PROJECT REPORT

University of Victoria

Analyzing Hardware Trojan Attacks and Countermeasures

Karthiga Thangavelu (V00925048)
Sahil Nayyar (V00944136)
Tony Massoud (v00903118)

Date: 4th August 2020

Analyzing Hardware Trojan Attacks and Countermeasures

Karthiga Thangavelu (V00925048) Sahil Nayyar (V00944136) Tony Massoud (V00903118)

***Abstract*-- Computer security does not rely just on software security, but it also depends on hardware security. Earlier, it was considered that the hardware used in the computer system is fully secure but due to the emergence of hardware Trojan attacks, it is clear that we cannot consider all the hardware components to be secure without testing them. The hardware Trojan attacks directly violate the root of trust. The hardware Trojan attacks can inject malicious code in the chip and modify how that chips works and process the information. These attacks can gain access to sensitive information and data which poses major security concerns. The other major concern with these attacks is that they can be used to cause a complete operational failure of a working machine. Information security for any company or organization is crucial and important to keep this data safe and encrypted from unauthorized access. In this paper, we are going to mention the different kinds of security threats and analyze those threats. Explaining the possible hardware Trojan attacks on the system. Also, we are going to provide some models on how to detect the attacks and, in the end, we are going to spot a light on the countermeasures.**

***Keywords*-- Hardware intellectual property (IP), Hardware, Trojan attacks, Side-channel analysis, Trojan Taxonomy, Trojan detection, Logic testing.**

I. INTRODUCTION:

Hardware Trojan is malicious medication of integrated circuits, insertion of Hardware Trojan is considered to be threat for military, governments, and banks. Hardware Trojan is inserted during the fabrication stage or design stage by fab-house or design house. The purpose of a Trojan is to get unauthorized access to sensitive information, enable a denial of service, changing the functionality of a circuit, or reducing the reliability of the circuit. To overcome this threat, various techniques are used to detect the Hardware Trojan, such as side-channel analysis, logic test, etc. there are numerous techniques available to prevent Hardware Trojan from insertion and counteract inserted Hardware Trojan.

In our project, we will focus on analyzing the Hardware Trojan threats and attacks. In addition, we will present attack scenarios, types, models, and examples. Secondly, we will discuss about an overview of Trojan detection in hardware. Lastly, we will discuss different countermeasures approaches in order to protect the hardware device.

II. THREATS

Hardware Trojan attacks have a huge impact on the security of the integrated circuits (IC). The hardware Trojan attacks can be related to a malicious modification of an IC parties. [1] This type of business model minimizes the control of IC design houses over the design and manufacturing of ICs which exposes them to malicious implants. during the design stage and fabrication and during the manufacturing phase in an untrusted environment. These attacks can also provide covert channels or a backdoor to reach sensitive information and cause a leakage of private information. An adverse situation is created when the Trojan is more advance, and it evades the conventional post-manufacturing test. This type of Trojan can be only detected when the infected device is being used for long hours of field operation. For an IC which is not very complex, the number of possible Trojans can be very high. These Trojans can vary in their activation mechanisms (triggers) and effects (payloads). The figure below shows a block diagram of a hardware Trojan, in which we have a trigger logic which when meets some specific conditions gets triggered and activates the payload logic. This payload logic when activated causes malfunctioning of the signal by modifying it from S to S'. [1][2][3]

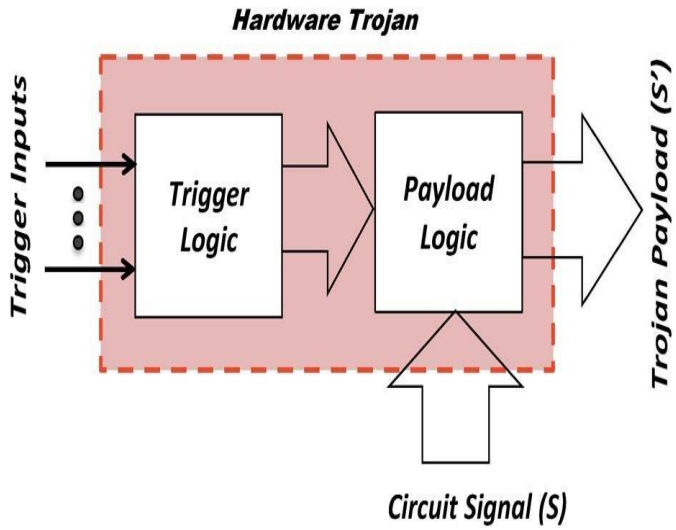


Fig. 1: General structure of hardware Trojan design [1]

Two main features of hardware Trojan are:

1- It should have malicious intent.
2- It should not be detected by conventional post-manufacturing test or validation process. [1]

One main reason for hardware Trojan attacks is the IC companies relay on untrusted third parities when designing and manufacturing their ICs. Most of the new ICs are being manufactured in unsecured fabrication factories [9]. Moreover, the suppliers of the intellectual property (IP) cores are the untrusted third-party vendors and the design and the test services are also outsourced. Even the electronic design automation (EDA) software tools are provided by different

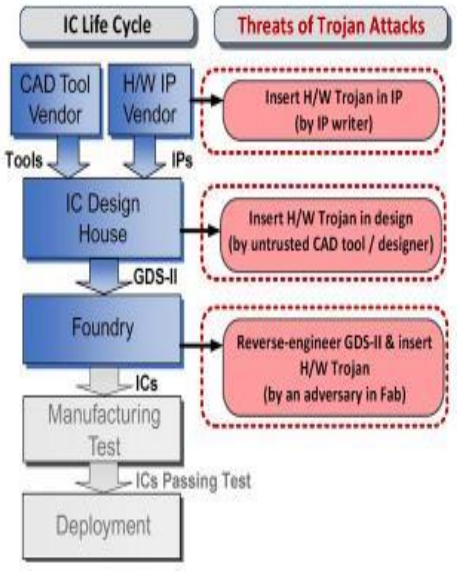


Fig. 2: Hardware Trojan attacks by different parties [1]

Fig.2 shows how Trojan attacks can be introduced in the IC during different stages of a typical IC life cycle. Every party which is involved with the design and the fabrication of an IC has the potential to tamper the IC. Tampering the IC can be achieved by adding/ deleting/ alternating the circuit structure or by modifying the manufacturing process steps that can cause reliability issues.

