality and compromise their stealth [46].



(a) The threshold is 5%

(b) The threshold is 10%

(c) The threshold is 15%

(d) The threshold is 20%

Figure 24: The direction determination of the first 100 sampling times in the horizontal direction

2) Modification HTs. As described in our threat model, NICE currently does not support the detection of modification HTs. These HTs can be created by making slight alterations to the dopant [5] or parameters [15] of transistors without introducing additional logic. However, this type of HT remains largely unexplored [46] due to the significant effort [12], prior knowledge, and specialized skills required for its design [48]. Currently, one feasible detection technique for these HTs is

Table 3: The capabilities of thermal cameras across chip technologies

Camera resolution	Chip technology	Number of equivalent gates	Detection boundary Previous	NICE
15µm*15µm	130nm	≈ 55	⩾110	⩾39 ✓
	65nm	≈ 117	⩾234	⩾82 ✓
	40nm	≈ 239	⩾478	⩾167
	28nm	≈ 446	⩾892	⩾312
	14nm	≈ 868	⩾1736	⩾608
12µm*12µm	130nm	≈ 36	⩾72 ✓	⩾25 ✓
	65nm	≈ 75	⩾150	⩾52 ✓
	40nm	≈ 153	⩾306	⩾107 ✓
	28nm	≈ 286	⩾572	⩾200
	14nm	≈ 556	⩾1112	⩾389
5µm*5µm	130nm	≈ 6	⩾12 ✓	⩾4 ✓
	65nm	≈ 13	⩾26 ✓	⩾9 ✓
	40nm	≈ 27	⩾54 ✓	⩾19 ✓
	28nm	≈ 50	⩾100	⩾35 ✓
	14nm	≈ 96	⩾192	⩾67 ✓

Table 4: The size of some existing HTs

Works	HTs	Number of equivalent gates
Siddik et al. [8]	PUF-based HT	≈ 125
Deng et al. [13]	A2(130nm)	≈ 43
Dharsee et al. [14]	Jinn	125
Jain et al. [23]	TAAL(32nm)	≈ 189
Kumar et al. [25]	edAttack(15nm)	≈ 37
Lin et al. [26]	TSC	⩽ 100
Trippel et al. [46]	A2(45nm)	91
	Key Leak(45nm)	187
Trust-Hub	AES-T400	≈ 90
	AES-T600	≈ 100
	AES-T700	≈ 80
	AES-T800	≈ 230
	AES-T900	≈ 840
	AES-T1000	≈ 80
	AES-T1100	≈ 80
	AES-T1200	≈ 840
	AES-T2000	≈ 80

Table 5: Summary of HT detection based on side-channel techniques

Physical information	Works	Detection rate	Golden chip	Testing vector	Resolution
Thermal radiation	This paper	91.82%	Not Required	Not Required	Pixel level
	Tang et al. [41]	44.36%			
Electromagnetic radiation	Ngo et al. [30]	83%	Required	Required	Sub-region of IC
	Chen et al. [10]	88%	Required		
	He et al. [19]	89.2%	Not Required		
Power	Hu et al. [22]	91%	Required	Required	Global IC

reverse engineering, which can achieve accurate detection but at the cost of significant economic and time resources.

3) HTs inserted in filler areas. There are alternative methods for implementing additive HTs, such as replacing filler cells with logic cells through the Engineering Change Order (ECO) flow [32, 33]. Filler cells can be divided into two types: (a) traditional fillers with no transistors, which resemble vacant areas on TRMs, and (b) active fillers in advanced technologies containing transistors. HTs implanted in active fillers may evade existing TR-based detection techniques. However, our preliminary exploration suggests there are some distinguishable features between logic areas and these fillers in the TRM. We believe that NICE can be enhanced by incorporating extension algorithms to further distinguish between active fillers and logic areas, which remains part of our future work.

Overall, the attacker and defender are engaged in an ongoing game of strategy and evolution. Despite the possibility of existing TR-based detection being evaded, we believe that enhancing sub-pixel HT detection by exploiting the potential of noise can provide valuable insights into post-silicon HT detection. Besides, recent studies have developed Reinforcement Learning (RL)-based HT insertion methods [16, 36, 37], which increase the difficulty of HT activation and show great potential to evade detection techniques that require HT activation. Given the activation-free characteristic, NICE is complementary to these activation-based detection methods and can be combined to enhance the overall HT detection capability.

