MINI PROJECT

Simulating Hardware Trojan Infected Circuit

December, 2019

Department of Information Technology

Indian Institute of Engineering Science and Technology, Shibpur

Supervisor

Surajit Kumar Roy

Associate Professor

Information Technology

Indian Institute of Engineering Science and Technology, Shibpur

Group Members

Sudipti Mandal (510817009)

Shivam Bhaskar (510817027)

Kinshuk Hazra (510817056)

Contents Page No.

Introduction

Information and Communication Technology has become an integral and very essential part of our civilisation. It has a wide application in banking, industries, healthcare and in defence and military technologies also. So, cyber-security has

become a great concern for the government and the researchers as well. In this context, we learn about Hardware Trojans, which are structural attacks on the integrated circuits (ICs) which disrupts their normal operation, functionality and can even breach information and system security.

So, hardware Trojans are a big threat to cyber-security as it is almost impossible to detect it with any practical detection scheme. As they can attack in various ways, the researchers have worked hard to find various methods to detect Hardware Trojan.

So, in this project we discuss about the Trojans, different published methods of finding Trojan, what method we adopted to detect Trojan and what conclusion we drew from it. In this project, we performed the logic testing to detect Trojan.

What is Hardware Trojan?

Hardware Trojan (**HT**) is a malicious modification of the circuitry of an integrated circuit. A hardware Trojan is completely characterized by its physical representation and its behavior. The payload of an HT is the entire activity that the Trojan executes when it is triggered.

In general, malicious Trojans try to bypass or disable the security fence of a system. It can leak confidential information by radio emission. HT's also could disable, derange or destroy the entire chip or components of it.

Hardware Trojan reside at the lowest level of information processing -on the integrated circuit (IC) board. They can cause incorrect functioning of a component. Hardware Trojans are an increasing threat to every processing environment, particularly for commercial applications, as well as to critical infrastructure like military technologies. The possibility for hardware Trojans to be inserted into hardware has been a growing concern. Integrated circuits can be infected with a Hardware Trojan either during manufacture or post-manufacture tampering. With the Outsourcing services and globalization of electronic component manufacture It is very difficult or impossible to ensure hardware, safety and the risk of hardware Trojans is increased rather than when the all phases of production of the product is done in same manufacture or at least in the same country.

