Wireless Personal Communications (2024) 136:1147–1182
https://doi.org/10.1007/s11277-024-11334-6

Hardware Trojans Detection and Prevention Techniques Review

R. Naveenkumar[1,2] · **N. M. Sivamangai**[1]

Accepted: 8 June 2024 / Published online: 25 June 2024

Abstract
Technological developments in semiconductors have created previously unheard-of chances for creativity, but they have also increased the danger of hardware Trojans, which are malevolent modifications introduced into integrated circuits (ICs) during the design or production phases. This research review addresses the changing landscape of threats and responses by examining the most recent advancements and trends in hardware trojan detection and prevention approaches. Proactive protections against Trojan insertion and dissemination include methods like cryptographic primitives, trust verification protocols, and hardware obfuscation. The field of detection approaches has expanded to include a multi-layered approach that integrates emerging technologies like artificial intelligence and machine learning with more established methods like testing and design-time analysis. Furthermore, it is possible to improve resistance to Trojan assaults while reducing performance overhead by incorporating hardware security features like physically unclonable functions and secure compartments directly into the IC architecture. Moreover, various prevention algorithms, detecting challenges and effects of the HT in recent applications are summarized with its solutions.

Keywords Hardware Trojan · Trust of the circuit · Formal verification · Trojan detection · Split manufacturing · Hardware Trojan prevention

1 Introduction

Electronic devices have gained popularity among all users due to the latest developments in chip technology, social networking, and security measures. In addition to their usefulness for information sharing, these electronic devices have certain disadvantages. Nevertheless, they are nevertheless crucial in modern life. Internal memory on most modern devices is

✉ R. Naveenkumar
naveentamil256@gmail.com

N. M. Sivamangai
nmsivam@gmail.com

[1] Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore 641114, Tamil Nadu, India

[2] Department of ECE, Sri Krishna College of Technology, Coimbatore 641042, Tamil Nadu, India

quite substantial. Passwords, account numbers, pin numbers, banking information, and other user files can be stored on it. When HTs are inserted into the device, it becomes easy to share this sensitive information with others [1]. HT is defined as a destructive modification that causes the circuit to perform wrongly. Because the hardware Trojan is deployed at the device's most basic level, it poses a serious threat as long as the device is in use. Malware or dubious software that is installed on hardware is typically referred to be HT. The circuits' behavior can be altered by this type of HT [2]. Additionally, it can be infused throughout the fabrication and design phases.

HT modifies the functionality of an affected IC by bypassing the IC's hardware or software security protections. Modifications to the IC can have several unintended consequences, including the loss of important and sensitive information, DoS attacks, and undetected service degradation that leads to system failure. Backdoor access to a system can also be provided by hardware Trojans, in which high-level software collaborates with the hardware Trojan to gain access to and disable highly secure systems [3]. Circuit functionality can be changed through the HT structure. When HT is activated, it results in erroneous activity (by changing the 1 to 0). Trigger circuits are typically used to mimic the HT activation process. Payload is the result of alterations, such as (i) DoS attacks.: HT will make the circuit, no longer capable to function correctly, so it does not process the function properly (ii) Reduce reliability: hackers can implement HT codes or programs in the system to reduce the reliability and degrade the system performance and HT consumes high power and if it is battery application, it will quickly discharging and stop the circuit operation, (iii) changing the function: Trojans changes, insert and remove the original circuit function [4]. In rocket launching towards a specific target, it can change and set the wrong target destination. And it leads to improper operation of the circuit. (iv) Leakage of information: mainly HT designed for revealing the secret key or text through primary output signals.

Broadly, a trojan is groped into hardware and software Trojans. HT resides in the hardware components of the IC; during the process, it will be activated and once the IC is manufactured, HT cannot be eliminated [5]. The Software Trojan (ST) is a malware program that uses malicious code to gain special access to the operating system, with the potential to steal data or harm the host computer (for example, erasing or destroying data). ST Attacks can usually be cured by running counterattacks, Trojan horse programs, and monitoring and deleting Trojans [6]. The hardware Trojan is to inject into the IC before manufacturing. In ST, it is almost impossible to remove hardware Trojans after the chip is manufactured, it can be very difficult to Remedy in on-site operation. In ST firmware attack is a major role here some of the firmware-based HTs are discussed [7]. Hackers can easily inject malware into the firmware of the system to steal sensitive data or take control of the entire controller. Several forms of firmware attacks could occur, including the following:

(a) Maliciously designed input: An attacker could leverage buffer overflows to implant malware using this method. (b) Privilege escalation: An attacker uses System Management Mode (SMM) code injection to bypass security functions, (c) Data tampering: occurs when an attacker alters UEFI variables (Secure Boot, Configuration, etc.). (d) unauthorized access to sensitive information: contents of the System Management Random Access Memory (SMRAM) were revealed, (e) Data leakage: SMM-based malware; "secrets" stored in memory. (f). Denial of Service: The system is "bricked" due to serial peripheral interface (SPI) flash corruption.

To protect against data spillage caused by hardware Trojans, [8] devised a hardware isolation-based security method shown in Table 1. The secure application processor allowed them to catch stolen secret information in a safe isolated environment, which they were able to do, preventing data leakage from a hacked system. They demonstrated low overhead and

Table 1 Hardware isolation security analysis with different attack vectors [8]

S.No	Attack vectors	Hardware isolation			No Hardware isolation
		With output verification, it's safe	Hardware isolation controls access, making it secure	Only the memory access pattern side channel is secure	
1	HT attacks on information leaks	√			Un-Secured
2	Software attacks		√		Un-Secured
3	Attacks by DMA		√		Un-Secured
4	Attacks from the side channel			√	Un-Secured
5	Attacks that cause a denial of service		√		Un-Secured

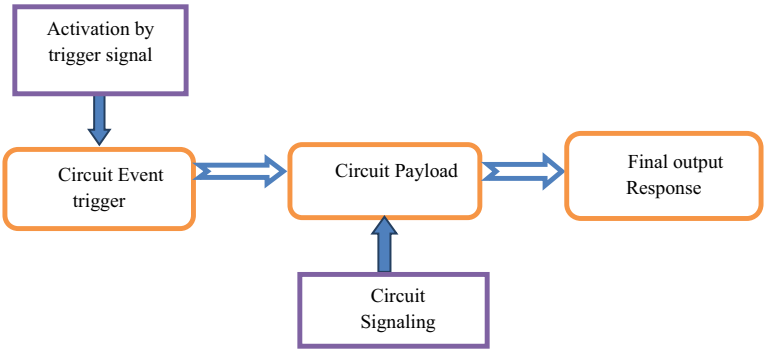

Fig. 1 Structure of hardware trojan

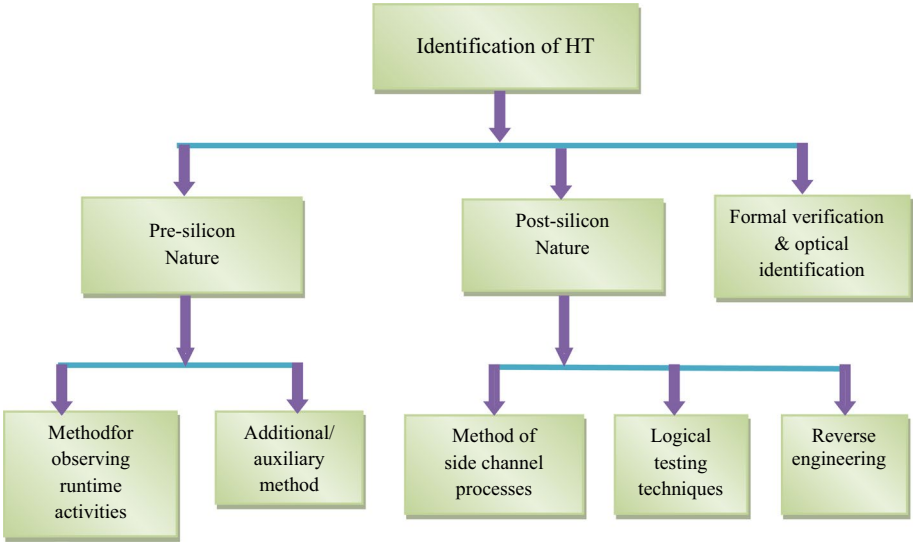

Fig. 2 Classification HT identification

effectiveness of the suggested method. The Xilinx-FPGA ZYNQ: 7000 SoC incorporates both the information leak hardware Trojan and hardware isolating protection. The boundary-scan technique was used to examine connections between different chips that support the JTAG (Joint Test Action Group) interface. Interconnecting the device through JTAG signal lines creates a boundary scan path inside the board. The multifunctional memory cell is the boundary scan cell (Figs. 1 and 2).

Deyati et al. [9]., devised an approach for detecting that works in conjunction with the existing pipeline scan chain testing system or JTAG boundary scan. Regarding both area and power, incorporating a Trojan approach for detecting into the scan chain has fewer effects. To detect the propagation of pulses through logic gates, a high-resolution current sensing technique was proposed. A single sensor can detect pulses in a variety of locations. To use end-to-end solutions for hardware Trojans, with reduced area overhead, the full concept of pulse-based Trojan detection was already incorporated further into the JTAG

boundary scan technique. Different approaches for detecting by using JTAG boundary scan testing. HT can be applied at the system level in integrated circuits (ICs), including tiny latency defect and transition error testing [10, 11]. Boundary-scan testing methods are beneficial for sensitive data nets that cannot be checked using in-circuit test devices or probe pads; the boundary scan testing methods are useful. For example, to evaluate if there is an open issue on a trace connecting two integrated circuits with boundary scan (BS) capabilities, BS testing can be conducted. The PCI-Express device specification specifies a boundary scan implementation for devices that comply, which could be employed to assure that any connected compliance devices are authenticated and reliable [12].

Furthermore, although external JTAG connection to the gadget is prohibited prior to deployment, the microprocessor or microcontroller may include self-verification operated methods that rely on such BS monitoring to ascertain whether field tampering is acknowledged [13, 14]. In recent applications, hardware Trojans have caused major effects. Recently Dong et al. [15], discussed smart gadgets, robotics, and self-driving cars that rely on SoC devices in the internet of things (IoT) simply cannot function without the use of sensors to detect sense operations. They will also endanger particular equipment on the linked network if they have holes in their HT security. Especially chips on radars that monitor the atmosphere with comparable flaws cause the same degradation [16, 17]. To get the authorization to modify the data. To gain unauthorized access to memory, an HT assault on the printed circuit boards (PCB), the JTAG interface, and the data bus are all that is required [18]. On network on chip (NoC), there existed a type of HT capable of delivering bandwidth denial of service (DoS) attacks. Communication was hindered, application performance was reduced, and system reliability was harmed as a result of this attack [19].

In a wireless network, HTs: Indeed, following more than a decade of investigation, digital IC defense and preservation technologies have advanced and proven to be successful [20]. Electronic gadgets are highly dependent on wireless connection technologies, these provide HTs additional chances to launch attacks and put up fresh barriers. HTs mitigation in internet connections will be a crucial task in the future. HTs in wireless encryption circuits were examined by Jin et al. [21]. To provide an instruction or disclose information, the HT must add format to the signal before it can be communicated. Advanced statistical analysis of several parameters of operation (amplitude, frequency, phase, etc.) was combined by the author, statistical techniques to find Malware in the test circuitry efficiently. A successful assault is impossible to carry out because the attacker has no idea what the designer would look at. Created a simulation of two different classification of HTs found in RF pulse generators as well as power amplifiers. The keys in the wireless network will be revealed by these HTs. On the output/input of both HTs are P-MOS transistors. Transistor P-MOS is activated once the key bit is set to 0, and deactivated whenever the key bit is set to 1. The HTs don't interfere with the regular circuit's functionality and only slightly alter the original circuit. He et al., worked on a memristor-oriented HT for the HT catcher technology was created with the help of artificial intelligence (AI)-chip, which is related to side-channel analysis, and decreases feature calculation memory overhead by at least a quarter [22]. This paper deals with various HT Detection methods like pre-silicon, post-silicon-based detection, formal verification, and optical detection. Also, discussed the HT prevention techniques like hardware obfuscation, physically unclonable function, split manufacturing, and trust of design. In prevention technique, utilization of layout filling effectively achieved in SCA-based hardware (design) obfuscation is discussed. Moreover, various prevention algorithms, detecting challenges and effects of the HT in the recent application are summarized with its solutions. The rest of the paper is organized, as

follows. Section 2, discussed classifications of trojan detection techniques, and their metrics are summarized from the literature. In Sect. 3, expands various prevention measures for HT. Section 4. Suggested the different HT prevention algorithms. Section 5 highlights the challenges and future scopes, and it concludes in Sect. 6.

2 Classifications of Trojan Detection Techniques

Among various HT classifications, only the primary varieties of formal verification, optical detection, post-silicon-based HT detection, and pre-silicon-based HT detection are examined.