From an attacker's point of view, the purpose of the attack is to gain access to the chip and have access to sensitive information and have access to the secure system. Recent investigation has shown that smart opponents can mount hard to detect Trojan attacks by using few transistors or logic gates in a multimillion transistors system-on-chip (SOC) [1].

Hardware Trojan attacks have spotted the light on the new challenges for trusting the manufacturing of electronics in the field [2]. The current manufacturing process of ICs demands trust validation of ICs along with malicious design modification through different stages of the IC lifecycle. So, this makes it a necessity to have a reliable detection of any malicious design modification during post-manufacturing test. It also makes it necessary to have trust validation in hardware IP cores that are made by third-party vendors.

A. Trojan Versus Fault

Post manufacturing tests are designed to detect all types of manufacturing imperfection. Faults are logical models of physical imperfections such as stuck-at faults or a path delay faults.

In the table (1) we are comparing the difference between faults and hardware trojan. Hardware Trojans are purposely inserted by an unauthorized person in order to serve a specific malicious task, whereas faults in design occur when during the manufacturing phase, some defects are introduced. A fault is activated at a specific known functional stage in a circuit. However, we can design a hardware trojan that will activate at a random complex condition, including a series of events at the circuit nodes. [1]

B. Software versus Hardware Trojans

Table (2) discusses the properties of Software and Hardware Trojan. Software Trojan horse is a malware program that has the ability to gain access to the operating system and is able to steal sensitive information and may cause damage to the host computer by corrupting or erasing data. Hardware Trojans are inserted to the chip during the fabrication process and it can be only removed physically. However, the Software Trojan horse can be removed using an anti-trojan software. Hence, a Hardware Trojan is more difficult to remove. [9]

C. Trojan attacks through Hardware IP or CAD tools

System on Chip (SoC) based on the reusable intellectual property is now being used more often in the semiconductor industry because of the huge reduction in design/ verification cost and time. The industry is relying more on reusable pre-verified hardware IPs and a wide verity of design automation tools during SoC design. This hardware is usually collected from untrusted third-party vendors, which affects the security and the trustworthiness of the SoC platform. A major integrity concern for SoC designers is the possibility of Hardware Trojan attacks in the third-party IP.

Due to the lack of golden models, it is extremely difficult to verify the trust of an IP sourced from an untrusted third-party vendor. Regular verification methods, which depend on the availability of golden or reference design, don't work for third-party IPs. However, the functional simulation doesn't give enough coverage because of the usual incomplete functional specifications. Therefore, even if the simulation validates that design is correct with respect to functional specifications, due to a trojan it cannot provide assurance against functionality. In the same way, untrusted computer-aided design (CAD) tools can also cause malicious insertions in a design. These types of attacks can be introduced in the early design stage to skip upcoming verification steps. Usually, a designer uses a CAD tool suite and similarly a verification tool from the same vendor which can potentially bypass a malicious insertion by a synthesis engine from the same vendor [10].

	Fault	Hardware Trojan
Activation	Usually at known functional state	Arbitrary combination/sequence of internal circuit states (digital/analog)
Insertion Agent	*Accidental* (due to imperfection in manufacturing process)	*Intentional* (inserted by an adversary during IC design or fabrication)
Manifestation	Functional/parametric failure	Functional/parametric failure or information leakage

Table 1: Faults vs hardware trojan [1]

	Software Trojan	Hardware Trojan
Activation	A type of malware that resides in a code and activates during its execution	Resides in hardware (e.g. IC) and activates during its operation
Infection	Spreads through user interaction e.g. downloading and running a file from Internet	Inserted through untrusted entities in design or fabrication house
Remedy	Can be *removed in field* through S/W support	*Cannot be removed* once IC is fabricated

Table 2: Software versus Hardware Trojan [1]

III. TROJAN MODEL, INSTANCES, AND CLASSIFICATION

A. Trojan models

In order to hide the tampering with the behavior of an IC, a smart adversary would make it very difficult to detect it using the conventional post-manufacturing testing. The adversary will make sure that the tampering is not triggered under casual conditions during testing. The tampering could be triggered in long-term operation.

In figure (3), we have different forms of Hardware Trojan attack. 3(a) combinational and sequential Trojans; general model of 3(b) combinational and 3(c) sequential Trojan, which can facilitate test generation for Trojan detection; 3(d) Trojans with the capability of leaking secret information from inside a crypto chip through power side channels [4].

In figure 3(a) we can see the combinational Trojan. Combinational Trojan does not contain any flip-flops and it depends only on a set of rare conditions that occur at the same time. Figure 3(b) and 3(c) shows the combinational and sequential models respectively.