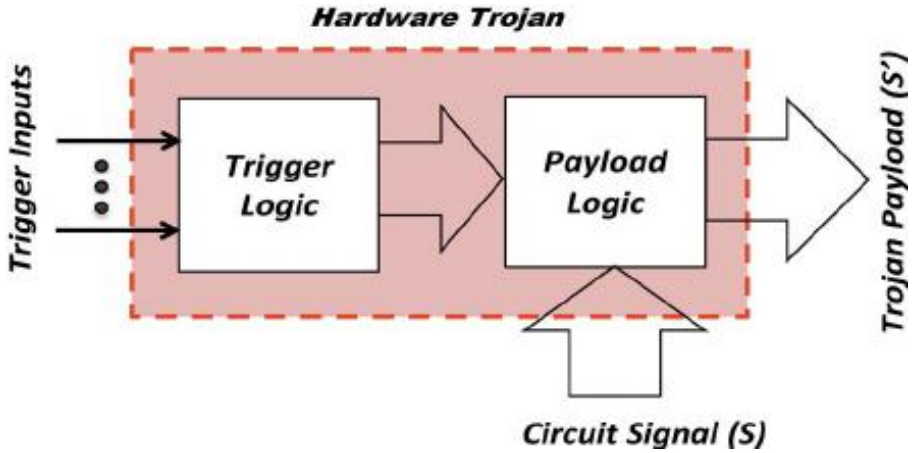


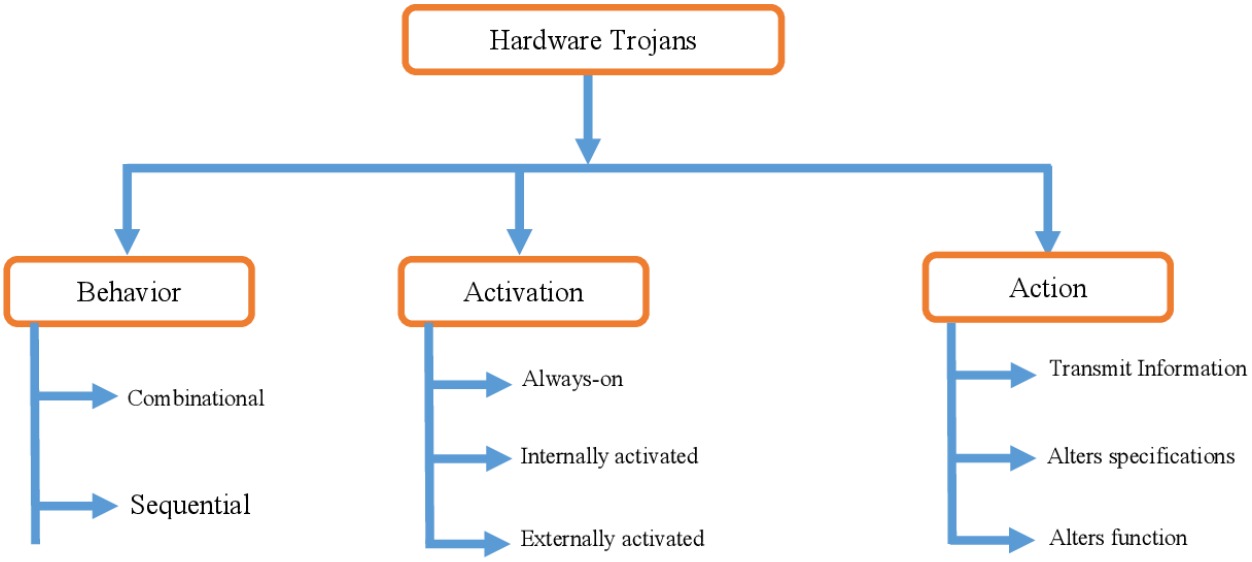
Figure 1: A Simple Hardware Trojan

Effects of Trojan

- Hardware Trojan horses can affect circuits during normal and routine activities or in the idle time and cause failure in the security mechanisms of the system.
- Hardware Trojan can cause hardware damage and adversely affect the system's normal operation .
- The effects of Trojans on target hardware or systems can range from subtle disturbances to catastrophic system failures.
- These attacks can acquire critical information of the system during executing, storing and transferring of information and send it to the specified destination.
- A Trojan can cause an error detection module to accept inputs that should be rejected.
- A Trojan can downgrade performance by intentionally changing device parameters, such as power and delay.
- A Trojan might leak a cryptographic algorithms secret key through unused RS-232 ports.
- Denial-of-service Trojans prevent operation of a function or resource. For example causing the processor to ignore the interrupt from a specific peripheral.

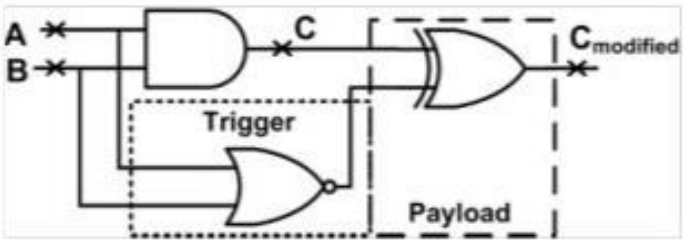
Different Types of Hardware Trojan

Trojans are classified based on behavior, activation and action. Each of them is discussed below in details.



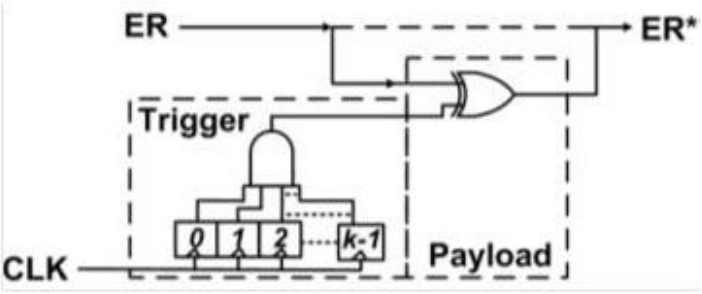
Behavior: Malicious circuit design inserted can be subdivided based on the type of design implementation. It can either be combinational design or sequential designs.