2.1 Pre-silicon HT Identification

The pre-silicon identification is the process of identifying HTs that are placed in by IP cores or inserted by EDA tools [23], in pre-silicon identification, netlist, domain, RTL code, and other objects are employed. Generally, hardware hackers introduce Trojans into the design to add to their undesired behavior. And updating firmware-based can't eliminate Trojans. This technique depends mainly on rare activating nets, process variation, and noises. This technique is divided into two types. (i) Monitoring the runtime activity method and (ii) Auxiliary method. Figure 3 shows the important two categories under pre-silicon HT identification.

2.1.1 Method for Observing Runtime Activities

It can monitor anomalous behavior and operate during run-time activity; it is a non-destructive method. matching the response to the conventional golden circuit. If the authentication is done during test time, identify the HT at all times (logic testing and side-channel testing are two examples). Examine the style, device structure, and objects on a regular basis when doing this procedure. This straightforward idea makes it easy to recognize the HT's activity. The process-oriented HT will be identified in its entirety using the IC. Furthermore, the identifying answer is not one-time. For all kinds of HT activities, the disadvantage is a little complicated [5, 24]. Khalid et al. proposed using the traditional checking approach to identify the HT. It increases the test time overhead by five to thirty times while consuming less space [25].

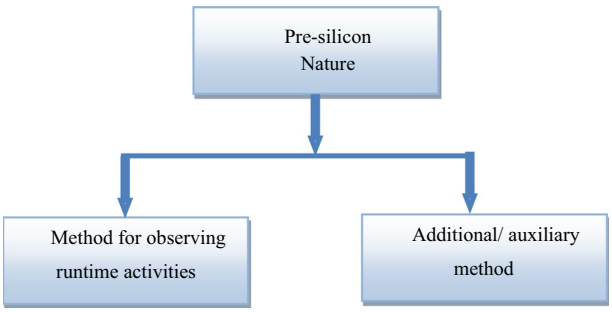

Fig. 3 Classification of Pre-silicon based HT

Machine Learning (ML) focused detection is one method that could be able to get beyond these restrictions. In comparison to formal inspection techniques, machine learning (ML) algorithms could be trained to identify patterns linked to hardware Trojans, potentially providing a more effective and scalable solution. A large number of known HT cases are analyzed so that ML models can be trained to spot minute changes in hardware performance that might point to the existence of a Trojan. In comparison to formal checking techniques, ML-based detection systems may be more flexible to various HT activities, scalable, and possibly less resource-intensive. Furthermore, ML models are able to updated and trained on a regular basis to accommodate new dangers related to HT.

A multi-scale detection model was put into practice by Pengcheng Ma et al. for automated feature extraction. Two methods are offered by the MHT text model to balance computation requirements and accuracy. The MHTtext model exhibits excellent stability, flexibility, and accuracy overall, the findings for the standard netlists, where one of the TextCNN's stability efficiency index coefficients scores first with a total rating of 71.21 in all comparing classifiers, and the average accuracy (ACC) in the entire strategy is as high as 99.26% [26].

K. Hasegawa et al. developed a maximum single output sub-module and algorithm for partition oriented identification of HT and diagnosis technique for gate-level netlists (GLNs) based on ML and graph theory (GT). Increased HT diagnostic accuracy by the use of the breadth-first comparison (BFC) implant node search technique. The Junjie Wang team conducted studies on HT identification and diagnosis using Trust-Hub examples, with the assistance of the previously indicated methodologies. According to the experimental data, the TNR exceeded 37%, the F1 values over 97%, and the TPR exceeded 95%. With this approach, the TNR for GLHTs is increased to 25%, the TPR for GLHT diagnosis is continuously higher than 93%, and the TNR is 100% [27]. In order to prevent unwanted access by third parties in the system-on-chip and network-on-chip, Hussain et al. introduced a designer authentication technique [28]. Bao et al. [29] explained their method of in-network on the chip level, the HT has injected a tag in a packet in the transmission part, and the receiving part of the packet is verified with the standard reference with the original tag. In case it's modified. It is considered like Trojan affected packet. The temperature of the IC is monitored by an extended Kalman filter-based identifies the thermal effects.

2.1.2 Auxiliary Method

By using this strategy, side-channel and logical testing methods become more accurate. The output response is contrasted with the circuits that are HT-affected and HT-not-affected [30]. Salmani et al., introduced a virtual scanning-based triggering approach. Initially, HT identification is tough during the probability of rare activation [31]. According to Zhang et al., there is a high likelihood of transition and a high probability of activating when dealing with the virtual scanning idea. With a shorter activation period, this method provides superior identification. HT in the design domain is identified using the recently suggested hardware trustworthiness and verification paradigm [32]. In most cases, HT is activated by input triggers and is not susceptible to triggering input signals. In this case, the HT is identified by evaluating the modules as part of the verification procedure [33]. Utilizing electron microscopy scanning, Courbon et al. proposed a method for detecting a single layer in a chip. Here, the conventional version of the circuit is compared, and it is then determined which has been altered in terms of location, routing, area, or any other way, as well as any additions or changes to the internal circuit design. Here, contrast

the images based on SEM with the HT. This one has the benefit of precisely identifying the HT, but it also has the disadvantage of being more expensive [34]. The important parameters are reviewed in Table 1. Table 2 suggests in order identifying HT based on good detection, smaller area, powerhead, and external factor influence, pre-silicon (run time monitoring) is preferable.

2.2 Post-silicon HT Identification

Conventional pre-silicon detection techniques have a number of flaws. Because many pre-silicon detection techniques require prior Trojan knowledge used to identify Trojans with a specific structure or intent. Formal verification and functional simulation can only discover Trojans with precise behavior. There are substantial limits to logic testing and other post-silicon detection approaches that require the Trojan to be activated for detection. It is challenging to turn on the malware in physical detection because there aren't any Trojan triggering events. Post-silicon detection is the identification of HTs inserted during the manufacturing and assembly phases. Post-silicon identification is utilized for identify actual IC produced by untrustworthy manufacturers [38], Fig. 4 depicts the three types of post-silicon HT detection; there are method of side channel processes, logical testing, and reverse engineering.

Side-channel process-based Trojan identification is a popular post-silicon method [34].

2.2.1 Side Channel Techniques

In side-channel signal analysis, the HT affected in the chip (or) IC can modify the characteristics of power consumption. Trojan's partial activation can be incredibly useful for power analysis. Side-channel Trojan identification focuses on observing the Trojan effects in the physical side-channel parameters like leakage current. Switching current, temperature, path delay, timing information, electromagnetic emission (EM), and power consumption are shown in Fig. 5.

Due to process variation, detection is extremely challenging to identify the Trojans. Through side-channel assessment, various side-channel analysis parameters, including voltage (V), temperature (T), and path latency (Pd), are compared between our experimental IC and the standard IC [1]. It is susceptible to noise from processes. Very easy to create test patterns and highly appropriate for large Trojans; nevertheless, ultra-small Trojan identification is more challenging and has more fault coverage [5]. It is figuring out if HT has an impact on the experimental IC or not. Discussed the side-channel signal analysis approach and conclude the hardware overhead and circuit design remain unchanged [39]. Additionally, a frequent term in VLSI is "side-channel assessment signal identification," which also refers to a different method of inspecting the design in terms of side-channel signal parameter features. By comparing the delay of an experimental circuit with a common reference circuit model, the result can be obtained with ease. The easiest and fastest way to program or code is to use a path delay-based strategy. A novel side-channel signal assessment identification method that relies on the path delay has been proposed [40]. This idea combines the design and latch structure. Moreover, the latch structure makes it simple to see every scenario in which an HT could be inserted. Furthermore, this method compensates for the inferior performance of side-channel signal analysis by eliminating the impacts of noise. Proposed the location of HT based on thermal imaging analysis [41]. In cases where the gate count is fewer than 20, it is typically utilized to determine the HT.

Table 2 Diverse parameter evaluations of Pre-silicon identification

Identification	Techniques	Standard golden reference model required	Detection Accuracy	Area, Powerhead	Cost	External parameters influence	References
Auxiliary detection nature	Utilizing fast SEM imaging	√	√	Less	High	×	[34]
Auxiliary detection nature	Timing nature based detection	×	Good Detection probability: 95% False positive:5%	Less Area overhead: 1.3–3.94%	Design based	√	[33]
Auxiliary detection nature	Verification of Hardware Trust	√	√	Based on the design	Moderate	×	[32]
Auxiliary detection nature	Virtual scan activator	×	√	Less	Based on the design	×	[31]
Runtime monitoring nature	Using a runtime monitor to explicitly model the HT	×	√	Less	Based on the design	×	[25]
Runtime monitoring nature	Framework for detection	×	√	Less	Based on the design	×	[35]
Runtime monitoring nature	Using the authentication method	√	√	Less area overhead: 3.37% Power: 2.61%	Based on the design	×	[28]
Runtime monitoring nature	Temperature in IC	×	√	Less	Very low cost	×	[29]
Runtime monitoring nature	on-chip ring-oscillator networks to identify anomalies through unsupervised clustering	×	99% accuracy rate	Less	Based on the design	×	[36]
Runtime monitoring nature	Based on true positive rate (TPR)	No	√	power overhead of 0.005%	Based on the design	×	[37]

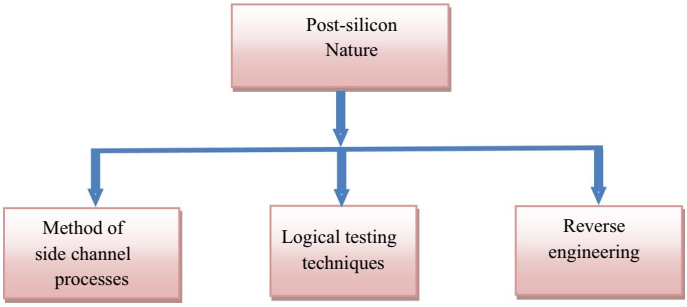

Fig. 4 Classification of Post-silicon based HT

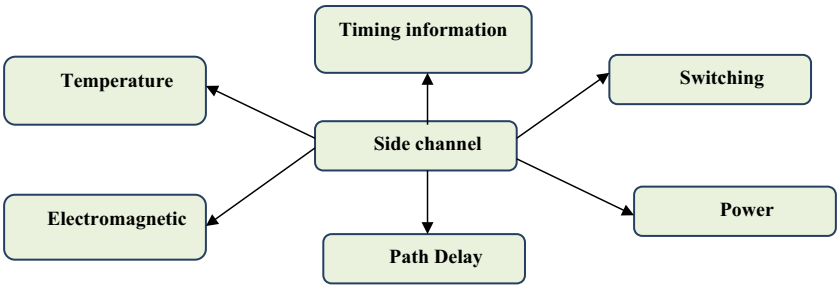

Fig. 5 Common parameters of side-channel signals

Here, the thermal variations of the HT in the redundancy thermal map define the activity factor. Side-channel identification via electromagnetic means is extremely precise if the gate count falls between ten and one hundred.

2.2.2 Logical Testing Techniques

Logical testing is employed to create any number of test vector data in order for stimulate the hardware Trojan under unexpected activation situations [1]. Although it is challenging to create test patterns and resistant to process noise, logical testing is appropriate for ultra-small Trojans. For larger Trojans, however, fault coverage is reduced and Trojan identification is challenging but straightforward to distinguish [5].

Logical testing combined with side-channel approaches can improve fault coverage. Due to the modest number of test patterns utilized, this method may work well for smaller circuits but may not scale well for bigger ones, which could allow Trojans to evade detection. In order to remedy this, scientists should concentrate on creating more thorough test patterns that encompass a larger spectrum of plausible Trojan behaviors. This might be accomplished by integrating machine learning techniques to produce tests that are more effective and diverse. Increasing fault coverage in hardware Trojan detection entails a number of techniques meant to improve the system's detection capabilities. Utilize a variety of test vectors in your testing to cover as many potential Trojan behaviors and activation scenarios as feasible.Functional and structural tests ought to be incorporated into these vectors in order to identify anomalies at various abstraction levels. Furthermore, by

learning from previous occurrences and especially adapting to new attack vectors, improved stochastic testing, Security-Oriented testing, side-channel analysis, formal verification, runtime monitoring, and machine learning can improve the detection capabilities.

With the use of sophisticated testing approaches, Huang et al. presented a novel way to improve fault coverage in hardware Trojan detection. Suggested combining transition delay fault testing with path delay testing to enhance Trojan detection capability. Comparing experimental results to conventional scan-based testing methods, a set of benchmark circuits showed an average 15% improvement in fault coverage [42].

A machine learning-assisted method was introduced by Maitreyi Ashok et al. to increase fault coverage in hardware Trojan detection. A convolutional neural network (CNN) was taught to identify patterns in circuit responses that suggested Trojan activities. When compared to conventional testing methods, experimental results on a variety of Trojan-infected circuits demonstrate an average 25% improvement in fault coverage [43].

Hardware Trojan detection can potentially be improved by using side channel analysis techniques like power and electromagnetic emission monitoring. In order to identify Trojans, J. He et al. suggested a novel method that combines power analysis with electromagnetic emission analysis. An average improvement of 18% in fault coverage was shown in experimental results conducted on real-world circuits [44].