7 Conclusion

In this paper, we observed that the noise caused by mechanical vibration can vary the pixel occupation of the HT, which can help us address the challenge for sub-pixel HT detection. To this end, we proposed NICE, a noise based pixel occupation enhancement mechanism. The experimental results demonstrate that NICE can greatly improve the performance for detecting the sub-pixel HTs compared with existing TR-based methods. Our results provide a direction for noise mitigation but also suggest how to use noise for performance improvement of TR-based methods.

Acknowledgments

We thank our anonymous reviewers and shepherd for their insightful feedback. This work was supported in part by National Natural Science Foundation of China (62272477), Natural Science Foundation of Hunan Province (2022JJ10066), and Key Laboratory of Advanced Microprocessor Chips and Systems. We also acknowledge the support from Yantai DC

Base Co., Ltd., whose contributions were instrumental in the experimental equipment and scheme development.

References

[1] 5 nm lithography process - WikiChip. https://en.wikichip.org/wiki/5_nm_lithography_process, 2022.

[2] Dakshi Agrawal, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. Trojan Detection using IC Fingerprinting. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 296–310, Berkeley, CA, 2007. IEEE.

[3] Josep Balasch, Benedikt Gierlichs, and Ingrid Verbauwhede. Electromagnetic circuit fingerprints for Hardware Trojan detection. In *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, pages 246–251, Dresden, Germany, 2015. IEEE.

[4] M. Banga and M. S. Hsiao. A region based approach for the identification of hardware Trojans. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 40–47, 2008.

[5] Georg T. Becker, Francesco Regazzoni, Christof Paar, and Wayne P. Burleson. Stealthy dopant-level hardware Trojans: Extended version. *Journal of Cryptographic Engineering*, 4(1):19–31, 2014.

[6] THEODORE L. BERGMAN, ADRIENNE S. LAVINE, FRANK P. INCROPERA, and DAVID P. DEWITT. *Fundamentals of Heat and Mass Transfer*. JOHN WILEY & SONS, Hoboken, NJ, seventh edition edition, 2011.

[7] Shivam Bhasin and Francesco Regazzoni. A survey on hardware trojan detection techniques. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2021–2024, Lisbon, Portugal, 2015. IEEE.

[8] Md. Abu Bokor Siddik and Sk Hasibul Alam. PUF-based Hardware Trojan: Design and Novel Attack on Encryption Circuit. In *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–5, Chittagong, Bangladesh, 2023. IEEE.

[9] Zhao Chen, Heming Wang, Yongkang Tang, and Ying Zhang. Detail enhancement analysis based hardware Trojan detection using thermal maps. *Computer Applications In Engineering Education*, 54(16):86–92, 2018.

[10] Zhe Chen, Shize Guo, Jian Wang, Yubai Li, and Zhonghai Lu. Toward FPGA Security in IoT: A New Detection Technique for Hardware Trojans. *IEEE Internet of Things Journal*, 6(4):7061–7068, 2019.

[11] Ted Conrad, Darren Haley, Thad Lieb, Martin Grabau, Sergio Miramontes, Edward Garcia, Samuel Chapple, and Bart Erwin. FLIR FL-100 miniature linear Stirling cryocooler development summary. In *Infrared Technology and Applications XLV*, volume 11002, page 1100202. SPIE, 2019.

[12] Franck Courbon, Philippe Loubet-Moundi, Jacques J.A. Fournier, and Assia Tria. A High Efficiency Hardware Trojan Detection Technique Based on Fast SEM Imaging. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, pages 788–793, Grenoble, France, 2015. IEEE Conference Publications.

[13] Ding Deng, Yaohua Wang, and Yang Guo. Novel Design Strategy Toward A2 Trojan Detection Based on Built-In Acceleration Structure. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):4496–4509, 2020.

[14] Komail Dharsee and John Criswell. Jinn: Hijacking Safe Programs with Trojans. In *32nd USENIX Security Symposium, USENIX Security 2023*, pages 6965–6982, Anaheim, CA, United states, 2023.