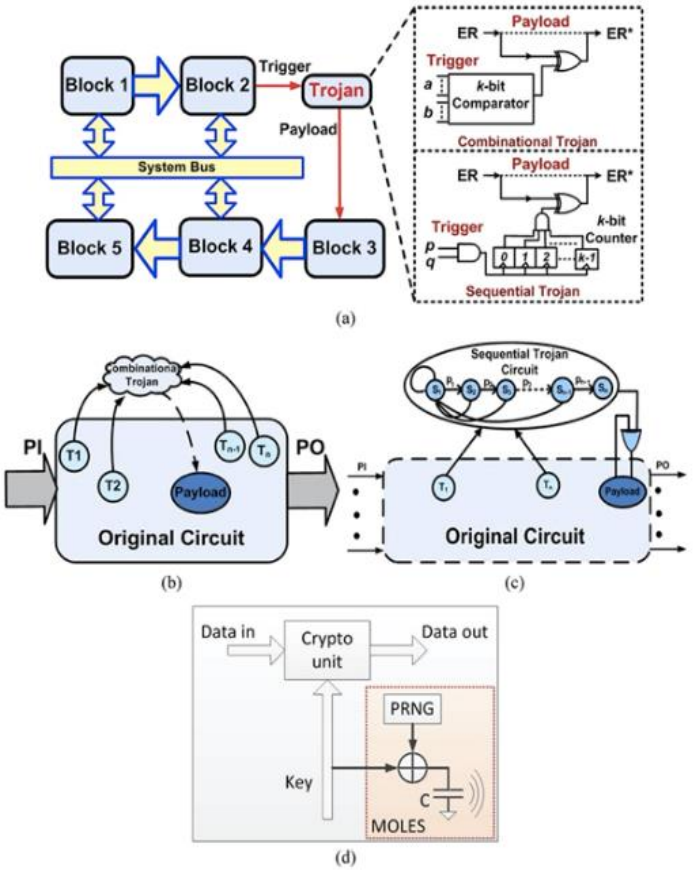


Fig. 3: Hardware Trojan attacks [11]

i. Trojan in Cryptographic engines

A Trojan attack on a cryptography engine may cause a huge break down to the security system.

A payload could create fake keys in order to cause a leakage of the secret hardware keys by the covert side channel. Figure (d) is an example of a Trojan trying to get a secret key from the cryptographic IC through power side-channels. Having a hardware modification can maximize the possibility of having leakage and extract some sensitive information. In order to access the security signal, a random number generator can be used to progress to the random session keys for a specific operation to get the passwords used in test mode to access the sensitive signal [11].

ii. Trojan in general-purpose processors

An attacker can create a backdoor in the fabrication process, which can be utilized by a software adversary. For example, the new processors enforce a hardware chain of trust in order to protect the hardware assets such as the secret keys. Using different levels of firmware and boot code authentication, to make sure that the OS is not compromised. However, the hacker and the untrusted third-party fabrication facility can insert a backdoor which disables the secure booting techniques. Buffer overflow attacks would bypass the memory range security and having access to sensitive information [12].

B. Trojan taxonomy

The taxonomy of hardware Trojan circuits can be classified in different forms, and it keeps evolving with the discovery of new attacks and Trojans.

In fig. 4 we will present a classification based on various activation mechanisms and Trojan effect.

Some researchers refer to these Trojans as reliability Trojans, that effect the reliability statement can also speed up the aging process of the devices. The reliability is reduced because of the rapid wearing out mechanisms for complementary metal-oxide-semiconductor (CMOS) transistors, for example, hot carrier injection (HCI) or negative bias temperature instability (NBTI). During the manufacturing process, selective malicious can be made, for example, the temperature being used during the nitrate layering process or changing the nitrate concentration in the gate oxide layer, which can cause the creation of infected ICs having a shorter lifetime.

The challenging part is detecting the tampering, as it can be local to a functional circuit block or even a tiny section of a block, which can possibly go unnoticed by the standard post-fabrication reliability characterization steps. In the case of the payload, upon triggering the Trojan may cause functional failure or can cause information leaking or heating of the die. In order to cause information leakage, the Trojan can transmit the data through a radio signal or serial data port interface (RS-232-C port). Also, it can include a side-channel attack that may cause information leakage through the power trace [5] or through optical modulation of an output LED or through thermal radiation [6]. A different type of Trojan payload is unauthorized changes in system behavior like DoS attack. To define and compare different Trojans, some more parameters include activation probability and hardware overhead [7]. To evade detection of the inserted Trojan, the area and power overhead should be extremely small relative to the design [8]. To reduce the overhead, the unused states and existing logic can be reused in the existing finite state machines (FSMs). Typically, the ICs layout has unused areas which can be used to insert extra gates without increasing the die footprint [1].

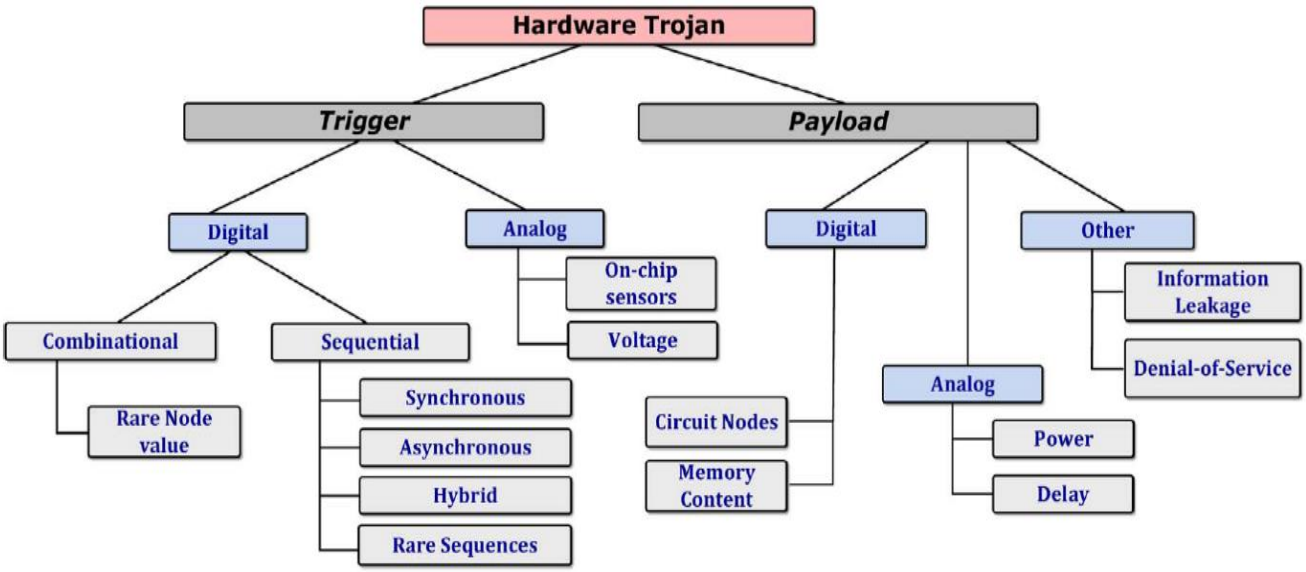


Fig. 4: Trojan taxonomy [11]