Combinational: The activation of combinationally triggered Trojan depends on the occurrence of a particular condition at certain internal nodes of the circuit. Fig. (a) shows an example of a combinationally triggered Trojan where the occurrence of the condition A= 0, B = 0 at the trigger nodes A and B causes a payload node C to have an incorrect value at C(modified). Typically, an adversary would choose an extremely rare activation condition so that it is very unlikely for the Trojan to trigger during conventional manufacturing test.

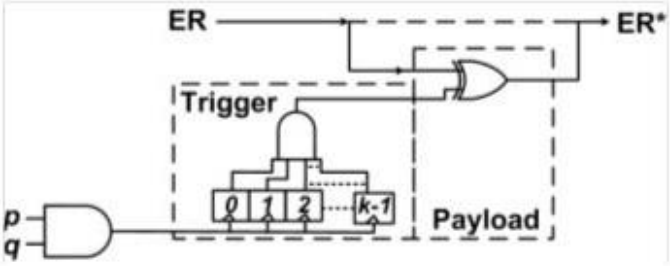


(a) Combinationally triggered Trojan

Sequential: The activation of sequencial triggered Trojan depends on the occurrence of a specific sequence of rare logic values at internal nodes.
So they are activated by the occurrence of a sequence, or a period of continuous operation. n. The simplest sequential Trojans are synchronous stand-alone counters, which trigger a malfunction on reaching a particular count. Fig. (b) shows a synchronous k-bit counter which activates when the count reaches 2 k−1, by modifying the node ER to an incorrect value at node ER. An asynchronous version is shown in Fig. (c), where the count is increased not by the clock, but by a rising transition at the output of an AND gate with inputs p and q.



(b) Synchronous counter ("time-bomb") Trojan



(c) Asynchronous counter Trojan

Note that more complex state machines of different types and sizes can be used to generate the trigger condition based on a sequence of rare events. In general, it is more challenging to detect sequential Trojans using conventional test generation and application, because it requires satisfying a sequence of rare conditions at internal

circuit nodes to activate them. The number of such sequential trigger conditions for arbitrary Trojan instances can be unmanageably large for a deterministic logic testing approach.

Activation:

Combinational and sequential designs are only activated with some special patterns. Based upon method of activation, Trojans are classified as Always-On, Internally activated and externally activated.

Always On: This class covers Trojans that are implemented by modifying the geometries of the chip such that certain nodes or paths in the chip have a higher susceptibility to failure.

Internally Triggered: An event that occurs within the target device activates an internally triggered Trojan. Internally activated Trojans are not activated till specific condition is met .i.e. Trojans can be activated only with special patterns. Activated Trojan modify the functionality of designs.

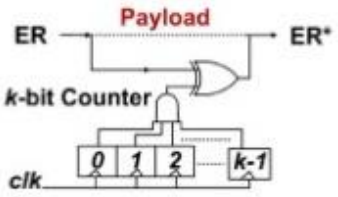


Figure : Internally Triggered

Externally Triggered: In this case, Trojan requires external input to the target module to activate. For example, data coming through external interfaces such as RS-232(a type of data transmission connector) can trigger a Trojan.

Action:

The Trojans are also classified on the basis of the actions that they perform:-

Alter Specifications – they may alter some internal signals which in turn can alter some specifications of an operation.
Alter Functions – they can alter functionalities of a system
Transit information – they can acquire information of the system and transmit it through radio signals or through covert channels created in the altered circuit.

Examples of Hardware Trojan Usage

As we learned from the section “Effects of Trojan”, the Trojans can adversely affect the system, thus stopping its normal operation and can even cause failure in the security mechanisms of the system. And in some cases, they can even acquire critical informations of the system and send it to a specified destination.