The method was used by Hicks et al. by contrasting the experimental outcomes with the typical original model outcomes of the simulated measurements. HT is clearly identifiable in the event that the experimental technique affects the results from the IC [45]. Zheng et al. [46] are concerned with the creation of test vectors for activating Trojan circuits and observing harmful payload issues at the major outputs. As per reference, this presents probability signatures -HT identification that is appropriate for the test vectors' structural and functional features. The probability signature-based experimental circuit is compared to the original probability signature circuit to inject HT. So this technique can easily find out all HT in the contaminated circuit, even though it needs some standard reference model (example: golden netlist sample). Bazzazi et al., introduced logical testing-based approaches, in terms of primary circuits to secondary circuits, each of the secondary circuits has some specific node values. To get the node with a larger similarity response by output nodes. These kinds of related nodes are used to test nodes whose relation matches the value and obtained values are the difference among the HT affected and not affected circuits [47]. A statistically scalable test generation technique has been proposed, this method is flexible and it overcomes noise and process fluctuation issues. It has the ability to produce excellent test patterns that can produce extremely similar activities in any type of HT instance. It produced effective test patterns using side-channel nature HT identification. In general combination of side-channel techniques and this logical testing can give better coverage. It is not suitable for large circuits, because only a few and more effective test patterns only used, so there are possible chances to escape the Trojans. The fault coverage is low for manufacturing tests [48].

M. Priyatharishini. et al. suggests, the compressive sensing method for recognizing HT for all test patterns that may be obtained using the proposed technique [49]. The logic testing method compares the logic values at the nodes in the golden circuit to the logic values in the Trojan IC circuit. Tables 3 and 4 compare the logic values of Trojan-infected and golden circuits. When a combinational form of Trojan is introduced in the minimal transition probability node, the logic in the nets changes, disclosing the presence of the Trojan module.

Tables 3 and 4 show the comparison of primary output for the various input pattern with internal nodes for the reference and trojan affected C17 benchmark circuits. Results

Table 3 In Logic Testing, the output function is compared to a golden reference

S.No	Input pattern	C17 benchmark-golden reference					
		Internal nodes				Primary output	
		N6	N7	N8	N9	N10	N11
1	1	1	1	1	0	0	1
2	7	1	0	1	1	0	0
3	14	1	0	1	1	0	0

Table 4 In Logic Testing, a comparison of the output function with a gold standard is made

S.No	Input pattern	C17 benchmark-trojan affected					
		Internal nodes				Primary output	
		N6	N7	N8	N9	N10	N11
1	1	1	1	1	0	0	1
2	7	1	**1**	1	1	0	**1**
3	14	1	**1**	1	1	**1**	**1**

clearly show that in trojan affected C17 benchmark circuits primary output changes for the input pattern of 7 and 14.

Popat et al. [50] introduced the True Positive Rate (TPR) metrics, which are used to validate the effectiveness of compressive sensing in identifying Hardware Trojans. The circuit under test is applied to a probability of detection (PD) test. The TPR value represents the number of Trojan nets that are harmful nets, whereas the false negative (FN) value indicates the number of the Trojan net that is mistakenly recognized as normal nets in binary classification.

Generally speaking, the logical testing method has certain drawbacks, such as false positives and false negatives, a high level of complexity at the large scale, vulnerability to evasion, and limited scalability and scope. Nevertheless, this technique is particularly useful for detecting Trojans in digital circuits where their behavior can be precisely determined by logical operations. It is possible to identify aberrant behavior suggestive of the presence of a Trojan by examining the circuit's responses to different inputs. This process also allows for functioning circuit certification. Because of its dual function, designers can guarantee that the circuit operates correctly and is secure from Trojan intrusion.

Deep learning accelerators (DLAs) are now essential parts of hardware for artificial intelligence. Nonetheless, a serious security concern arises from their susceptibility to hardware Trojans. A logic testing-based method for identifying Trojans in DLAs was presented by Govindan, V et al. They present new methods for creating test patterns and examining answers to find malevolent changes in the accelerator's behavior. Their approach is effective in detecting Trojans with 95% accuracy and only 2% area overhead, as demonstrated by experimental findings [51]

A key INV/BUFF concept, which generates a better design than XOR/XNOR with less overhead, was proposed for use by Naveenkumar et al. The proposed technique can greatly improve logical locking without sacrificing security. Furthermore, in contrast to the

XOR-based technique, it reduces overheads by 2.76 percent, 12.92 percent, and 12.7% for space, power, and time, respectively [52].

Although neuromorphic computing systems have a lot of potential uses, their security and functioning can be jeopardized by Trojan attacks. A logic testing-based method for Trojan identification in neuromorphic computer systems was developed by M. Grailoo et al. In order to identify possible Trojans, they provide methods for creating test patterns and examining neural network responses. Experiments conducted on cutting-edge neuromorphic devices show that their approach is 97% accurate for detecting Trojans with an acceptable false positive rate of 1.5% [53]. The goal of Naveenkumar et al. [52] was to combat SAT assaults, counterfeiting, and piracy. Created the obfuscation method for the Anti-Sat circuit block to provide security against SAT assaults. Through iterations, they created a structural obfuscation for the C17 benchmark that was anti-sat based. The recommended XOR/XNOR logic locking anti-sat with structural obfuscation idea improves security from an iteration standpoint, according to experimental results. Furthermore, the findings verified that, computationally speaking, the SAT attack is not possible, and that the number of SAT attack iterations required to reveal a correct key in a circuit with an Anti-Sat Block is proportionate to the key size.

Although quantum computing presents previously unheard-of levels of computational power, it also poses new security risks, such as the possibility of hardware Trojans. The use of logic testing techniques for Trojan detection in quantum circuits was studied by C. Chu et al. They presented new methods for modifying conventional logic testing strategies for use in the context of quantum computing. Tests carried out in the lab revealed that the technique was 92% accurate at identifying Trojans in quantum circuits while having no negative effects on the performance of quantum gates [54].

In hardware the integrity of integrated circuits is seriously threatened by trojans. A machine learning-assisted logic testing method for Trojan identification in hardware security was presented by F. Khalid and colleagues. They make use of machine learning techniques to improve logic testing's efficacy in detecting Trojans. Combining machine learning techniques with conventional logic testing results in increased efficiency and accuracy. The approach's efficiency in detecting Trojans with 96% accuracy and decreasing false positives to less than 1% was proved by experimental findings [55].

2.2.3 Reverse Engineering

Reverse engineering (RE) aims to investigate an IC's internal physical structure. Using delayering and a scanning electron microscope, RE is used to obtain layer-by-layer pictures. Discovering the transistor, gates, and routing circuits using the template matching method. Here, the experimental image is compared to the original, conventional reference image using the classic layout model as its foundation. This type of visual examination is carried out under a microscope [1]. In general, detection rates can be identified through reverse engineering, which yields far greater rates. Additionally, HT can be directly observed in the structure itself. Additional type assessment can be performed to examine the HT. An HT identification concept is proposed and relies on a physical inspection at the circuit level. First, it contrasts the image of the experimental circuit with that of a standard image circuit. Destructive pictures derived from reverse engineering were also recovered [56]. The parameters investigated in post-silicon identification techniques are displayed in Table 5. Due to its resistance to interference and overhead caused by process fluctuation,

Table 5 Diverse Parameter Evaluation Identifications Post Silicon

Identification	Strategies	Locate with ease and Improve Identification	Referring golden model is required	Interference from process variation	Over head	References
Side Channel Analysis	Thermal nature	Easy	Easy	NA	NA	[41]
Side Channel Analysis	Path delay nature	Easy	Difficult	Higher	Nil	[39]
Side Channel Analysis	Temperature and matrix nature	Easy	Easy	Higher	–	[57]
Side Channel Analysis	Path delay nature	Easy	Easy	Lower	Higher	[1]
Side Channel Analysis	Light Emission nature	Easy	Difficult	Lower	–	[58]
Side Channel Analysis	Side-channel analysis with unsupervised machine learning	Easy	Difficult	Lower	-	[36]
Logic testing	Scalable logic testing	Easy	Easy	Lower	–	[47]
Logic Testing	Probability Signature nature	Easy	Easy	–	–	[46]
Logic Testing	HT Based on Logical Testing	Easy	Easy	–	13% power and 15% area	[47]
Reverse engineering	On the basis of a visual examination at the circuit level	Easy	Easy	Lower	Lower LUT:0.5%	[56]
Reverse engineering	Using Support Vector Machines, or SVMs	Easy	Difficult	Lower	Higher	[59]

*NA: Not Applicable

the logical testing parameter suggested by this table, which relies on probability signature analysis, is efficient.

2.3 Formal Verification and Optical Detection

This technique is generally affected by computation complexity and also poor automation tools [2]. To avoid this issue, the proposed proof-carrying hardware (PCH) approach is used [60]. This proof-carrying hardware assures the trust of the device [53, 61–63]. PCH is a method for ensuring hardware's trustworthiness proposed the Proof-carrying code (PCC) is a type of proof-carrying code that was the inspiration for the PCH (Proof-Carrying Hardware) approach [64]. Unreliable software designers use the PCC method to validate their code. Software vendors develop safety policies and safety proofs that software clients submit for certification. Next, the provider provides the customer with a PCC binary file including official documentation attesting to the security features of the program. Using a proof checker to quickly verify the PCC binary file, the customer can be confident that the software code is safe. Because of the effectiveness of this strategy in lowering verification time at the client end, it has been adopted in a variety of applications.

Despite its benefits, the PCC technique necessitated a vast and reliable computing infrastructure (TCB). To get over this limitation of PCC, foundational PCC (FPCC) is a conceptual structure that makes use of foundation logic to describe the practical semantics underlying various assembly language instructions [65, 66]. Drzevitzky et al. [53] suggested a basic Proof-Carrying Hardware (PCH) for dynamically reconfigurable machines based on the PCC architecture. The main concept of reconfigurable platforms is to perform runtime combinational equivalences testing to compare the conceptual design and the design execution. Despite its name, PCH is a combinational equivalence verification method based on an SAT solver, with the difference that the resolution proof traces are treated as functional equivalence proofs. The imposed safety policy has no bearing on the security of property in this framework. Rather, as a safety precaution, IP users and bitstream providers reached an agreement to utilize the propositional calculus to construct and verify proofs, the conjunctive normal form to describe combinational processes, and the same bitstream formats.

In general, formal verification assures that the intellectual property (IP) core is the same as its specifications. On hardware, and IP cores, author Yier Jin et al. developed a novel proof-carrying hardware (PCH) system [2, 60]. Also, for IP trustworthiness assessment, offer a formal protection strategy based on SAT solvers. Depending on the PCH framework, a new trustworthy IP gathering and delivery protocol is proposed [60]. The set of security attributes is the most significant component of the PCH framework defined in [62]. A comprehensive set of attributes will improve core IPs trustworthiness by identifying wrong logic if it exists. There are three different methods of verification based on it will be processed.

(i) Property verification: here, each of the requirements is mentioned as an assertion in the test bench is verified.
(ii) Equivalence verification: checking the equivalence of the register transfer level (RTL) code gate-level netlist and GDSll file.
(iii) Model verification: it represents the system model. Ex: C, C++, HDL, and Verilog) and verify the desired behaviors and their properties. During verification, one of the statements is false; the circuit is assumed as unsecured.

RT-level code security is also verified using a SAT solver, in addition to proof-carrying approaches [67]. For example, we provided a four-step approach for filtering and locating suspect logic in third-party IPs. First, signals that were simple to identify were eliminated by employing functional vectors created by the successive Automatic Test Pattern Generation process (ATPG). The following step involved identifying hard-to-excite and/or spread signals using the full-scan N-detect ATPG. In the third phase, the suspicious netlist, which included the signals that were triggered strangely, was compared to the netlist of the circuit displaying correct behavior using the SAT solver in order to reduce the number of suspected signals and identify the genuine gates associated with the Trojan. Clusters of unverifiable gates in the circuit were found to employ the region separation method on the suspicious signals list during the penultimate step of the investigation. Zhang, X. et al. proposed a a multi-phase approach that included sequential ATPG use, redundant circuit reduction, equivalency analysis, code coverage analysis, and assertion-based verification for the identification of questionable signals [16]. This approach's efficiency in detecting Trojan signals measured between 67.7% and 100%, as proven on an RS232 circuit. Furthermore, side-channel signal analysis and optical detection are quite comparable methods. This establishes whether HT-based signals produced by the IC during the procedure are present in the design, such as heat and light. The Picoseconds Imaging Circuit Analysis (PICA) tool is used in this method. Which is adaptable to observed circuit change [58]. Here, light emission is an optical signal based on the verification process that is running. Whenever the light emission density and emission change at the time, it is assumed, that it may be untrusted. Finally, screening the HT is concerned with detecting light changes in the emission map.

3 HT Prevention Techniques

In HT prevention is mainly focused on avoiding the attacks from inserting the Trojan inside of the design. Some of the HT prevention methods are shown in Fig. 6, Hardware (design) obfuscation and layout filling; split manufacturing concept; trust in the design, and locking method.