[15] Samaneh Ghandali, Thorben Moos, Amir Moradi, and Christof Paar. Side-Channel Hardware Trojan for Provably-Secure SCA-Protected Implementations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(6):1435–1448, 2020.

[16] Vasudev Gohil, Hao Guo, Satwik Patnaik, and Jeyavijayan Rajendran. ATTRITION: Attacking Static Hardware Trojan Detection Techniques Using Reinforcement Learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1275–1289, Los Angeles CA USA, 2022. ACM.

[17] Youngjune L. Gwon, H.T. Kung, and Dario Vlah. DISTROY: Detecting integrated circuit trojans with compressive measurements. In *6th USENIX Workshop on Hot Topics in Security, HotSec 2011*, San Francisco, CA, United states, 2011.

[18] Yuichi Hayashi and Shinichi Kawamura. Survey of Hardware Trojan Threats and Detection. In *2020 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pages 1–5, Rome, Italy, 2020. IEEE.

[19] Jiaji He, Yanjiang Liu, Yidong Yuan, Kai Hu, Xianzhao Xia, and Yiqiang Zhao. Golden chip free Trojan detection leveraging electromagnetic side channel fingerprinting. *IEICE Electronics Express*, 16(2):20181065–20181065, 2019.

[20] Jiaji He, Yiqiang Zhao, Xiaolong Guo, and Yier Jin. Hardware Trojan Detection Through Chip-Free Electromagnetic Side-Channel Statistical Analysis. *IEEE*

Transactions on Very Large Scale Integration (VLSI) Systems, 25(10):2939–2948, 2017.

[21] Bo Hou, Chunhua He, Liwei Wang, Yunfei En, and Shaofeng Xie. Hardware Trojan detection via current measurement: A method immune to process variation effects. In *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*, pages 1039–1042, Guangzhou, China, 2014. IEEE.

[22] Taifeng Hu, Liji Wu, Xiangmin Zhang, Yanzhao Yin, and Yijun Yang. Hardware Trojan Detection Combine with Machine Learning: An SVM-based Detection Approach. In *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 202–206, Xiamen, China, 2019. IEEE.

[23] Ayush Jain, Ziqi Zhou, and Ujjwal Guin. TAAL: Tampering Attack on Any Key-based Logic Locked Circuits. *ACM Transactions on Design Automation of Electronic Systems*, 26(4):1–22, 2021.

[24] Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. In *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, LEET 2008*, San Francisco, CA, United states, 2008.

[25] Atul Kumar, Dipika Deb, Shirshendu Das, and Palash Das. *edAttack* : Hardware Trojan Attack on On-Chip Packet Compression. *IEEE Design & Test*, 40(6):125–135, 2023.

[26] Lang Lin, Markus Kasper, Tim Güneysu, Christof Paar, and Wayne Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, pages 382–395. Springer, Berlin, Heidelberg, 2009.

[27] C. Liu, P. Cronin, and C. Yang. A mutual auditing framework to protect IoT against hardware Trojans. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 69–74, 2016.

[28] Subhasish Mitra, H.-S. Philip Wong, and Simon Wong. The Trojan-proof chip. *IEEE Spectrum*, 52(2):46–51, 2015.

[29] Sparsh Mittal. A Survey of Architectural Techniques for Managing Process Variation. *ACM Computing Surveys*, 48(4):1–29, 2016.

[30] X. Ngo, I. Exurville, S. Bhasin, J. Danger, S. Guilley, Z. Najm, J. Rigaud, and B. Robisson. Hardware Trojan detection by delay and electromagnetic measurements. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 782–787, 2015.

[31] Abdullah Nazma Nowroz, Kangqiao Hu, Farinaz Koushanfar, and Sherief Reda. Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(12):1792–1805, 2014.

[32] Tiago Perez, Malik Imran, Pablo Vaz, and Samuel Pagliarini. Side-Channel Trojan Insertion - a Practical Foundry-Side Attack via ECO. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, Daegu, Korea, 2021. IEEE.

[33] Tiago D. Perez and Samuel Pagliarini. Hardware Trojan Insertion in Finalized Layouts: From Methodology to a Silicon Demonstration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(7):2094–2107, 2023.

[34] Endres Puschner, Thorben Moos, Steffen Becker, Christian Kison, Amir Moradi, and Christof Paar. Red Team vs. Blue Team: A Real-World Hardware Trojan Detection Case Study Across Four Modern CMOS Technology Generations. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 56–74, San Francisco, CA, USA, 2023. IEEE.