Trojans can be employed by cyber-thieves and hackers trying to gain access to users' systems. Users are typically tricked by some form of social engineering into loading and executing Trojans on their systems. Once activated, Trojans can enable cyber-criminals to spy on you, steal your sensitive data, and gain backdoor access to your system. These actions can include:

- Deleting data
- Blocking data
- Modifying data
- Copying data
- Disrupting the performance of computers or computer networks

Trojans can also be used by adverseries of a country or an organization for stealing vital information, stopping them from doing a major operation by causing some problems in the system deliberately.
A recently reported Trojan attack involves US Navy, who discovered a hardware

"backdoor" in a microchip used in Different industries. For example the chips could have been hacked, able to shut off a missile in the event of war or just lie around waiting to malfunction.
In another case Chinese Information Technology firms have long attracted suspicion from international governments, with telecommunications firms recently coming under suspicion in both the US and UK. As nowadays, most of the computer hardware manufacturing hubs are set in China, even though, this is just a suspicion.

Hardware Trojan Detection Methods

Attack detection is the first and perhaps most important step in any security system. So the attack detection is the most important action to counter the Hardware Trojan. It is not possible to completely prevent the insertion of a Hardware Trojan into the system during the design phase. Where preventative measures are used to protect against Hardware Trojans being inserted into an IC, detection techniques are used to discover the presence of a Hardware Trojan.
Detecting a hardware trojan requires overcoming numerous challenges. Namely:
1. Handling large architectures.
2. Being non-destructive to the IC.
3. Being cost effective.
4. Ability to Detect trojans of all sizes.
5. Authenticating chips in as small a time frame as possible.
6. Dealing with variations in manufacturing processes.
7. Detecting all trojan classifications.
8. Detecting trojans in a reasonable time frame.

There is no single method capable of detecting all types of hardware trojans, nor overcoming all the challenges described here-above. Over the years, several methods have been developed to detect different types of trojans. These methods are described here-after.

Hardware Trojan detection methods can be divided into different categories as shown in the figure:

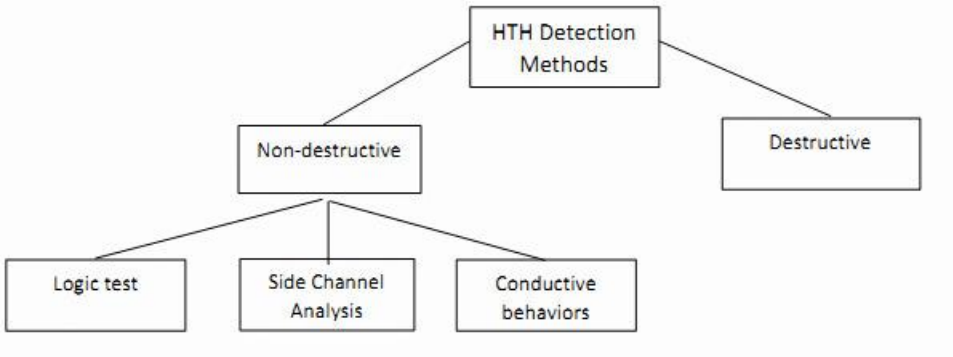


Fig.

Destructive Methods

The Destructive methods for Hardware Trojan detection completely destroy the IC that they examine, lessening the usefulness of such techniques. In this method that is the first and easiest method to detect the Hardware Trojan the first protective layer of the circuit are going to be opened and then all of its components separated by reverse-engineering techniques and are checked by specific physical devices or chemical materials. But it should be considered that reverse engineering a complex modern IC is a time consuming and expensive process. In Destructive methods, scanning optical microscopy (SOM) and light induced voltage alteration (LIVA) techniques are used for reverse engineering.

Destructive methods are more Costly and time consuming. Because Hardware Trojan can be inserted into the circuit by remove or modify a few logic gates. the new circuits including large-scale integration (LSI) and Very-large-scale integration (VLSI) circuits may be contain billions of logic gate and if destructive methods are used to detect hardware Trojans all of these gates should be checked .

Destructive methods have many Problems beside its benefits. A Hardware Trojan may be infecting the IC by the insertion, deletion or modification of as few as two logic gates whereas modern ICs may consist of billions of gates. Finding this “needle in a haystack” requires complete reverse – engineering at the gate level of the IC. In addition, there is no guarantee that IC that have a Hardware Trojan will generate a different fingerprint.