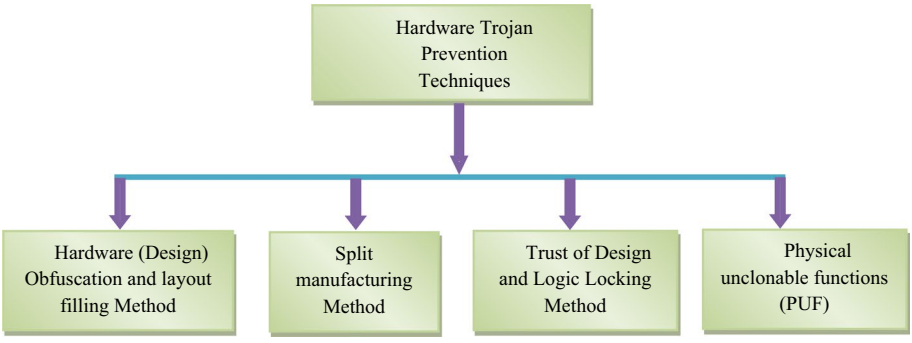

Fig. 6 Classification of Prevention of HT

3.1 Hardware (Design) Obfuscation and Layout Filling Method

Hardware obfuscation is the method of making a design more difficult to understand. Hardware obfuscation is a convenient way to protect unencrypted IPs from attackers. Changing the logic or functionality of a circuit in a controllable method through hardware obfuscation. Hardware obfuscation allows for the design to be successfully hidden and disabled while still allowing for structural testing and static/dynamic parameter analysis [68, 69]. Because of its ease, obfuscation is a popular security solution and a developing study issue. Great range in recent years, obfuscation strategies, and obfuscation attacks have been presented.

Chen Dong et al. approached, the new design with a circuit modification of a new one with the same function or behavior as the original circuit. It is much harder for hackers as obtaining reverse engineering, due to a lack of grasp of the internal logic. Generally, design obfuscation is used to add to the circuit layout that has no meaningful behavior or functionality of the existing design. These additional circuits are mainly used to prevent attacks from third-party access [1]. In a Layout filling technique, a functional design is added to the layout of the chip. It is named OBISA and stands for obfuscated self-built authentication [70]. This circuit combines the standard model to cause misunderstanding. So this confusion is based on eliminating the filing space and identification of own. Generally, the BISA (Built-in self-Authentication) technique involves putting a circuit in every vacant place that can easily test itself. Normally, HT has restrictions on space-filling or space living [1]. This filling concept adequately utilizes the empty cell in the design. Injecting the design identified or eliminated the empty spaces to restrict the injection space of the HT. Moreover, with additional layouts created, an adversary can't implant the HT, because it prevents attacks from others.

3.2 Physical-Unclonable Functions (PUF)

Physical-Unclonable functions (PUF) are a hardware-specific security technique that provides secure functionality for encrypted communication between embedded devices. Due to changes in the production process, The PUF's physical configuration is taken into easy to create but difficult or impossible to replicate. However, a vast analytics community feels that hardware nature PUF has cleared the path for its use in delivering reliable safety. This PUF leverages device mismatches caused by the process variation to create a unique identity and the response of each chip [71]. Hardware area protection is orthogonal to the strengthening of safety fundamentals, which have been studied because they are less expensive and have more accuracy than software programs. The physically unclonable function (PUF) is the main prototype, which employs a tool that does not recognize the created by technique alterations to generate unique identities for every chip as well as consistent risk response pairings. We can evaluate the PUF design using a variety of metrics.

(i) Randomness: It indicates the response to input test patterns. (ii) Uniqueness: specifies identity indication. (iii) Enhanced Security: Measure the security checks to see whether they will affect or not affect the circuit [72]. Various attack types were enhanced, including device knowledge and equipment modeling, and we were able to break that physically unclonable function design based on the physically unclonable function response. Many strategies were proposed in order to improve the measurement of error-correcting algorithms. Various reviews explored the evaluations of physically unclonable functions [53].

Figure 7 shows the various types of PUFs, Non-silicon PUF type includes all PUF types that aren't silicon PUFs and are made from non-electronic components due to their inherent mismatch. Here are several non-silicon PUF examples that rely on mismatches created by physical systems rather than electrical discrepancies. Pappu, R. et al. introduced the major non-silicon PUF named Optical PUF (O-PUF), this is founded on the response (scattering) that a laser beaming on a transparent epoxy wafer loaded with bubbles produces [73]. Silicon PUFs look for and exploit IC incompatibilities to create a pattern or function that may be used to identify electronic circuits [74].

The same circuit die is usually used to make silicon PUFs, with designs differing based on mismatch capabilities, demands, architecture, and challenge-response pair (CRP) performing techniques. There are two types of silicon PUFs: delay and memory-based PUFs. Utilizing the propagation delay of the circuit paths and the speed at which the microelectronic circuitry transition their results to 0 or 1 are the main objectives of delay-based PUF [75]. Further delay PUFs are divided into two categories. Arbiter PUF (A-PUF) and ring oscillator PUF (RO-PUF). The A-PUF is one of the first proposed designs that rely on circuit path delay, as stated by Lee et al. [76]. Considering that the A-PUF mechanism is linear, Lee et al. investigated the creation of a timing module that may mimic it in the event of a security attack. To change the module parameters, the attacker will require, a significant number of CRPs will be required by the attacker. The number of CRPs obtaining such a large quantity of CRPs is difficult because CRPs are normally kept in a very secure location. According to, the output of a PUF circuit can be adjusted by adding XOR to output bits or applying feed-forward circuits (FFCs) to an A-PUF design making it non-linear. Suh and Devadas proposed RO-PUF that is dependent on production changes in oscillator frequencies [71]. Due to the general fabrication process, the frequency parameter mismatches between identical RO-PUF. Mismatches are inevitable in the manufacturing process, resulting in random PUF values.

Focusing on the memory aspect is the memory-based PUF model if its mismatches that result in a random value at the start-up stage. In contrast to latency based PUFs, which call for specific designs for various PUF types, these PUFs have relied on memory cells and this are frequently found in FPGAs. The signature and identification for these circuits can

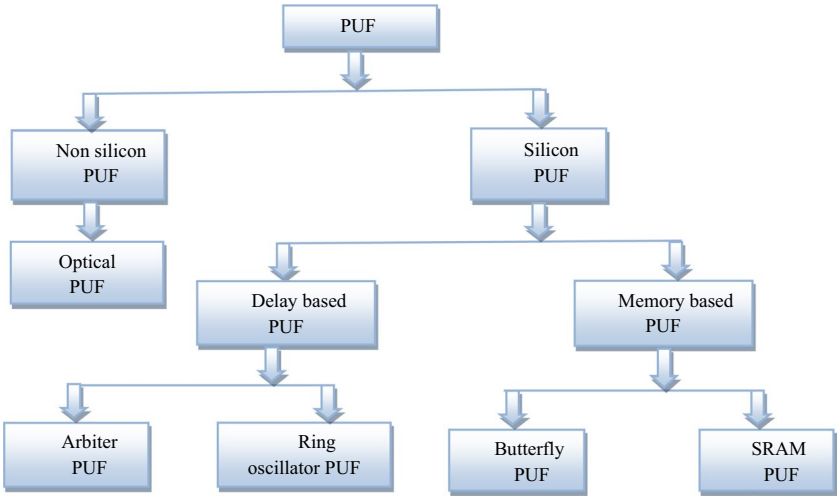

Fig. 7 Various types of PUFs

be generated using the memory chip in the circuit [77]. Butterfly PUF (B-PUF) is based on bistable elements. It's more suited to creating hidden keys. Because the length of all electrical paths will be the same, the manufacturing process is accurate. It will take some time to settle down.

SRAM-PUF:

(Holcomb, D.E. et al., were the first to introduce the Static Random Access Memory (SRAM)-PUF or (S-PUF), which gets its randomness from the startup state. Once the SRAM cell is activated, one of the stable states is selected based on the manufacturing process variances of its components [77, 78]. It can also be utilized for randomization because it is unpredictable.

SRAM cells are classed as follows based on these differences:

(i) Cells that have a strong tendency to power up as 0 or 1.
(ii) Cells with no strong tendency to power up as 0 or 1.
(iii) Cells with a strong inclination to power up as 0 or 1;

Designers do not need to build specific hardware because many electronic systems already use SRAM as a rapid memory. The major portion of cells is more likely to start up in state 1 than in state 2. As a result, Inter Hamming distance (Inter-HD) has a percentage of 27.62%. It also revealed Intra-HD within 4%. SRAM PUFs were first established [79]. Vijayakumar et al. [80] designed SRAM PUFs made up of cross-coupled inverters linked by access transistors and often settle into a "0" or "1" state consistently due to inherent manufacturing variances. Cambou et al. and Helfmeier et al. explained, that as the SRAM was turned on, its initial values for the cells were also used to establish a unique fingerprint. When SRAMs transition states, they emit energy that can be recognized by using a signal analyzer to verify the wavelength of the laser. Once this side-channel data is exposed, an attacker will have enough information about the device to replicate it [81, 82].

In order to leak information, M. A. Bokor Siddik and S. H. Alam introduced a novel approach using a new type of HT based on physical unclonable function (PUF). The purpose of this effort is to increase awareness of the advanced form of HT while analyzing and assessing the effectiveness of the PUF-based HT. They used a full-custom design flow to reduce the size of the HT while maintaining its layout. With only 749 FETs needed and an area of 10.6437 mm2, this lightweight solution shows a high barrier to detection by traditional testing and inspection techniques. Furthermore, the suggested PUF-based HT's signature is physically encrypted, preventing unauthorized HT activation even in the event that the trojan was discovered. The authors ran physical tests on an FPGA and simulations using Tanner EDA to evaluate the viability of our PUF-based HT. [83].

The architecture of the XOR arbiter PUF (XORAPUF) with the 3 characteristics of uniformity, uniqueness, and reliability was proven by Naveenkumar et al. The field programmable gate array (FPGA) version of the XORAPUF, according to experiments, obtains an inter-chip hamming distance (HD) that is closer to 50% with good uniqueness and uniformity of 48.74% and 49.88%, respectively. Additionally, the planned PUF's reliability is tuned to 99.20%. After comparing these outcomes with those of other traditional PUF, they deduced that the XORAPUF circuit produced superior outcomes [84].

Reconfigurable interconnection networks, a vital component of FPGA-based CNN accelerators, are the target of a hardware Trojan created by Hou J. et al. More specifically, when the hardware Trojan is activated, it modifies the data routes, leading to faulty connections in the arithmetic circuit and, ultimately, inaccurate convolutional

computations. In order to protect the reconfigurable interconnection network against hardware Trojan assaults, the author presents a novel detection technique based on physically unclonable functions (PUFs) to handle this problem. The experimental results show that the suggested hardware Trojan can significantly reduce the inference accuracy of common neural network architectures such as LeNet, AlexNet, and VGG by a range of 8.93% to 86.20%, just by adding a small 0.27% hardware overhead to the accelerator. In a reconfigurable interconnection network, the implemented arbiter-PUF circuit on a Xilinx Zynq XC7Z100 platform successfully identifies the location and existence of hardware Trojans. This study demonstrates how reconfigurable CNN accelerators are susceptible to hardware Trojan attacks and offers a promising detection method to reduce any security threats. The results highlight how crucial it is to take hardware security issues into account when designing and implementing AI systems that use FPGA-based CNN accelerators [85].

3.3 Split Manufacturing Method

The split manufacturing concept secures the design (or) model by hiding important data from the foundry level itself [86]. And the fundamental goal of the split manufacturing process is to split the design into two different parts. Because the manufacturing before splitting the two different circuits, one is consisting of a transistor, and routing wires and another one is consisting of some other components with routing wires. And finally, it will be fabricated in various industries. So these parts are named as (FEOL) front end of the line and (BEOL) back end of the line. The bottom layer, where the transistor is built is named FEOL; the top layer, where the metal layer is built is named BEOL. When comparing split manufacturing with design for trust, split manufacturing always has a larger manufacturing cycle and high manufacturing cost with the design for trust [1]. In split manufacturing, unsecured fabrication parties do not interfere in the BEOL process. It hardly judges the location of injected HT on the device [24]. This approach is expensive, till now split manufacturing into Two-dimensional (2D) and Three-Dimensional (3D) levels [87]. Presented a manufacturing process standard model with a split manufacturing process with the complex integrated circuits [88]. The split manufacturing alone, Will does not assure the secured production of chips [89]. Xie et al. suggested two attacks in the physical design flow, known as Proximity type and SAT type in the 2.5 D split manufacturing process. These two attacks can be avoided by adding layout and functionality to the IC. It can prevent that kind of attack. This can prevent almost all hackers from interfering with the key data of the BEOL and FEOL signals [90].

3.4 Trust of Design Method

It is generally used by injecting/ adding some extra circuit design or obfuscation components. And also improving layout filling. The drawback is high extra overhead and high precision application; it is suitable for avoiding the HT. Dupuis et al., suggested two ideas to avoid the HT. The first one is Obfuscation; here the IC functionality and behavior are hidden. Moreover, the blurred images result from confusion in the circuit architecture, so hackers won't be able to do it easily. The standard original circuit will not be affected by the HT [30]. The second concept is the locking mechanism; here the locking concept is injected into the original circuit to hide that circuit behavior. Presented ending piracy of integrated circuits (EPIC) techniques for securing the IC via locking mechanism with

minimal overhead [91]. Rajendran et al., implemented the technique in this technique, each chip has specific unique identity numbers. Based on that unique number it is activated, so possibly injecting HT is difficult. Presented a logic encryption concept to avoid attacks from injecting the HT [92].