IV. HARDWARE TROJAN DETECTION

The IC Fabrication outsourcing from abroad because of its cost-effectiveness, is one of the security threats to business and military applications. Attackers can insert Trojan ICs instead of genuine ones or insert extra circuits during design or fabrication. These types of Trojans are very difficult to detect based on functional testing, bringing huge damage to system IC is related to, by leaking information using the backdoor, works as a kill switch. Such attacks trigger on certain bit patterns and input from the network or received from buses. Up-till now there's no valid and long-lasting solution that exists of Trojan detection. Developed countries have started building ICs for military purposes to avoid Trojan attacks. Other developing countries are far from this approach. Attackers try to put effort in maligning the design of IC circuits without its operation knowledge [12].

In the past many techniques have been proposed and become successful in extracting internal information for example, electromagnetic emanation, power consumption and timings of embedded systems. In side-channel attack technique works effectively even if information is masked by many noises [12].

In order to detect Trojans different tests are run on circuit to monitor its behavior on certain patterns. Presence of Trojans can be observed due to its malicious behavior. These tests are logic test based, side channel analysis (SCA). In table. 3 Pros and Cons of Logic testing and side channel analysis are discussed. Logic testing isn't useful when there is large input space to trigger. This problem can be dealt with using side channel information which might be modified in the design phase [1].

B. Logic Testing

In logic testing the exact method of Trojan detection deterministic test is not a practical solution for all Trojan detection as we have mentioned earlier rather than a statistical Trojan detection approach may be a better solution [9]. An example of this approach is multiple excitation of rare occurrence (MERO). In this methodology if a hardware Trojan is exited with rare values is increased with times of hardware Trojan trigger is satisfied on signal values [13].

Another approach in which invalid trigger conditions are eliminated using random sampling of Trojans and all input conditions are checked to see if they satisfy the trigger and Trojan at output or input flip flop scan. This approach is another way of measuring Trojan deterministically [1].

C. Side channel analysis (SCA):

In logic testing there are few drawbacks as mentioned in table 1. So, to overcome those drawbacks side channel analysis is used. In SCA side channel information is monitored if there is presence of Trojans it can be observed. There's a probability of malicious Trojan insertion during fabrication or design of circuit and such malicious circuit would have some effects on power consumptions. The delay can be caused by the circuit paths. Variation in power is likely due to circuit components addition or alteration. The continued monitoring of Trojans for their activation increases power consumption [1]. Signal-to-noise ratio (SNR) and Trojan-to-circuit are the two parameters that side channel analysis depends on. In SNR, the effect of HTs are isolated from noise because the effect of HTs on side-channel parameter such as current and delay can be easily masked by noise causing false detection. TCR is the ratio between the original and HTs affected parameter. The detection sensitivity depends on TCR when there is no noise. The side-channel analysis is based on different parameters such as transient current, static current and path delay.

i. Static Current analysis

In presence of HTs, the current is drawn form power supply in the circuit affected by HTs even without triggering Trojans. The leakage current from adding extra gates will vary from the golden chip (HT free chip) in HT affected circuit. The leakage current due to adding few extra gates in many numbers of circuits will gives poor sensitivity due to TCR. Since chip-to-chip variation in leakage current is very high which leads to low SNR. Another technique segmentation-

	Logic Testing	Side-Channel Analysis
Pros	• Robust under process noise • Effective for detecting ultra-small Trojans	• Effective for large Trojans • Easy to generate test vectors
Cons	• Difficult to generate test vectors • Large Trojan detection challenging	• Vulnerable to process noise • Ultra-small Trojan detection challenging

Table 3: Pros and Cons of Logic Testing and Side-Channel Analysis [1]

based diagnosis isolates Trojans in affected circuits. The overall leakage current of the Trojan gates can be used to identify even when they are inactive. In this method the circuit under evaluation is divided into number of segments and each segment is closely inspected for any high leakage current [13].

ii. Transient current analysis

Transient current analysis is to detect switching activity inside a Trojan circuit. In this method, the natural variation in current flow is carefully considered and its influence is minimized. Depending on the threshold, the number of false positive and false negative are balanced [14]. Another technique to find the presence of small Trojan is using IC fingerprint technique. This technique uses signal processing techniques such as Karhunen-Loeve to separate Trojan from power trace signal in the calibration of power noise. Fig. 5 shows the separated current signal from the Trojan with statistical distribution of process noise [1].

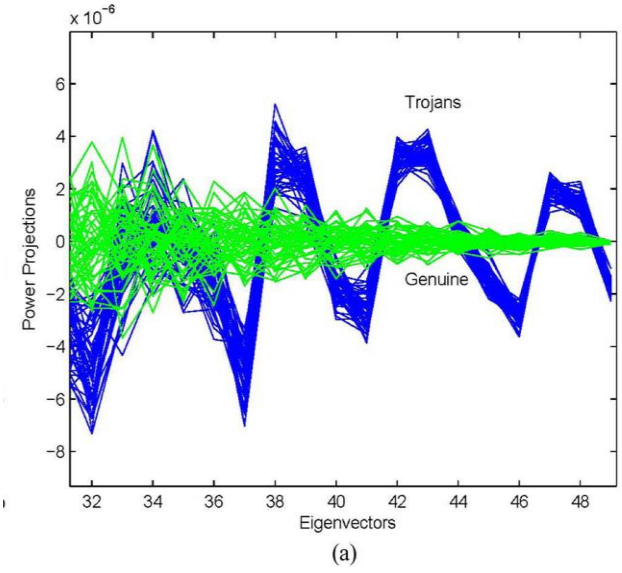


Fig. 5: Distribution of process noise using Karhunen-Loeve for separation of Trojan from current signal [1]