[35] Devendra Rai and John Lach. Performance of delay-based Trojan detection techniques under parameter variations. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 58–65, San Francisco, CA, USA, 2009. IEEE.

[36] Amin Sarihi. Hardware Trojan Insertion Using Reinforcement Learning. 2022.

[37] Amin Sarihi, Ahmad Patooghy, Peter Jamieson, and Abdel-Hameed A. Badawy. Trojan playground: A reinforcement learning framework for hardware Trojan insertion and detection. *The Journal of Supercomputing*, 2024.

[38] Gao Shen, Yongkang Tang, Shaoqing Li, Jihua Chen, and Binbin Yang. A general framework of Hardware Trojan detection: Two-level temperature difference based thermal map analysis. In *2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 172–178, Xiamen, 2017. IEEE.

[39] Ting Su, Shaoqing Li, Yongkang Tang, and Jihua Chen. Part I: Evaluation for Hardware Trojan Detection Based on Electromagnetic Radiation. *Journal of Electronic Testing*, 36(5):591–606, 2020.

[40] Ting Su, Jiahe Shi, Yongkang Tang, and Shaoqing Li. Golden-Chip-Free Hardware Trojan Detection Through

Thermal Radiation Comparison in Vulnerable Areas. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1052–1059, 2020.

[41] Y. Tang, S. Li, L. Fang, X. Hu, and J. Chen. Golden-Chip-Free Hardware Trojan Detection Through Quiescent Thermal Maps. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(12):2872–2883, 2019.

[42] Y. Tang, S. Li, F. Zhang, and L. Fang. Thermal maps based HT detection using spatial projection transformation. *IET Information Security*, 12(4):356–361, 2018.

[43] Yongkang Tang, Liang Fang, and Shaoqing Li. Activity Factor Based Hardware Trojan Detection and Localization. *Journal of Electronic Testing*, 35(3):293–302, 2019.

[44] Yongkang Tang, Shaoqing Li, Liang Fang, and Jihua Chen. Statistical Analysis based on Temperature Matrix for Hardware Trojan Detection. In *2018 IEEE International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 143–149, 2018.

[45] Mohammad Tehranipoor and Farinaz Koushanfar. A Survey of Hardware Trojan Taxonomy and Detection. *IEEE Design & Test of Computers*, 27(1):10–25, 2010.

[46] Timothy Trippel, Kang G. Shin, Kevin B. Bush, and Matthew Hicks. ICAS: An Extensible Framework for Estimating the Susceptibility of IC Layouts to Additive Trojans. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1742–1759, San Francisco, CA, USA, 2020. IEEE.

[47] Xiaoxiao Wang, Mohammad Tehranipoor, and Jim Plusquellic. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 15–19, Anaheim, CA, USA, 2008. IEEE.

[48] Mingfu Xue, Chongyan Gu, Weiqiang Liu, Shichao Yu, and Máire O'Neill. Ten years of hardware Trojans: A survey from the attacker's perspective. *IET Computers & Digital Techniques*, 14(6):231–246, 2020.

[49] Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester. A2: Analog Malicious Hardware. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 18–37, San Jose, CA, 2016. IEEE.

[50] Mori Yvon. *Mechanical Vibrations: Applications to Equipment*. Wiley-ISTE, 2017.

A Algorithm Details

We introduce more implementation details of NICE.

A.1 TRMs Classification

As mentioned in § 4.2, the direction-based TRMs classification is the primary algorithm of NICE mechanism, involving a linear model and a soft voting method to classify all TRMs into different direction sets. The processing flow of our implementation consists of four stages, as shown in Algorithm 1.