Logic locking is a measure to avoid piracy and counterfeiting of outsourced IC designs by deceitful foundries. To obscure the original functionality of an IC design, new key-controlled logic gates (key-gates), key inputs, and on-chip memory are incorporated. Figure 8, represents the key gate that can be constructed using XOR/XNOR gates, to constitute the encoded circuit. Whenever the right key is supplied a locked IC performs its expected correct functions.

The worth of the right keys K1, K2, and K3 are 101. This technique is vulnerable to an attack from the satisfiability (SAT).

The two ways of logic encryption are Combinational encryption (i.e., modifying the netlist at the gate level) [69, 93] and sequential encryption (i.e. state transition graph alteration) [94]. In the first scenario, extra logic gates have been added to hide the design's original function when the erroneous key is pressed. In the second scenario, the state transition has been changed such that the design can only reach a valid state and then use the correct input sequence. Roy. et al. [69] propose a combinational encryption scheme. It entails putting XOR/XNOR gates at random into the design. One input of each of these gates is connected to a newly added key input. A second strategy is described in improvement by ensuring that the outputs are not corrupted by incorrect keys [93]. The locations of the gates XOR/XNOR operations were intended to offer a 50% of the Hamming distance (HD) among the accurate and inaccurate values.

R. S. Chakraborty. et al. Present an Obfuscation design approach that alters state transition processes to generate two functional modes: regular and obfuscated, the key (a particular input sequence) and the IC are both in obfuscated mode by default permits it to switch to normal mode [94, 95]. In addition to preventing excess production, it assists in blocking HT injected by, for example, constructing some HTs harmless (individuals who work well in an obfuscated manner). Lookup table (LUT) s can also be used as key gates replaced by using XOR/XNOR gates [96]. The contents of the LUTs function as a secret key and, as a result, logical barriers. Data flows as anticipated within the architecture when LUT programming is done correctly. Later, strategies it was presented on how to increase the effectiveness of locking systems using very large scale integration (VLSI) testing concepts, In other words, an invalid key

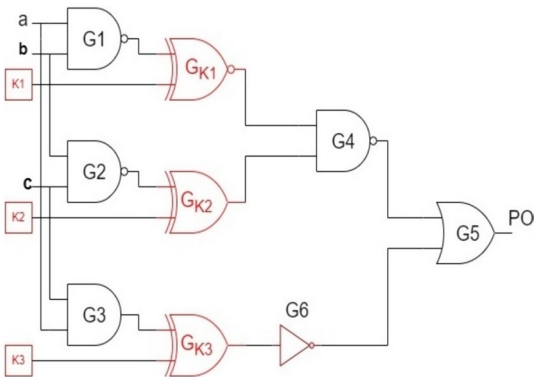

Fig. 8 demonstrates an encrypted circuit with an XOR/XNOR key

results in incorrect output. Rajendran et al., Yasin et al. and Chakraborty et al. Examining the most widely used logic locking methods and vulnerabilities in literature. Classifying current attacks and defenses to determine how the protection strategies are effective against various types of attacks. , developing novel logic locking applications that go beyond just safeguarding functionality, outlining prospective research directions and summarizing current trends in the area of logic locking [97, 98].

Table 6 represents the various prevention measures based on the PUF and it mentions the attack levels and potential countermeasures also given.

Table 6 suggests (Merli et al. [102] and Helfmeier et al. [103]) SCA-based key generation parameter, the design obfuscation is better prevention of HT. Also, it proves that photon emission analysis can fully and linearly characterize the traditional arbiter PUF and its popular extended variant (i.e. feed-forward with XOR enhancement). Tajik et al. suggested, that the actual results of implementing a complicated programmable logic chip built by the arbiter PUF using the 60 nm process and machine learning methods are introduced. In addition, with this technology that there is no need for a PUF response, our physical delay extraction does not need to know the actual PUF response. The attack has a high level of severity, and countermeasures are to effectively characterize the relying on the arbiter PUF, the minimum amount of measurements [99]. Rührmair et al. implemented dynamic and differential CMOS logic techniques, in this method the power side channel is noisier than the timing side channel, resulting in faster convergence. Potential countermeasures are to avoid repetitive measurements [100].

Merli et al. proposed a local electromagnetic attack, which can destroy the RO-PUF implementation of the previously proposed countermeasure protection, can identify attacks. It represents a prerequisite for local electromagnetic attacks on RO-PUF. To protect RO PUF equipment, it proposes measurement path randomization and staggered placement and design confusion as countermeasures, while the latter does not require any additional resources [101]. Physical cloning is not the same as modeling to characterize PUF. The PUF protocol's physical response is an invariant feature of the PUF protocol. Helfmeier et al. developed a physical clone of PUF with the same appearance were successfully made. In the arrays of high-density SRAM cells determining PUF response is difficult and physical cloning is a very successful method of production. Here the countermeasure is a design obfuscation, which is minimal in cost, has a lower PUF response density, and has a better distribution in SRAM [102].

Rührmair et al. introduce a new attack model against the Strong PUF protocol, including the "PUF reuse model" and "bad PUF model". These frameworks represent an attack strategy that is relevant and difficult to detect and is closely related to actual PUF usage scenarios. A potential countermeasure is to respond to erasable PUF [53, 103]. Countermeasures to our attacks. Zeitouni et al. and Gao et al. improved SRAM PUF resistance by constructively using remanence attenuation. It is based on the attenuation of residual magnetism in SRAM memory and is used as a timing mechanism for clock-less devices with low power consumption. The key to prevention is the constant-time response readout and the confusion of the SRAM power-on state [104, 105].

4 Hardware Trojan Prevention Algorithm

To develop and test HT detection and prevention, different certified algorithms were chosen for different encryption strategies. The various detection and prevention algorithms of HT are given below in Table 7. In Table 7 detection of HT algorithms are mixed-feature

Table 6 Potential countermeasures to attacks on regularly used PUFs

References	Methods	Attack level			Affected PUF			Attack difficulty	Potential countermeasure	
		Authentication	Key generation	Key Exchange	Arbiter based PUF and Lightweight PUF	Ring Oscillator (RO) PUF	SRAM PUF		Design obfuscation	Others
[99]	Machine learning	✓			✓			High		The arbiter PUF can be effectively characterized with a small amount of measures
[100]	Machine learning	✓			✓			Medium		Repeated measurements are avoided owing to dynamic and differential CMOS logic
[101]	Side-channel analysis			✓		✓		Medium	✓	-
[102]	Side-channel analysis			✓			✓	High	✓	-
[103]	PUF reuse			✓	Strong PUFs	Low		Response erasability (erasable PUF)		
[53]	Machine learning	✓			✓			Low	✓	Adversary machine learning

Table 6 (continued)

References	Methods	Attack level			Affected PUF			Attack difficulty	Potential countermeasure	
		Authentication	Key generation	Key Exchange	Arbiter based PUF and Lightweight PUF	Ring Oscillator (RO) PUF	SRAM PUF		Design obfuscation	Others
[104, 105]	Side-channel analysis			✓			✓	Medium	✓	Response readout time is constant, SRAM power-up state obfuscation
[106]	Machine learning	✓			✓			Low	✓	Adversary machine learning

Table 7 Various HT detection and Prevention algorithms

S. No	Algorithm	Detection/prevention	Description	References
1	Mixed-feature gene expression programming (or) mixed-feature GEP	Detection	It can be used to locate hardware Trojans and reduces the amount of isomorphic circuits	[107]
2	Support vector Machine–(SVM)	Detection	Classification of Trojans using this supervised machine learning tool. It can categorize a designate if the file is Trojan-infected or Trojan-free, or classify the Trojan infection kind using a two-class SVM	[108, 109]
3	K-Nearest Neighbor–(KNN)	Detection	It determines the label of the test sample by using the K-nearest training samples	[108, 110]
4	Statistical characteristics and the approach of distance learning	Detection	On genuine chips, With 10–5 TCR, the lowest (Trojan to circuit ratio) TCR, has a good detection effect	[108, 111]
5	Unsupervised Learning Approach	Detection	Using a method known as Principal Component Analysis (PCA), which reduces dimensionality and facilitates computation, this work aids in the detection of trojans	[112]
6	Unsupervised Detection Method using the local outlier factor (LOF) algorithm and principal component analysis (PCA)	Detection	Unsupervised hardware The principle component analysis (PCA) and local outlier factor (LOF) algorithm coupled to create the PL-HTD Trojans detection methodology demonstrates the viability and effectiveness of hardware Trojans identification by using a method independent of class label information	[113]
7	Dual Discriminator Assisted Conditional Generation Adversarial Network (D2ACGAN)	Detection	According to the works, the D2ACGAN-based hardware Trojan classification model can achieve an average detection accuracy of 97.08%, which is better than detection models based on CNN, SVM, and other models	[114]
8	Assignment of decoys to target constants algorithm	Prevention	An obfuscation approach that hides sensitive constants among decoy constants employing additional logic with keyed inputs to conceal them from an adversary at an untrustworthy foundry. At the RTL (register-transfer level), the suggested approach implements the obfuscation of the TMCM process (RTL)	[115]

Table 7 (continued)

S. No	Algorithm	Detection/prevention	Description	References
9	Partitioning algorithm	Prevention	By examining the weighted observability, the FIR filter is classified as critical or non-critical. The critical and non-critical gates are discovered using this approach	[116]
10	Key driven structurally obfuscated(or) key-based structural obfuscation	Prevention	This key-driven, architecturally obfuscated architecture is difficult to decipher for an attacker, making identification and understanding difficult. PSO-OT achieves excellent hardware security and low area, power, and performance overheads	[117]
11	Hierarchical contiguous folding (HCF)	Prevention	To obscure this DSP circuit, a design obfuscation technique based on the HCF algorithm was added to the original 18th order IIR filter	[118]
12	Transformation algorithm	Prevention	The algorithm transformation approaches do not modify the system's input–output behavior, but they do change the algorithm's underlying structure to boost concurrency	[119]
13	Iterative ranking algorithm	Prevention	Netlist Level Obfuscation for Authentication. After the metric for the nodes in a gate-level design has been calculated, an iterative ranking method is used to rank the nodes	[120]

GEP, (SVM) Support Vector Machine, (KNN) K-Nearest Neighbor, and Statistical features and distance learning methods. HT prevention algorithms include the assignment of decoys to target constants algorithm, partitioning algorithm, Key-based structural obfuscation, Hierarchical Contiguous Folding (HCF), Transformation algorithm, and Iterative Ranking Algorithm.

Mohri et al. and Sun et al. suggest statistical features and the distance learning method is better than the existing methods above three detection methods [108, 109]. And Sengupta et al. author suggested a PSO (particle swarm optimization)-based Obfuscation Technique (PSO-OT) system that provides the optimal option for key insertion positions and methods [117]. To frustrate the embedded field and also care for third-party IP cores, our PSO-OT achieves excellent hardware security and low area, power, and performance overheads so the key-driven is structurally obfuscated (or) Key-based structural obfuscation is a better prevention method compared to other existing prevention methods.

5 Challenges and Future Scope

Sun et al. proposed a detection approach to select the ideal test vector and do multiple sampling for the chip's side-channel signal only. Other circuit detection data categories, such as IC image-based detection and circuit physical characteristics-based detection are major challenges [111]. Yuan et al. sugsuggest the future, to boost the success rate of detection techniques, we need research in superior test-bench generation technology. This technique will become more advanced if the circuit scale is increased by a large order of magnitude. They tried to figure out the best technique for selecting elements from the P and Q sets so that the difference between them is as small as possible [121]. Becker et al. analyzed the HT in the semiconductor. It is put into existing transistors by altering the polarity of doping. This sort of HT does not necessitate any new circuit primitives, and the IC's appearance and functioning remain unchanged when the HT is turned off. When the infected circuit's temperature reaches a particular high, the HT is executed [122]. The circuit's polarity is deflected at that time, resulting in an anomaly in the IC's function. Optical detection and side-channel analysis detection approaches are currently proving to be nearly hard to detect as HT. Logic testing is a method of comparing the test answer to the expected response based on simulation calculations. HTs have recognized if the IC's responses changed during the testing process [45]. The fundamental issue in a logic-testing-based VLSI approach is that it is computationally impossible to generate a comprehensive set of test vectors for detecting all possible HTs [5].

Process variation and noise issues are one of the major challenges with hardware Trojans. It is prevented by present golden chip-free hardware based on the power side-channel technique for detecting Trojans. This technique adapts the original concept to combat the issue that a similar structure can readily be an adversary, has bypassed it, and is unable to cover the entire circuit. It took advantage of the fact that the number of logic gate toggles is related to physical power usage. For self-reference detection, it has two circuits with the same toggles for particular inputs. The adversary's complexity in defeating this detection technique is theoretically guaranteed to be O (2(n/2) log2 (n/2). the author demonstrates the ability to reduce process variation through simulation tests. Also, Yuan et al. proposed solution become more sophisticated as the scale of the circuit is increased up to a higher order of magnitude. To solve this issue, use the proper procedure to minimize the difference between them while selecting entries from the P and Q sets [121, 122]. He

et al. suggested, a new technique for detecting HT based on electromagnetic side-channel spectrum modeling and analysis they used design data early in the life cycle of IC, and the resulting spectrum can be used as a gold standard, eliminating the requirement for produced golden chips [44]. Another noteworthy characteristic is that this approach is theoretically impervious to process fluctuation. The suggested method may efficiently detect Trojans even with extremely slight traces, according to experiments on selected advanced encryption standard benchmark circuits using FPGA have yielded promising results.