HTs detection sensitivity can be improved by the method called scalable side channel approaches which is based on self-referencing for ICs under large process variation. In this method, the transient current from one region is compared to another and it also nullifies the effect of process noise in terms of process variation. This side channel approach is scalable with respect to both process variation and design size. Furthermore, fig. 6 explain the comparison between HT free IC and tampered chip and fig explains the current at different points for golden IC and tampered IC. This method is also useful for tracing the source of a Trojan attack [15].

iii. Path delay analysis

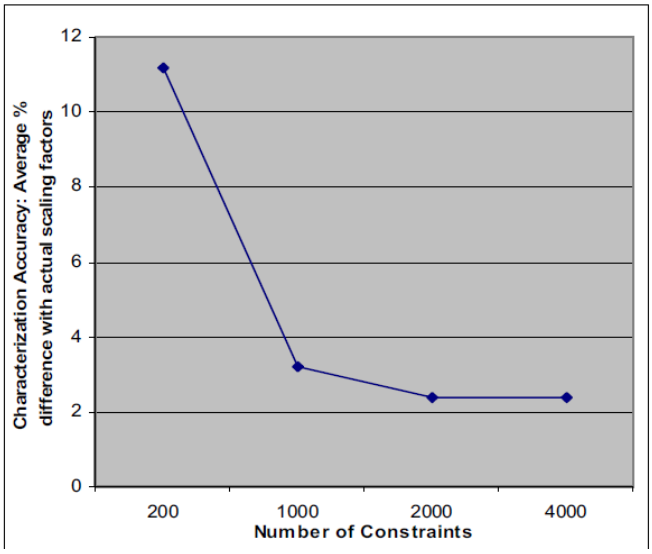
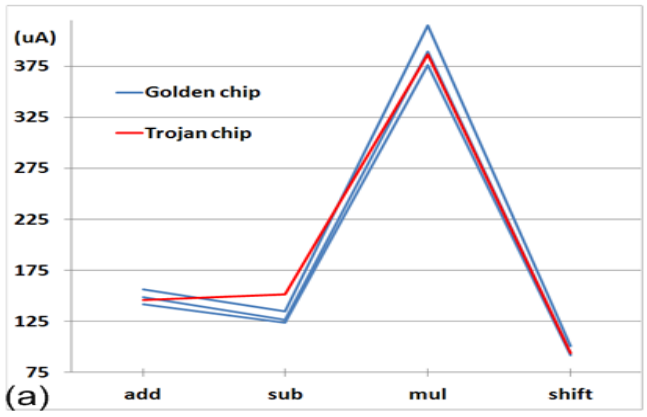
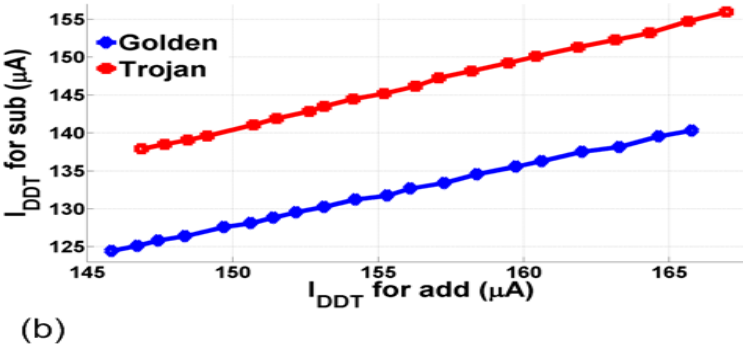


Fig. 7: Accuracy of the gate-level characterization vs number of constrain [16]

Another parameter for detecting HTs is path delay analysis. The Trojan activation in malicious ICs will cause delay in the functional path in the ICs. However, the effect of Trojan in path delay is insignificant. Small delays are masked by the process variation. Therefore, the importance should be given to choosing appropriate vector to sense the path delay by





Trojan

Fig. 6: a) comparison of supply current between golden ICs and tampered chip. b) Correlation of current at different process points for golden and tampered ICs [1]

in the affected ICs. For measuring all path delays including small path delays design-time techniques can be used. In order to isolate Trojan effect from a side-channel parameter such as delay and current, Trojan detection can be formulated as Linear Programming Problem (LPP). Single extra gates in benchmark circuits can be detected by this technique. LP (Linear Programming) is used to evaluate the consistency of characterization of gate under the assumption of added circuit. Fig. 7 shows the gate-level characterization accuracy and the different number of constrain on c432benchmarks. The number of contains is the number of equations used in LP which is the number of different inputs used to measure the leakage [16].

V. COUNTERMEASURES

The combination of state-of-the-art Hardware Trojan prevention and pre-deployment detection cannot provide a complete solution for manufactured ICs or designs free from the trojan. But still, in the presence of Hardware Trojan (HT), a useful trustworthy operation can be completed, and there are few implementations that can prevent the activation of certain Trojans. Traditional post-manufacturing test using functional and random patterns performs poorly to reliably detects Hardware Trojans. This is because during an attempt to detect defects or unacceptable variations in device parameters that cause deviation from functional or parametric specification. Due to Trojan or deviation in circuit behavior triggered by random event additional functionalities are not identified. Successful countermeasures should allow ICs to operate trustworthily even in presence of Hardware Trojans. There are several important challenges related to the reliable detection of hardware trojan using the post-silicon test/validation process [1][17].