Algorithm 1 Direction-based TRMs Classification

```
Input: $TRM$, $REF$, $DIR$, $threshold$
Output: Classified $TRM$
      // Preprocessing of TRM Sequences and Golden References
 1: $TRM \leftarrow \textsc{Wavelet}(TRM)$
 2: for $trm_t \in TRM$ do
 3:     $\Delta trm_t \leftarrow trm_{t+1} - trm_t$ // Calculating TR increment
 4: end for
      // Calculating occupation change for each pixel
 5: for $REF\_pixel_{ij} \in REF$ do
 6:     $occups\_range_{ij} \leftarrow \textsc{Max}(REF\_pixel_{ij}) - \textsc{Min}(REF\_pixel_{ij})$
 7:     $DIFF \leftarrow \textsc{Differ}(REF\_pixel_{ij}, DIR)$
 8:     for $diff_n \in DIFF$ do
 9:         if $|diff_n.value| < threshold$ then
10:             $nochange\_set^n_{ij} \leftarrow diff_n.pair$
11:         else if $diff_n.value \geq threshold$ then
12:             $increase\_set^n_{ij} \leftarrow diff_n.pair$
13:         else
14:             $decrease\_set^n_{ij} \leftarrow diff_n.pair$
15:         end if
16:     end for
17: end for
      // Fitting the linear model
18: for $tr\_pixel_{ij} \in \Delta trm$ do
19:     $occup_{ij} \leftarrow \textsc{Approx}(tr\_pixel_{ij}, occup\_range_{ij})$
20:     $\textsc{LinearModel} \leftarrow \textsc{FitModel}(tr\_pixel_{ij}, occup_{ij})$
21: end for
      // Determining trends of pixel occupation over time
22: for $TR\_pixel_{ij} \in \Delta TRM$ do
23:     $PRED\_occup_{ij} \leftarrow \textsc{LinearModel}(TR\_pixel_{ij})$
        // Estimating possible dithering directions
24:     for $pred\_occup_t \in PRED\_occup_{ij}$ do
25:         $\Delta occup_t \leftarrow pred\_occup_{t+1} - pred\_occup_t$
26:         if $|\Delta occup_t| < threshold$ then
27:             $pred\_dir^t_{ij} \leftarrow \textsc{Match}(nochange\_set_{ij})$
28:         else if $\Delta occup_t \geq threshold$ then
29:             $pred\_dir^t_{ij} \leftarrow \textsc{Match}(increase\_set_{ij})$
30:         else
31:             $pred\_dir^t_{ij} \leftarrow \textsc{Match}(decrease\_set_{ij})$
32:         end if
33:     end for
34: end for
      // Classifying TRMs into different direction sets
35: for $trm_t \in TRM$ do
36:     $most\_dir_t \leftarrow \textsc{SoftVoting}(pred\_dir_t)$
37:     Classified $TRM \leftarrow \textsc{Classify}(trm_t, most\_dir_t)$
38: end for
```

1) Preprocessing of TRM sequences and golden references (Line 1-17). The preprocessing of TRM sequences (TRM) involves wavelet filtering and the difference method to calculate the TR increment (Line 1-4). In this paper, WAVELET represents an eight-level wavelet filtering with the "Sym6" base. Golden references (REF) are used to calculate the occupation variation interval ($occups\_range$) specific to each pixel, generated from the IC design based on preset dithering directions. In the subsequent stage, differential pixel occupations ($DIFF$) between references in various directions (DIR) are calculated using the difference method (DIFFER) (Line 5-7). Directions pairs ($diff.pair$) are then classified into three categories by comparing the difference value ($diff.value$) with the threshold: "significantly increasing" ($increase\_set$), "significantly decreasing" ($decrease\_set$), or "no change" ($nochange\_set$) (Line 8-17).

2) Determining trends of pixel occupation over time (Line 18-23). After the TR increment of each pixel evenly corresponds to approximate occupation values ($occup_{ij}$) based on the occupation variation interval ($occups\_range$), the TR increment data of all pixels is used to fit the linear model (LINEARMODEL), which can estimate the occupation of each pixel at every sampling time ($PRED\_occup_{ij}$).

3) Estimating possible dithering directions (Line 24-34). Based on outputs of the linear model, the occupation change of each pixel ($\Delta occup_t$) is calculated using the difference method, and evaluated for significance based on the threshold. Possible dithering directions ($pred\_dir$) are approximated by matching (MATCH) results to the corresponding category.