Sharma and Ranjan used machine learning to build a unique method to increase fault coverage in hardware Trojan detection. In order to find patterns suggestive of Trojan behavior, they developed a kind of convolutional neural network (CNN) over a dataset of circuit responses, comprising timing and power consumption metrics. In comparison to conventional testing methods, experimental results on a variety of Trojan-infected circuits demonstrate an average improvement of 25% in fault coverage and a 40% reduction in mean time to detect (MTTD) [123]. Ashutosh Ghimire et al. employed a thorough methodology based on redundancy and diversity techniques to detect hardware Trojans. They seek to improve fault coverage by utilizing triple modular redundancy (TMR) and various redundant designs, such as scan chain modification and clock gating. Comparing experimental assessments on different Trojan-infected circuits to non-redundant testing techniques, the average fault coverage is improved by 20%, and the mean time between failures (MTBF) is decreased by 30% [124].

Gor Piliposyan and Saqib Khursheed used real-time hardware Trojan identification through in-field testing and monitoring. They continuously evaluated circuit behavior by integrating runtime monitoring procedures, such as power analysis and electromagnetic emission analysis, with built-in self-test (BIST). According to experimental results in a simulated environment, the methodology reduces the false positive rate (FPR) by 25% while improving fault coverage on average by 30% when compared to offline testing methods [125]. Protecting Internet-of-Things (IoT) devices from hardware Trojans has become essential due to their rapid growth. A machine learning-based anomaly detection method for locating hardware Trojans in Internet of Things devices was recently proposed by Yu, S et al. They used an SVM classifier to classify the features they extracted from measurements of power use. Using a dataset of 100 IoT devices, this technique achieves an average detection accuracy of 97.3% with a false positive rate of 2.1% [126].

Preventing the installation of hardware Trojans during the manufacturing procedure requires ensuring the security and integrity of a supply chain. M. S. U. I. Sami et al. suggested an authentication system based on blockchain technology to protect the supply chain and stop unauthorized changes to integrated circuits (ICs). A blockchain is used to record every step of the IC manufacturing process, giving rise to an unchangeable log of all transactions and changes. The simulated outcomes show that 100% of attempted hardware Trojan insertions are effectively detected and prevented by our method. The findings showed that the supply chain had 10 entities, a 100% detection and prevention rate, a 6-month simulation period, 5000 records on the blockchain, and a 15% cost reduction in supply chain management due to increased operational efficiency of 20% [127]. Hardware Integrated circuits (ICs) are vulnerable to security breaches due to trojans' ability to introduce malicious features that jeopardize system integrity. Jingxin Zhong et al. used thermal signature analysis to create a revolutionary hardware Trojan detection technique. This method makes use of the fact that hardware Trojans' changed behaviour might result in localized heating in integrated circuits. They find Trojans by employing infrared thermography to analyze temperature fluctuations. Experimental results show an average detection accuracy of 96.4% with a false positive rate of 2.8% on a variety of benchmark circuits, including the ISCAS-85 and ISCAS-89 benchmarks [57].

A deep learning-based method for hardware Trojan identification via thermal profiling was presented by Yan et al. This technique analyzes thermal pictures of ICs using convolutional neural networks (CNNs) to find unusual temperature patterns that point to the presence of Trojans. They test on multiple benchmark circuits and show a 97.2% average detection accuracy and 1.5% false positive rate [128].

It is extremely difficult to predict when the HT will be triggered because the HT activation condition is always an uncommon event. Because there are so many different varieties of HT, it's difficult to determine which technique will work best. One of the challenges with logical testing detection approaches is that they rely on golden references only to identify the HT.

6 Conclusion

In conclusion, the new paradigm in hardware trojan detection and prevention is represented by the confluence of obfuscation, PUFs, and AI. Stakeholders can protect vital electronic systems from trojan attacks and strengthen ICs by utilizing the beneficial relationships between these strategies. To ensure that these strategies are effective in the face of changing threats, more research and cooperation are necessary to further optimize and refine them. This paper deals with various HT detection methods like Pre-silicon, Post-silicon-based detection, formal verification, physically unclonable function, and optical detection. Also, discussed the HT prevention techniques such as Hardware (design) obfuscation, split manufacturing, and trust of design. Evaluations show that Pre-silicon (run time monitoring), which requires less space, power head, and outside interference, is more effective at HT detection. Logical testing utilizing probability signature inspection is effective in post-silicon detection because it is resistant to overhead and interference from process variations. In prevention technique, utilization of layout filling effectively achieved in SCA-based hardware (design) obfuscation, it leads to better prevention of HT. Moreover, various prevention algorithms, detecting challenges and effects of the HT in the recent application are summarized with its solutions.

Funding The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

Human and Animal Rights Not Applicable.

References

1. Dong, C., Xu, Y., Liu, X., Zhang, F., He, G., & Chen, Y. (2020). Hardware Trojans in chips: a survey for detection and prevention. *Sensors (Basel, Switzerland), 20*(5165), 1–37. https://doi.org/10.3390/s20185165
2. Jin, Y. (2015). Introduction to hardware security. *Electronics, 4*, 763–784. https://doi.org/10.3390/electronics4040763

3. Liu, Y., Jin, Y., Nosratinia, A., & Makris, Y. (2017). silicon demonstration of hardware Trojan design and detection in wireless cryptographic ICs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25*, 1506–1519. https://doi.org/10.1109/TVLSI.2016.2633348
4. Naveenkumar, R., Sivamangai, N. M., Napolean, A., & Janani, V. (2021). A survey on recent detection methods of the hardware trojans. In *2021 3rd International conference on signal processing and communication (ICPSC)* (pp. 139-143). IEEE. https://doi.org/10.1109/ICSPC51351.2021.9451682
5. Bhunia, S., Hsiao, M. S., Banga, M., & Narasimhan, S. (2014). Hardware Trojan attacks: threat analysis and countermeasures. *Proceedings of the IEEE, 102*, 1229–1247. https://doi.org/10.1109/JPROC.2014.2334493
6. Hughes, L. A., & DeLone, G. J. (2007). Viruses, worms, and Trojan horses: serious crimes, nuisance, or both? *Social Science Computer Review, 25*(1), 78–98. https://doi.org/10.1177/0894439306292346
7. Rahman, M. A., Rahman, M. T., Kisacikoglu, M. C., & Akkaya, K. (2020). Intrusion detection systems-enabled power electronics for unmanned aerial vehicles. *IEEE CyberPELS (CyberPELS), 2020*, 1–5. https://doi.org/10.1109/CyberPELS49534.2020.9311545
8. Hu, N., Ye, M., & Wei, S. (2019). Surviving information leakage hardware Trojan attacks using hardware isolation. *IEEE Transactions on Emerging Topics in Computing, 7*(2), 253–261. https://doi.org/10.1109/TETC.2017.2648739
9. Deyati, S., Muldrey, B. J., & Chatterjee, A. (2016). Trojan detection in digital systems using current sensing of pulse propagation in logic gates. In *2016 17th International Symposium on Quality Electronic Design* (*ISQED*) (pp. 350-355). IEEE. https://doi.org/10.1109/ISQED.2016.7479226
10. Ahmed, N., Tehranipoor, M., & Jayaram, V. (2006). Timing-based delay test for screening small delay defects. In *Proceedings of the 43rd annual Design Automation Conference* (pp. 320-325). https://doi.org/10.1145/1146909.1146993
11. Cheng, K.-T. (1993). Transition fault testing for sequential circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 12*(12), 1971–1983. https://doi.org/10.1109/43.251160
12. PCI-SIG. PCI-Express Base Specification (2009), 3. Revision 2.1
13. Bhunia, S., MironAbramovici, D. A., Bradley, P., Hsiao, M. S., Plusquellic, J., & Tehranipoor, M. (2013). Protection against hardware Trojan attacks: towards a comprehensive solution. *IEEE Design & Test, 30*(3), 6–17. https://doi.org/10.1109/MDT.2012.2196252
14. Hennessy, A., Zheng, Y., & Bhunia, S. (2016) JTAG-based robust PCB authentication for protection against counterfeiting attacks. In *2016 21st asia and south pacific design automation conference (aspdac)* (pp. 56-61). IEEE. https://doi.org/10.1109/ASPDAC.2016.7427989
15. Dong, C., He, G. R., Liu, X. M., Yang, Y., & Guo, W. Z. (2019). A multi-layer Hardware Trojan protection framework for IoT chips. *IEEE Access, 7*, 23628–23639. https://doi.org/10.1109/ACCESS.2019.2896479
16. Zhang, J., & Gong, W. (2012). Atmospheric boundary layer observations based on raman lidar. In *International Photonics and Optoelectronics Meetings (POEM)* 2011: Optoelectronic Sensing and Imaging (Vol. 8332, pp. 169-175). SPIE. https://doi.org/10.1117/12.914769
17. Zhang, J., Tong, Y., Yang, X., Gong, J., & Gong, W. (2011, February). Detection of atmospheric composition based on lidar. In *Journal of Physics: Conference Series* (Vol. 276, No. 1, p. 012036). IOP Publishing. https://doi.org/10.1088/1742-6596/276/1/012036
18. Rosenfeld, K., & Karri, R. (2010). Attacks and defenses for JTAG. *IEEE Design & Test of Computers, 27*, 36–47. https://doi.org/10.1109/MDT.2010.9
19. JS, R., Ancajas, D. M., Chakraborty, K., & Roy, S. (2015). Runtime detection of a bandwidth denial attack from a rogue network-on-chip. In *Proceedings of the 9th International Symposium on Networks-on-Chip* (pp. 1-8). https://doi.org/10.1145/2786572.2786580
20. Vosatka, J. (2018). Introduction to hardware Trojans. In S. Bhunia & M. M. Tehranipoor (Eds.), *The Hardware Trojan War* (pp. 15–51). Cham: Springer International Publishing AG, Ltd. https://doi.org/10.1007/978-3-319-68511-3_2
21. Jin, Y., Maliuk, D., & Makris, Y. (2016). Hardware Trojan detection in analog/RF integrated circuits. In C. H. Chang & M. Potkonjak (Eds.), *Secure System Design and Trustable Computing* (pp. 241–268). Cham: Springer International Publishing Switzerland Ltd. https://doi.org/10.1007/978-3-319-14971-4_7
22. He, G., Dong, C., Huang, X., Guo, W., Liu, X., & Ho, T. Y. (2020). Htcatcher: Finite state machine and feature verifcation for large-scale neuromorphic computing systems. In *Proceedings of the 2020 on great lakes symposium on VLSI* (pp. 415-420). https://doi.org/10.1145/3386263.3406955
23. Li, H., Liu, Q., Zhang, J., & Lyu, Y. (2015). A survey of hardware trojan detection, diagnosis and prevention. In *2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)* (pp. 173-180). IEEE. https://doi.org/10.1109/CADGRAPHICS.2015.41