The first challenge is that the opponents can exploit many Trojans which are varying in size and forms [18]. They vary in function and structural properties including triggering conditions and payloads, which makes it difficult to develop a logical model of Trojans. The number of possible Trojan instance has a combinatorial dependency on the number of circuit nodes [1].

Secondly, because of its stealthy nature, it is extremely challenging to activate random hardware instances and observing their effects. Therefore, deterministic and exhaustive approaches are not feasible [1]

Finally, observing Trojans in a physical parameter such as delay and supply current, process, and measurement noise are the major challenges [1].

In principle, hardware trojans themselves are vulnerable to detection by test post-manufacture or by coverage-oriented technique during the design phase. For example, the functional test developed by legitimate members of the team is cannot target RTL inserted by an opponent outside normal development processes. In a large system, trojans appear as unreachable code. Since the team may not have access to test run on the IP (Intellectual Property) supplied at RTL by the third party [19].

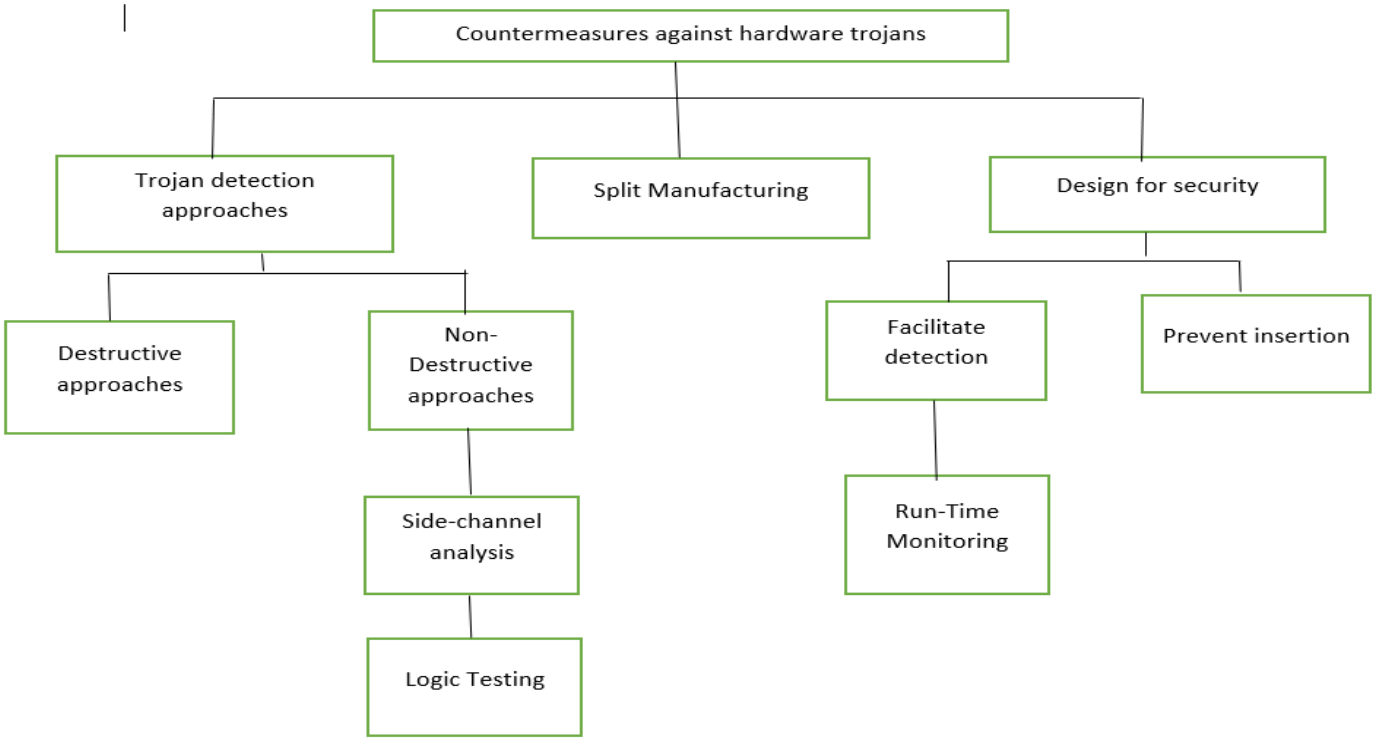


Fig. 8: Countermeasure Categories

The three board classes of solution that focuses on protecting against Hardware Trojans are:

1. Trojan detection approaches
2. Design for security approaches (DFS) and
3. Split manufacturing

From the Fig. 8, Trojan detection approach detects the trojan in IP level using the pre-silicon technique. It also uses non-destructive approaches during post-silicon manufacturing tests through a trust validation process. The destructive approaches demetallize the manufactured ICs using chemical mechanical polishing to extract layer-by-layer images using scanning electron microscopes. Non-destructive approaches further classified into test-time. Test-time approaches can be aided by DFs (Design-for-security) circuits similar to design for testability circuits like scan-chain or build-in-Self-Test-Circuitry. Time-test is further classified into Logic testing and side-channel analysis [20].

A. Detection Approaches

i. Logic testing

Logic testing focuses on generating appropriate test patterns for detecting trojans. Logic testing approaches can be combined with side-channel analysis. To carry out logic testing, the test signal is applied to the ICs under evaluation and the outcome is compared to the expected result. If the IC response doesn't match the expected result, then the IC is affected by HT. The goal of detection methods is to activate potential ICs within a reasonable time because the HT is dormant until it gets triggered based on condition. The fig. 9 explains the rare values-based HT model. In a rare values-based HT model, the triggering condition is created by the trigger input signal 0 as low or 1 as high passing through the AND gate. As the trigger input signals are controllable, the combination of trigger signals which are difficult necessarily leads to creating the condition for the payload to activate. This method is feasible in a combinational trigger where it requires combination input signals [20, 21].