4) TRMs classification (Line 35-38). For the entire TRM at each sampling time (trm_t), the most likely direction ($most\_dir_t$) is determined using the soft voting method (SOFTVOTING), as depicted in the following equation, enabling the classification of all TRMs into the corresponding direction sets (Classified TRM).

$$Prob_{max} = \max_{1 \le k \le n} \{ \sum_{i=1}^{M} \sum_{j=1}^{N} p_{ij}^{dk} \}$$

$$p_{ij}^{dk} = \begin{cases} 0 & , dk \in possible\ directions \\ \frac{1}{number\ of\ possible\ directions} & , dk \notin possible\ directions \end{cases}$$

Where, the probability of entire TRM dithering in every direction is determined by the sum of probabilities (p_{ij}^{dk}) of pixels dithering in that direction (dk). The maximum of results ($Prob_{max}$) indicates the most probable dithering of TRMs.

A.2 HT Detection

As mentioned in § 4.3, each direction set of TRMs (TRM_{dir}) is subjected to HT detection. Algorithm 2 shows the processing flow of our implementation.

Reference [41] has discussed the principle of the AA shape generation in detail. This stage involves selecting some pixel samples ($pixel\_sample$) from the entire TRM until their TR increment data conforms to a normal distribution, as verified by the K-S statistic (Line 1-18). In this paper, SELECT is implemented by traversing all rows and columns of the TRM. Then, according to the Pauta criterion, vacant pixels can be distinguished from logic pixels based on the mean ($mean$) and standard deviation (std) extracted from the TR increment data of logic pixel. To prevent random errors, a pixel must be considered a vacant pixel at most sampling times, which is determined by P_vacant (Line 19-26).

After comparing the result ($type_{ij}$) with the golden reference at corresponding direction (ref_pixel_{ij}), several suspicious pixels (SUS) can be identified (Line 27-30). To reduce the possibility of false alarms, further analysis is conducted to determine whether the pixel is the logic region in most references at other directions when the suspicious pixel is detected in only one direction (Line 31-41). Beyond that, other suspicious pixels should be regarded as potential HT pixels.

Algorithm 2 HT Detecting and results aggregating

```
Input: $TRM, REF, DIR$
Output: $result$
     // HT detection for each direction set
 1: for $dir_n \in DIR$ do
 2:     $TRM_{dir} \leftarrow TRM[dir_n]$, $ref_{dir} \leftarrow REF[dir_n]$
 3:     for $trm_t \in TRM_{dir}$ do
 4:         $\Delta trm_t \leftarrow trm_{t+1} - trm_t$
            // K-S statistic
 5:         repeat
 6:             $pixel\_sample \leftarrow$ SELECT($\Delta trm_t$)
 7:             if K-S($pixel\_sample$) is True then
 8:                 $mean \leftarrow$ MEAN($pixel\_sample$)
 9:                 $std \leftarrow$ STANDARD($pixel\_sample$)
10:             end if
11:         until K-S($pixel\_sample$) is True
            // Pauta criterion
12:         Initialize a 2D matrix: $P\_vacant = 0$
13:         for $tr\_pixel_{ij} \in \Delta trm_t$ do
14:             if $mean - tr\_pixel_{ij} \ge 3 \times std$ then
15:                 $P\_vacant[ij] += 1$
16:             end if
17:         end for
18:     end for
19:     Get the number of TRMs in each direction set: $N$
20:     for $p\_vacant_{ij} \in P\_vacant$ do
21:         $ref\_pixel_{ij} \leftarrow ref_{dir}[ij]$
22:         if $p\_vacant_{ij} > N/2$ then
23:             $type_{ij}$ is Vacant Pixel
24:         else
25:             $type_{ij}$ is Logic Pixel
26:         end if
            // Identifying suspicious pixels
27:         $sus\_pixel_{ij}^{n} \leftarrow$ COMPARISON($type_{ij}, ref\_pixel_{ij}$)
28:     end for
29:     $SUS[dir_n] \leftarrow sus\_pixel^n$
30: end for
     // Result aggregating
31: for $SUS\_pixel_{ij} \in SUS$ do
32:     if $SUS\_pixel_{ij}$ is the Suspicious Pixel at only one direction then
33:         if $REF\_pixel_{ij}$ is the Logic Pixel at most of directions then
34:             $result_{ij} \leftarrow$ is not HT Pixel
35:         else
36:             $result_{ij} \leftarrow$ is HT Pixel
37:         end if
38:     else
39:         $result_{ij} \leftarrow$ is HT Pixel
40:     end if
41: end for
```