24. Xiao, K., Forte, D., Jin, Y., Karri, R., Bhunia, S., & Tehranipoor, M. (2016). Hardware Trojans: lessons learned after one decade of research. *ACM Transactions on Design Automation of Electronic Systems, 22*, 23. https://doi.org/10.1145/2906147
25. Khalid, F., Hasan, S. R., Hasan, O., & Awwad, F. R. (2018). Runtime hardware Trojan monitors through modeling burst mode communication using formal verification. *Integration, 61*, 62–76. https://doi.org/10.1016/j.vlsi.2017.11.003
26. Ma, P., Wang, Z., & Wang, Y. (2024). "A pre-silicon detection based on deep learning model for Hardware Trojans." *Journal of Circuits Systems and Computers, 33*(08), 2450144. https://doi.org/10.1142/S0218126624501433
27. Hasegawa, K., Hidano, S., Nozawa, K., Kiyomoto, S., & Togawa, N. (2023). R-HTDetector: robust hardware-Trojan detection based on adversarial training. *IEEE Transactions on Computers, 72*(2), 333–345. https://doi.org/10.1109/TC.2022.3222090
28. Hussain, M., Guo, H., & Parameswaran, S. (2018). A customized authentication design for traffic hijacking detection on hardware-Trojan infected NoCs. *Journal of Computer and Communications, 2018*(6), 135–152. https://doi.org/10.4236/jcc.2018.61015
29. Bao, C., Forte, D., & Srivastava, A. (2015). Temperature tracking: toward robust run-time detection of hardware Trojans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34*, 1577–1585. https://doi.org/10.1109/TCAD.2015.2424929
30. Dupuis, S., Flottes, M., Di Natale, G., & Rouzeyre, B. (2018). Protection against hardware Trojans with logic testing: proposed solutions and challenges ahead. *IEEE Design & Test, 35*, 73–90. https://doi.org/10.1109/MDAT.2017.2766170
31. Salmani, H., Tehranipoor, M., & Plusquellic, J. (2012). A novel technique for improving hardware Trojan detection and reducing Trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20*(1), 112–125. https://doi.org/10.1109/TVLSI.2010.2093547
32. Zhang, J., Yuan, F., Wei, L., Liu, Y., & Xu, Q. (2015). VeriTrust: verification for hardware trust. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34*, 1148–1161. https://doi.org/10.1109/TCAD.2015.2422836
33. Xue, M., Gu, C., Liu, W., Yu, S., & O'Neill, M. (2020). Ten years of hardware Trojans: a survey from the attacker's perspective. *IET Computers & Digital Techniques, 14*, 231–246. https://doi.org/10.1049/iet-cdt.2020.0041
34. Courbon, F., Loubet-Moundi, P., Fournier, J. J., & Tria, A. (2015, March). A high efficiency hardware trojan detection technique based on fast SEM imaging. In *2015 design, automation & test in Europe conference & exhibition (DATE)* (pp. 788-793). IEEE.https://doi.org/10.7873/DATE.2015.1104
35. Hou, Y., He, H., Shamsi, K., Jin, Y., Wu, D., & Wu, H. (2019). On-chip analog trojan detection framework for microprocessor trustworthiness. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38*, 1820–1830. https://doi.org/10.1109/TCAD.2018.2864246
36. Ghimire, A., Amsaad, F., Hossain, T., Hoque, T., & Sherif, A. (2023, August). FPGA Hardware Trojan Detection: Golden-Free Machine Learning Approach. In *NAECON 2023-IEEE National Aerospace and Electronics Conference* (pp. 181-186). IEEE. https://doi.org/10.1109/NAECON58068.2023.10365812
37. Amornpaisannon, B., Diavastos, A., Peh, L., & Carlson, T. E. (2024). Secure run-time hardware Trojan detection using lightweight analytical models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 43*, 431–441. https://doi.org/10.1109/TCAD.2023.3316113
38. Li, H., Liu, Q., & Zhang, J. (2016). A survey of hardware Trojan threat and defense. *Integration, 55*, 426–437. https://doi.org/10.1016/j.vlsi.2016.01.004
39. Amelian, A., & Borujeni, S. E. (2018). A side-channel analysis for hardware trojan detection based on path delay measurement. *Journal of Circuits, Systems and Computers, 27*, 1850138. https://doi.org/10.1142/S0218126618501384
40. Zarrinchian, G., & Zamani, M. S. (2017). Latch-based structure: a high resolution and self-reference technique for hardware Trojan detection. *IEEE Transactions on Computers, 66*, 100–113. https://doi.org/10.1109/TC.2016.2576444
41. Tang, Y., Fang, L., & Li, S. (2019). Activity factor based hardware Trojan detection and localization. *Journal of Electronic Testing, 35*, 1–10. https://doi.org/10.1007/S10836-019-05803-1
42. Huang, D. C., Hsiao, C. F., Chang, T. W., et al. (2022). A security method of hardware Trojan detection using path tracking algorithm. *EURASIP Journal on Wireless Communications and Networking, 2022*, 81. https://doi.org/10.1186/s13638-022-02165-9
43. Ashok, M., Turner, M. J., Walsworth, R. L., Levine, E. V., & Chandrakasan, A. P. (2022). Hardware Trojan detection using unsupervised deep learning on quantum diamond microscope

magnetic field images. *ACM Journal on Emerging Technologies in Computing Systems, 18*(4), 1–25. https://doi.org/10.1145/3531010
44. He, J., Zhao, Y., Guo, X., & Jin, Y. (2017). Hardware Trojan detection through chip-free electromagnetic side-channel statistical analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25*(10), 2939–2948. https://doi.org/10.1109/TVLSI.2017.2727985
45. Hicks, M., Finnicum, M., King, S. T., Martin, M. M., & Smith, J. M. (2010). Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. In *2010 IEEE symposium on security and privacy* (pp. 159-172). IEEE. https://doi.org/10.1109/SP.2010.18
46. Zheng, Z. X., Li, Y. F., Yu, L., Tian, Y., & Liu, Z. L. (2014). Hardware Trojan detection technology based on probabilistic signature. *Computing Engineering, 40*, 18–22.
47. Bazzazi, A., Shalmani, M. T., & Hemmatyar, A. M. (2017). Hardware Trojan detection based on logical testing. *Journal of Electronic Testing, 33*, 381–395. https://doi.org/10.1007/s10836-017-5670-0
48. Huang, Y., Bhunia, S., & Mishra, P. (2018). Scalable test generation for trojan detection using side channel analysis. *IEEE Transactions on Information Forensics and Security, 13*, 2746–2760. https://doi.org/10.1109/TIFS.2018.2833059
49. Priyatharishini, M., & Devi, M. N. (2018). Detection of malicious circuit in hardware using compressive sensing algorithm. In *2018 Second international conference on advances in electronics, computers and communications (ICAECC)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICAECC.2018.8479492
50. Popat, J., & Mehta, U. S. (2016). Transition probabilistic approach for detection and diagnosis of Hardware Trojan in combinational circuits. *IEEE Annual India Conference (INDICON), 2016*, 1–6.
51. Govindan, V., & Chakraborty, R. S. (2018). Logic testing for hardware Trojan detection. In S. Bhunia & M. Tehranipoor (Eds.), *The hardware Trojan war*. Cham: Springer. https://doi.org/10.1007/978-3-319-68511-3_7
52. Naveenkumar, R., Sivamangai, N.M., Napolean, A., Puviarasu, A., & Saranya, G. (2022). Preventive Measure of SAT Attack by Integrating Anti-SAT on Locked Circuit for Improving Hardware Security. 2022 7th International Conference on Communication and Electronics Systems (ICCES), 756–760. https://doi.org/10.1109/ICCES54183.2022.9835923
53. Grailoo, M., Leier, M., & Pagliarini, S. (2022). Hardware Trojans for Confidence Reduction and Misclassifications on Neural Networks. In *2022 23rd International Symposium on Quality Electronic Design (ISQED)* (pp. 1-6). IEEE. https://doi.org/10.1109/ISQED54688.2022.9806246
54. Chu, C., Jiang, L., Swany, M., & Chen, F. (2023). Qtrojan: A circuit backdoor against quantum neural networks. In *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.
55. Khalid, F., Abbassi, I. H., Rehman, S., Kamboh, A. M., Hasan, O., & Shafique, M. (2022). ForASec: formal analysis of hardware Trojan-based security vulnerabilities in sequential circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41*(4), 1167–1180. https://doi.org/10.1109/TCAD.2021.3061524
56. Bhasin, S., Danger, J., Guilley, S., Ngo, X. T., & Sauvage, L. (2013). Hardware Trojan horses in cryptographic IP cores. *Workshop on Fault Diagnosis and Tolerance in Cryptography, 2013*, 15–29. https://doi.org/10.1109/FDTC.2013.15
57. Zhong, J., & Wang, J. (2018). Thermal images based Hardware Trojan detection through differential temperature matrix. *Optik, 158*, 855–860. https://doi.org/10.1016/j.ijleo.2017.12.145
58. Song, P., Stellari, F., Pfeiffer, D., Culp, J., Weger, A., Bonnoit, A., & Taubenblatt, M. (2011). MARVEL—Malicious alteration recognition and verification by emission of light. In *2011 IEEE International Symposium on Hardware-Oriented Security and Trust* (pp. 117-121). IEEE. https://doi.org/10.1109/HST.2011.5955007
59. Bao, C., Forte, D., & Srivastava, A. (2014). On application of one-class SVM to reverse engineering-based hardware Trojan detection. In *Fifteenth International Symposium on Quality Electronic Design* (pp. 47-54). IEEE. https://doi.org/10.1109/ISQED.2014.6783305
60. Guo, X., Dutta, R. G., Jin, Y., Farahmandi, F., & Mishra, P. (2015, June). Pre-silicon security verification and validation: A formal perspective. In *Proceedings of the 52nd annual design automation conference* (pp. 1-6). https://doi.org/10.1145/2744769.2747939
61. Drzevitzky, S., Kastens, U., & Platzner, M. (2009). Proof-carrying hardware: towards runtime verification of reconfigurable modules. *International Conference on Reconfigurable Computing and FPGAs, 2009*, 189–194. https://doi.org/10.1109/ReConFig.2009.31
62. Love, E., Jin, Y., & Makris, Y. (2012). Proof-carrying hardware intellectual property: a pathway to trusted module acquisition. *IEEE Transactions on Information Forensics and Security, 7*, 25–40. https://doi.org/10.1109/TIFS.2011.2160627

63. Love, E., Jin, Y., & Makris, Y. (2011) Enhancing security via provably trustworthy hardware intellectual property. In *2011 IEEE international symposium on hardware-oriented security and trust* (pp. 12-17). IEEE. https://doi.org/10.1109/HST.2011.5954988
64. Necula, G. C. (1997) Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages* (pp. 106-119). https://doi.org/10.1145/263699.263712
65. Appel, A. W. (2001) Foundational proof-carrying code. In *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science* (pp. 247-256). IEEE.https://doi.org/10.1109/FITS.2003.1264926
66. Appel, A. W., & McAllester, D. (2001). An indexed model of recursive types for foundational proof-carrying code. *ACM Transactions on Programming Languages and Systems, 23*, 657–683. https://doi.org/10.1145/504709.504712
67. Banga, M., & Hsiao, M. S. (2010) Trusted RTL: Trojan detection methodology in pre-silicon designs. In *2010 IEEE international symposium on hardware-oriented security and trust (HOST)* (pp. 56-59). IEEE.https://doi.org/10.1109/HST.2010.5513114
68. Rahman, M. T., Forte, D., Shi, Q., Contreras, G. K., & Tehranipoor, M. (2014) CSST: an efficient secure split-test for preventing IC piracy. In *2014 IEEE 23rd North Atlantic Test Workshop* (pp. 43-47). IEEE. https://doi.org/10.1109/NATW.2014.17
69. Roy, J. A., Koushanfar, F., & Markov, I. L. (2008) EPIC: Ending piracy of integrated circuits. In *Proceedings of the conference on Design, automation and test in Europe* (pp. 1069-1074). https://doi.org/10.1109/DATE.2008.4484823
70. Xiao, K., Forte, D., & Tehranipoor, M. M. (2015). Efficient and secure split manufacturing via obfuscated built-in self-authentication. *IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2015*, 14–19. https://doi.org/10.1109/HST.2015.7140229
71. Suh, G. E., & Devadas, S. (2007) Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual design automation conference* (pp. 9-14).
72. Hospodar, G., Maes, R., & Verbauwhede, I. (2012) Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability. In *2012 IEEE International workshop on Information forensics and security (WIFS)* (pp. 37-42). IEEE. https://doi.org/10.1109/WIFS.2012.6412622
73. Pappu, R., Recht, B., Taylor, J., et al. (2002). Physical one-way functions. *Science, 297*(5589), 2026–2030. https://doi.org/10.1126/science.1074376
74. Gassend, B., Clarke, D., Van Dijk, M., & Devadas, S. (2002). Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 148-160).https://doi.org/10.1145/586110.586132
75. Morozov, S., Maiti, A., & Schaumont, P. (2010). An analysis of delay based PUF implementations on FPGA. In *Reconfigurable Computing: Architectures, Tools and Applications: 6th International Symposium, ARC 2010, Bangkok, Thailand, March 17-19, 2010. Proceedings 6* (pp. 382-387). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12133-3_37
76. Lee, J. W., Lim, D., Gassend, B., Suh, G. E., Van Dijk, M., & Devadas, S. (2004). A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)* (pp. 176-179). IEEE. https://doi.org/10.1109/VLSIC.2004.1346548
77. Holcomb, D. E., Burleson, W. P., & Fu, K. (2009). Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions Computers, 58*(9), 1198–1210. https://doi.org/10.1109/TC.2008.212
78. Holcomb, D. E., Burleson, W. P., & Fu, K. (2007). Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proceedings of the Conference on RFID Security* (Vol. 7, No. 2, p. 01).
79. Böhm, C., Hofer, M., & Pribyl, W. (2011). A microcontroller sram-puf. In *2011 5th International Conference on Network and System Security* (pp. 269-273). IEEE. https://doi.org/10.1109/ICNSS.2011.6060013
80. Vijayakumar, V., Patil, V., & Kundu, S. (2017). On improving reliability of SRAM-based physically unclonable functions. *Journal of Low Power Electronics and Applications, 7*(1), 2. https://doi.org/10.3390/jlpea7010002
81. Cambou, B., & Orlowski, M. (2016). PUF designed with Resistive RAM and Ternary States. In *Proceedings of the 11th Annual Cyber and Information Security Research Conference* (pp. 1-8). https://doi.org/10.1145/2897795.2897808
82. Helfmeier, C., Boit,C. Nedospasov, D. Tajik, S. Seifert, J.-P. (2014). Physical vulnerabilities of physically unclonable functions, In *Proceedings of the Conference on Design, Automation & Test in Europe, European Design and Automation Association* (pp. 1–4) https://doi.org/10.7873/DATE.2014.363