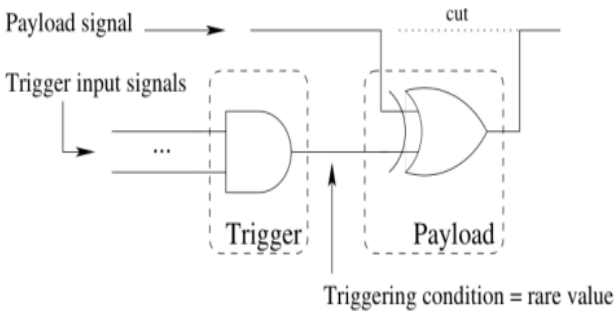


Fig. 9:" Rare Value-Based" HT model

ii. Side-Channel Analysis

Side-channel analysis focuses on monitoring physical characteristics like delay, power consumption or electromagnetic energy. The characteristic of golden ICs which is free from Hardware Trojans is compared with the characteristic of ICs under evaluation. There should be noticeable changes in characteristics of ICs if it is infected by Hardware Trojan such as electromagnetic emission, delay in paths, and changes in power consumption.

The drawbacks in Side-Channel Analysis, it requires golden ICs to compare the characteristic which is hard to obtain. The comparison ICs should be similar to HT affected ICs which can be accomplished only by reverse engineering. The process for obtaining golden ICs through reverse engineering is costly and time-consuming. However, the ICs obtained at the end of reverse engineering cannot be used and only the information for a single IC can be obtained [1].

The other drawback of this method is that due to process and environmental variations the effect of small HTs is insignificant. In order to magnify small HTs pattern generation techniques were used. Gate-level characterized methods are also used to detect HTs which magnifies the power consumption around several gates. Even though, smaller HTs evade being detected by these methods. However, golden ICs are not required for comparison since it compares the gate properties [21].

iii. Visual inspection

The visual inspection is to only observe the top level of the metal layer. This will not destroy the ICs under evaluation, unlike reverse engineering. However, detecting HTs on the low-level metal layer is impossible in visual inspection [21,22].

B. Design for Test

i. Prevent insertion

By hiding ICs functionality, it is difficult to insert HTs. To create a stealthy trigger, a good understanding of function is required. In such a case, function hiding can prevent HTs insertion. The hiding of functionality can be done by modifying the state transition graph or modifying the layout of the standard cell.

Secondly, creating a dense layout can prevent attackers to exploit available space. This can be done by replacing dummy filter cells with logic cells. These logic cells can be tested; hence the replacement of those cells is noticed during test time [21].

ii. Facilitate Detection

Detection facilitation is done by facilitating the detection methods such as by providing a characteristic of HTs free ICs to compare with HTs affected ICs for side-channel analysis. The obfuscation method facilitates power and path delay in the side-channel analysis. In path delay analysis, the shorter paths are included in the nets of a longer path. Hence, the shorter path has less PV (Process Variation) than a longer path which makes it easier to detect HTs inserted in the shorter part. However, this is an assumption using delay analysis which might be true from a mathematical point of view, not from a measurement point of view [23]. In order to facilitate logic testing, is to design a circuit in such a way it is more difficult for the attacker to create stealthy trigger condition so that the potential trigger can be detected during the time of testing [21].

1. Run-time monitoring

Run-time monitoring comes under facilitating detection. Concurrent Error Detection (CED) technique is the run-time monitoring method for preventing HTs. This method makes the insertion of HTs difficult and possible to detect it during run-time. CED is based on Modified Dual Modular Redundancy (MODMOR) which introduces redundancy through duplication of the circuit along with checker. Fig. 10 explains if there are two circuits one is original (c1) and the other is duplicate of c1. Inserting HT in one circuit and activating it during run-time, then the output of two circuits will not be same. There is a possibility that the attacker can insert the HTs in both circuits if they build in the same way. By providing two different layers for the original and duplicate circuit, it is impossible for the attacker to identify from the inspection of the layout of the circuit for HTs insertion which is shown in Fig. 11[24].

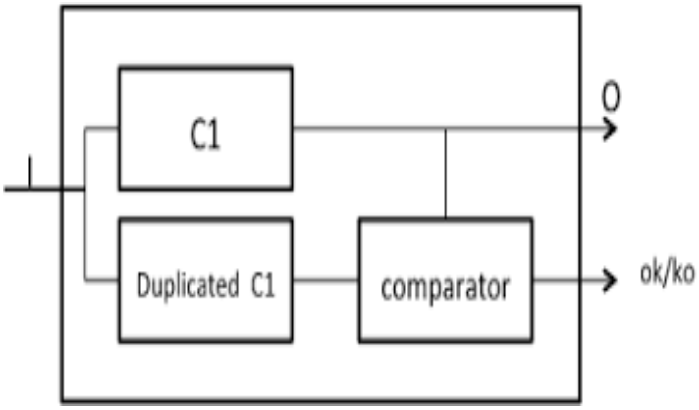


Fig. 10: Original circuit C1 and its' duplicate

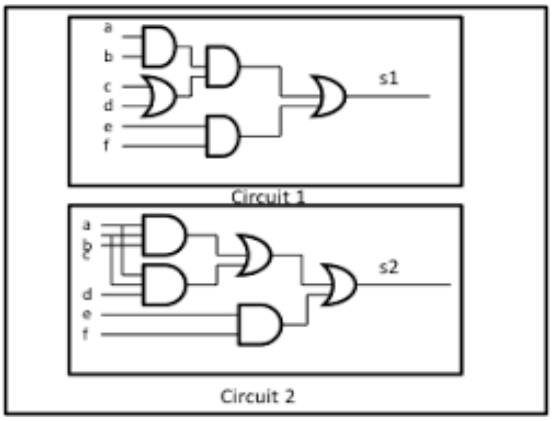


Fig. 11: Original netlist and its duplicate version

C. Split manufacturing

Split manufacturing evolved to protect ICs fabricated in untrusted factories. In this, the front-end layer, transistor, and lower metal layer are fabricated in unreliable foundry and the backend layers and higher metal layer are fabricated in the reliable foundry. Using two foundries for manufacturing makes it costly. Moreover, it is challenging for backend layer manufacturing foundry as it has to manufacture a layer for already available frontend layer. However, split manufacturing doesn't provide 100% security [25].