83. Siddik, M. A. B., & Alam, S. H. (2023) PUF-based Hardware Trojan: Design and Novel Attack on Encryption Circuit. In *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-5). IEEE. https://doi.org/10.1109/ECCE57851.2023.10101599.
84. Naveenkumar, R., Sivamangai, N. M., Napolean, A., et al. (2023). Design of INV/BUFF logic locking for enhancing the hardware security. *Journal of Electronic Testing, 39*, 141–153. https://doi.org/10.1007/s10836-023-06061-y
85. Hou, J., Liu, Z., Yang, Z., & Yang, C. (2024). Hardware Trojan attacks on the reconfigurable interconnections of field-programmable gate array-based convolutional neural network accelerators and a physically unclonable function-based countermeasure detection technique. *Micromachines., 15*(1), 149. https://doi.org/10.3390/mi15010149
86. Perez, T. D., & Pagliarini, S. N. (2020). a survey on split manufacturing: attacks, defenses, and challenges. *IEEE Access, 8*, 184013–184035. https://doi.org/10.1109/ACCESS.2020.3029339
87. Imeson, F., Emtenan, A., Garg, S., & Tripunitara, M. (2013) Securing Computer Hardware Using 3D Integrated Circuit (IC) Technology and Split Manufacturing for Obfuscation. In *22nd USENIX Security Symposium (USENIX Security 13)* (pp. 495-510).
88. Hill, B., Karmazin, R., Otero, C. T. O., Tse, J., & Manohar, R. (2013) A split-foundry asynchronous FPGA. In *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference* (pp. 1-4). IEEE. https://doi.org/10.1109/CICC.2013.6658536
89. Karmazin, R., Otero, C.T., & Manohar, R. (2013). cellTK: Automated Layout for Asynchronous Circuits with Nonstandard Cells. 2013 *IEEE 19th International Symposium on Asynchronous Circuits and Systems* (pp. 58–66). https://doi.org/10.1109/ASYNC.2013.27
90. Xie, Y., Bao, C., & Srivastava, A. (2017). Security-aware 2.5D integrated circuit design flow against hardware IP piracy. *Computer, 50*, 62–71. https://doi.org/10.1109/MC.2017.121
91. Roy, J. A., Koushanfar, F., & Markov, I. L. (2010). Ending piracy of integrated circuits. *Computer, 43*, 30–38. https://doi.org/10.1109/MC.2010.284
92. Rajendran, J., Kanuparthi, A. K., Zahran, M. M., Addepalli, S., Ormazabal, G., & Karri, R. (2013). Securing processors against insider attacks: a circuit-microarchitecture co-design approach. *IEEE Design & Test, 30*, 35–44. https://doi.org/10.1109/MDAT.2013.2249554
93. Rajendran, J., Pino, Y., Sinanoglu, O., & Karri, R. (2012) Logic encryption: A fault analysis perspective. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 953-958). IEEE. https://doi.org/10.1109/DATE.2012.6176634
94. Chakraborty, R. S., & Bhunia, S. (2009) Security against hardware Trojan through a novel application of design obfuscation. In *Proceedings of the 2009 International Conference on Computer-Aided Design* (pp. 113-116). https://doi.org/10.1145/1687399.1687424
95. Chakraborty, R. S., & Bhunia, S. (2011). Security against hardware Trojan attacks using key-based design obfuscation. *Journal of Electronic Testing, 27*(6), 767–785. https://doi.org/10.1007/s10836-011-5255-2
96. Baumgarten, A., Tyagi, A., & Zambreno, J. (2010). Preventing IC piracy using reconfigurable logic barriers. *IEEE Design & Test of Computers, 27*(1), 66–75. https://doi.org/10.1109/MDT.2010.24
97. Yasin, M., Mazumdar, B., Ali, S. S., & Sinanoglu, O. (2015) Security analysis of logic encryption against the most effective side-channel attack: DPA. In *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)* (pp. 97-102). IEEE. https://doi.org/10.1109/DFT.2015.7315143
98. Chakraborty, A., et al. (2020). Keynote: a disquisition on logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39*(10), 1952–1972. https://doi.org/10.1109/TCAD.2019.2944586
99. Tajik, S., Dietz, E., Frohmann, S., Dittrich, H., Nedospasov, D., Helfmeier, C., Seifert, J., Boit, C., & Hübers, H. (2016). Photonic side-channel analysis of arbiter PUFs. *Journal of Cryptology, 30*, 550–571. https://doi.org/10.1007/s00145-016-9228-6
100. Rührmair, U., Xu, X., Sölter, J., Mahmoud, A., Majzoobi, M., Koushanfar, F., & Burleson, W. P. (2014). Efficient power and timing side channels for physical unclonable functions. *CHES*. https://doi.org/10.1007/978-3-662-44709-3_26
101. Merli, D., Heyszl, J., Heinz, B., Schuster, D., Stumpf, F., & Sigl, G. (2013). Localized electromagnetic analysis of RO PUFs. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 19-24). IEEE. https://doi.org/10.1109/HST.2013.6581559
102. Helfmeier, C., Boit, C., Nedospasov, D., & Seifert, J. P. (2013). Cloning physically unclonable functions. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (pp. 1-6). IEEE. https://doi.org/10.1109/HST.2013.6581556
103. Rührmair, U., Sölter, J., Sehnke, F., Xu, X., Mahmoud, A., Stoyanova, V., Dror, G., Schmidhuber, J., Burleson, W. P., & Devadas, S. (2013). PUF Modeling Attacks on Simulated and Silicon Data. *IEEE*

Transactions on Information Forensics and Security, 8, 1876–1891. https://doi.org/10.1109/TIFS.2013.2279798

104. Zeitouni, S., Oren, Y., Wachsmann, C., Koeberl, P., & Sadeghi, A. (2016). Remanence decay side-channel: the PUF case. *IEEE Transactions on Information Forensics and Security, 11*, 1106–1116. https://doi.org/10.1109/TIFS.2015.2512534
105. Gao, Y., Al-Sarawi, S. F., & Abbott, D. (2020). Physical unclonable functions. *Nature Electronics, 3*, 81–91. https://doi.org/10.1038/s41928-020-0372-5
106. Naveenkumar, R., Sivamangai, N., Napolean, A., Sridevi, S., Priya, S., & Sivamangai, N. M. (2022). Design and evaluation of XOR arbiter physical unclonable function and its implementation on FPGA in hardware security applications. *Journal of Electronic Testing, 38*, 653–666. https://doi.org/10.1007/s10836-022-06034-7
107. Zhang, X., & Tehranipoor, M. (2011). Case study: Detecting hardware Trojans in third-party digital IP cores. In *2011 IEEE International Symposium on Hardware-Oriented Security and Trust* (pp. 67-70). IEEE. https://doi.org/10.1109/HST.2011.5954998
108. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018) Foundations of machine learning. MIT press.
109. Kulkarni, A., Pino, Y., & Mohsenin, T. (2016) SVM-based real-time hardware Trojan detection for many-core platform. In *2016 17th International Symposium on Quality Electronic Design (ISQED)* (pp. 362-367). IEEE.
110. Noor, N. Q. M., Sjarif, N. N. A., Azmi, N. H. F. M., Daud, S. M., & Ka-mardin, K. (2017). "Hardware Trojan identification using machine learning-based classification." *Journal of Telecommunication Electronic and Computer Engineering (JTEC), 9*, 23–27.
111. Sun, C., Cheng, L. Y., Wang, L. W., Huang, Q., Huang, Y., & Feng, G. (2021). A machine learning method for hardware Trojan detection on real chips. *AIP Advances, 11*, 055006. https://doi.org/10.1063/5.0038773
112. Samyukta, K., & Ramesh, S. R. (2023). Detection of Hardware Trojan Horse using Unsupervised Learning Approach. In *2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* (pp. 77-82). IEEE. https://doi.org/10.1109/DISCOVER58830.2023.10316694
113. Dong, C., et al. (2020). An unsupervised detection approach for hardware Trojans. *IEEE Access, 8*, 158169–158183. https://doi.org/10.1109/ACCESS.2020.3001239
114. Tang, W., Su, J., & Gao, Y. (2023). Hardware Trojan detection method based on dual discriminator assisted conditional generation adversarial network. *Journal of Electronic Testing, 39*, 1–12. https://doi.org/10.1007/s10836-023-06054-x
115. Aksoy, L., Nguyen, Q. L., Almeida, F., Raik, J., Flottes, M. L., Dupuis, S., & Pagliarini, S. (2021). High-level intellectual property obfuscation via decoy constants. In 2021 IEEE *27th International Symposium on On-Line Testing and Robust System Design (IOLTS)* (pp. 1-7). IEEE.https://doi.org/10.1109/IOLTS52814.2021.9486714
116. Alaql, A., Hoque, T., Forte, D., & Bhunia, S. (2019) Quality obfuscation for error-tolerant and adaptive hardware IP protection. In *2019 IEEE 37th VLSI Test Symposium (VTS)* (pp. 1-6). IEEE. https://doi.org/10.1109/VTS.2019.8758637
117. Sengupta, A., & Rathor, M. (2020). Enhanced security of dsp circuits using multi-key based structural obfuscation and physical-level watermarking for consumer electronics systems. *IEEE Transactions on Consumer Electronics, 66*, 163–172. https://doi.org/10.1109/TCE.2020.2972808
118. Lao, Y., & Parhi, K. K. (2014). Protecting DSP circuits through obfuscation. *IEEE International Symposium on Circuits and Systems (ISCAS), 2014*, 798–801. https://doi.org/10.1109/ISCAS.2014.6865256
119. Parhi, K. K. (1989). Algorithm transformation techniques for concurrent processors. *Proceedings of the IEEE, 77*(12), 1879–1895. https://doi.org/10.1109/5.48830
120. Chakraborty, R. S., & Bhunia, S. (2008). Hardware protection and authentication through netlist level obfuscation. In *2008 IEEE/ACM International Conference on Computer-Aided Design* (pp. 674-677). IEEE. https://doi.org/10.1109/ICCAD.2008.4681649
121. Yuan, Y., Zhang, Y., Zhao, Y., Zhang, X., & Tang, M. (2021). Process variation-resistant golden-free hardware Trojan detection through a power side channel. *Security and Communication Networks, 1*(8839222), 15. https://doi.org/10.1155/2021/8839222
122. Becker, G. T., Regazzoni, F., Paar, C., & Burleson, W. P. (2013). Stealthy dopant-level hardware trojans. In *Cryptographic Hardware and Embedded Systems-CHES 2013: 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings 15* (pp. 197-214). Springer Berlin Heidelberg https://doi.org/10.1007/978-3-642-40349-1_12
123. Sharma, R., & Ranjan, P. (2021). A review: machine learning based hardware trojan detection. In *2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)* (pp. 1-4). IEEE. https://doi.org/10.1109/IEMECON53809.2021.9689165.

124. Ghimire, A., Alkurdi, M., Amsaad, F., Rahman, M. T., & Jhanjhi, N. Z. (2024). AI-enabled Hardware Trojan Detection for Secure and Trusted Context-Aware Embedded Systems. *Authorea Preprints*. https://doi.org/10.36227/techrxiv.170630749.99115711/v1
125. Piliposyan, G., & Khursheed, S. (2023). PCB hardware Trojan run-time detection through machine learning. *IEEE Transactions on Computers, 72*, 1958–1970. https://doi.org/10.1109/TC.2022.3230877
126. Yu, S., Gu, C., Liu, W., & O'Neill, M. (2021). Deep learning-based hardware Trojan detection with block-based netlist information extraction. *IEEE Transactions on Emerging Topics in Computing, 10*(4), 1837–1853. https://doi.org/10.1109/TETC.2021.3116484
127. Sami, M. S. U. I., et al. (2024). Advancing trustworthiness in system-in-package: a novel root-of-trust hardware security module for heterogeneous integration. *IEEE Access, 12*, 48081–48107. https://doi.org/10.1109/ACCESS.2024.3375874
128. Yan, M., Wei, H., & Onabajo, M. (2021). On-chip thermal profiling to detect malicious activity: system-level concepts and design of key building blocks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29*(3), 530–543. https://doi.org/10.1109/TVLSI.2020.3047020

R. Naveenkumar is a Research scholar at the department of electronics and communication engineering at the Karunya Institute of Technology and Sciences, Tamilnadu, India. He got his M.E. degree from Sri Shakthi Institute of Engineering and Technology, India, in 2014. He has 9 years of teaching experience, 3 years of research experience, and 1 year of industry experience. He has a number of international high impact journal publications and conference proceedings papers to his credit. He has presented several research papers in international and national conferences. His research interests are in hardware security of microelectronics. He is a member of MIEEE, MISTE, MIEANG, MIRED, and MSDIWC.

Dr. N. M. Sivamangai is an Associate Professor, Department of ECE, Karunya Institute of Technology and Sciences, India. She received her Ph.D. degree from Anna University, Chennai, India in 2011. She has 13 years of teaching experience. She was instrumental in the fabrication of IC jointly with Indian Institute of Science—Bangalore, in the year 2008. Her research interests are to design and test high performance semiconductor memories and to design VLSI based systems.