VI. CONCLUSION

We analyzed and studied various detection methods and countermeasures for Hardware Trojans and finally presented the most effective methods. In addition, we reviewed how a Trojan attack works and how it is inserted in a circuit/device. Also, we discussed Hardware Trojan threats and we spotted a light on the difference between Hardware and software Trojans. Moreover, we showed some models and attack scenarios. Also, we analyzed and reviewed various detection methods and countermeasures to prevent Hardware Trojan insertion. We focused on detection approaches which reveal that small Hardware Trojan can evade side-channel analysis and logic testing. However, it is not possible to ensure the detection of every Hardware Trojan.

REFERENCES

[1] Swarup Bhunia, “Hardware Trojan Attacks: Threat Analysis and Countermeasures”, 15 July 2014

[2] S. Bhunia et al., ‘‘Protection against hardware Trojan attacks: Towards a comprehensive solution,’’ IEEE Design Test Comput., vol. 30, no. 3, pp. 6–17, May–Jun. 2013

[3] S. Adee, ‘‘The hunt for the kill switch,’’ IEEE Spectrum, vol. 45, no. 5, pp. 34–39, May 2008

[4] L. Lin, W. Burleson, and C. Paar, ‘‘MOLES: Malicious off-chip leakage enabled by side-channels,’’ in Proc. Int. Conf. Comput. -Aided Design, 2009, pp. 117–122

[5] L. Lin, W. Burleson, and C. Paar, ‘‘MOLES: Malicious off-chip leakage enabled by side-channels,’’ in Proc. Int. Conf. Comput. -Aided Design, 2009, pp. 117–122.

[6] F. Kiamilev and R. Hoover, ‘‘Demonstration of hardware Trojans,’’ presented at the DEFCON 16, Las Vegas, NV, USA, Aug. 8–10, 2008

[7] M. Tehranipoor and F. Koushanfar, ‘‘A survey of hardware Trojan taxonomy and detection,’’ IEEE Design Test Comput., vol. 27, no. 1, pp. 10–25, Jan.–Feb. 2010.

[8] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, ‘‘Trustworthy hardware: Identifying and classifying hardware Trojans,’’ IEEE Computer, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[9] G. McGraw and G. Morrisett, ‘‘Attacking malicious code: A report to the Infosec Research Council,’’ IEEE Software, vol. 17, no. 5, pp. 33–41, Sep.–Oct. 2000

[10] M. Potkonjak, ‘‘Synthesis of trustable ICs using untrusted CAD tools,’’ in Proc. Design Autom. Conf., 2010, pp. 633–634

[11] L. Lin, W. Burleson, and C. Paar, ‘‘MOLES: Malicious off-chip leakage enabled by side-channels,’’ in Proc. Int. Conf

[12] Agrawal, Dakshi, et al. "Trojan detection using IC fingerprinting." *2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE, 2007.

[13] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In International Conference on Cryptographic Hardware and Embedded Systems (CHES’09), pp. 396–410, 2009.

[14] S. Narasimhan et al., ‘‘Hardware Trojan detection by multiple-parameter side-channel analysis,’’ IEEE Trans. Comput., vol. 62, no. 11, pp. 2183–2195, Nov. 2013.

[15] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, ‘‘Trojan detection using IC fingerprinting,’’ in Proc. IEEE Symp. Security Privacy, 2007, pp. 296–310. (ori)

[15] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia, ‘‘Self-referencing: A scalable side-channel approach for hardware Trojan detection,’’ in Proc. 12th Int. Conf. Cryptogr. Hardware Embedded Syst. Workshop, 2010, pp. 173–187

[16] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, ‘‘Hardware Trojan horse detection using gate-level characterization,’’ in Proc. Design Autom. Conf., 2009, pp. 688–693.

[17] S. Narasimhan et al., "Hardware Trojan detection by multiple-parameter side-channel analysis", *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2183-2195, Nov. 2013.

[18] Mark Beaumont, Bradley Hopkins and Tristan Newby,” Hardware Trojans – Prevention, Detection, Countermeasures (A Literature Review)”, 2011

[19] Hardware trojan attack and countermeasures, https://www.techdesignforums.com/practice/guides/hardware-trojan-security-countermeasures/.

[20] Mohammad Tehranipoor, Cliff Wang, “Introduction to Hardware Security and Trust”, Springer, LLC 2012, pp. 339 - 344.

[21] Sophie Dupuis, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre. Protection against Hardware Trojans with Logic Testing: Proposed Solutions and Challenges Ahead. IEEE Design & Test, IEEE, 2018, 35 (2), pp.73-90.

[22] S. Bhasin, J.L. Danger, X.T. Ngo and S. Guilley, “Hardware trojans horses in cryptographic IP cores”, In Fault Diagnostic and Tolerance in Cryptography (FDTC’13), pp. 15–29, 2013

[23] Nejat, Arash et al. “Facilitating side channel analysis by obfuscation for Hardware Trojan detection.” *2015 10th International Design & Test Symposium (IDT)* (2015): 129-134.

[24] Manikandan Palanichamy, Papa-Sidy Ba, Sophie Dupuis, Marie-Lise Flottes, Giorgio Di Natale, et al. Duplication-based Concurrent Detection of Hardware Trojans in Integrated Circuits. *TRUDEVICE*, Nov 2016, Barcelona, Spain. ⟨lirmm-01385551⟩

[25] K. Xiao, D. Forte and M. M. Tehranipoor, "Efficient and secure split manufacturing via obfuscated built-in self-authentication," *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, Washington, DC, 2015, pp. 14-19, doi: 10.1109/HST.2015.7140229.