Non-Destructive Methods

Difficulties and high costs of destructive methods caused the introduction of nondestructive methods. Non - destructive methods for Hardware Trojan detection do not destroy the IC being tested, and are classified as being either invasive, or non - invasive.

Non - invasive techniques leave the design unaltered, whereas the invasive techniques modify the design in order to embed features to assist with Trojan detection. These methods can be divided into several categories that in the following are reviewed.

Logic Testing: Logic testing includes equivalence checking at the presilicon design phase and generation of specific test patterns through Automatic Test Pattern Generation (ATPG) to excite critical paths during chip testing. These methods are based on the analysis of the IC’s logic structures and divided into functional behavior analysis and finding hidden features and methods. In functional behavior analysis method,

researchers insert some test vector into the inputs of electronic circuit and analyze the outputs. If the output is incompatible with the input, an anomaly is recognized. In fact, this method generally is used for the detection of functional errors and beside it can detect parametric Hardware Trojan (adding hardware Trojans by modify in the structure of the circuit) and cannot detect functional hardware Trojans (adding hardware Trojans by Add / subtract some elements in the circuit). The most important problem with logic test's functional behavior analysis methods is large scale of the test environment in ICs. It even makes the entire test almost is impossible in large ICs. To overcome this limitation, some methods have been presented. Researchers proposed a method Based on randomization. In this method, different patterns implement in the input of circuit and then a probabilistic fingerprint has been formed for circuit by the outputs of the circuit. Then the same pattern implemented to examined circuit and compare the output result with the probabilistic finger print. If there are differences, it is assumed that circuit infected by a Hardware Trojan.

Side Channel Analysis: Side channel analysis techniques are some of the most commonly used procedures in hardware trojan detection. These techniques generally measure signals such as power and path delay, looking for fluctuations potentially caused by trojans. In some methods electromagnetic wave propagation and dynamic current values are also used as a parameter to detect Trojan. Side channel analysis can have a high success rate as even in a dormant state the trojans trigger signal will cause some current leakage. Power based side channel methods use the power parameter of the circuit for Hardware Trojan detection. To get result in the Hardware Trojan detection by power analyze method there should be a method for analyzing the feedbacks that are received from the circuit.

In the path delay based Hardware Trojan detection methods according to other side channel analyze methods and by replacing the delay factor instead of other parameters trying to detect Hardware Trojan.

There are some other methods also for detecting hardware Trojans which are explained as follows:

Physical Inspection

One of the most obvious methods of detection is physical inspection of the board itself. This method is sometimes classified as a failure analysis based technique. Those techniques usually comprise two steps: (1) cutting and lifting the molding coat to expose the circuitry; and (2) performing various scans.

Built-in-Self-Techniques

Built-In-Self-Test (BIST) techniques are commonly used to detect manufacturing faults and are present in many chips. If unknown or malicious logic is detected during these tests a bad checksum result is given, although designed to detect manufacturing faults on some occasions these tests can detect hardware Trojans.

LOGIC TESTING FOR TROJAN DETECTION

Out of all the methods we discussed before, we have performed Logic Testing for the detection of Trojan.

In Logic Testing, the tester inputs some test vector into the circuit to be tested and analyze the output. So, the tester has to keep a record of the output of the circuit for that input set when it is Trojan free that is, in its original state. Now, while testing, if the output is incompatible with the original output then an anomaly is recognized which implies that Trojan is PRESENT in the circuit.

We have used Xilinx ISE Design Suite for simulating the circuits and we wrote the codes in Verilog.

In some cases the, Trojan may not be detectable by testing some input vectors.

This is because, they might be present at some places where it is difficult to detect, that is, it is triggered very rarely. Researchers have found those regions to have low signal transition probability.

Anyways, we have considered the following approaches for those cases.

Approaches we have considered for logic testing :

1. **Random Input Sets:** Random sets of inputs can be used to trigger Trojan . This method is statistically fair for triggering Trojan and also it is applicable in large circuits Though main drawback of this method is that there can be nodes which are rarely activated and hence their probability to be activated is significantly decreased .

2. **Exhaustive set of inputs:**

   As we know that the Trojan may not be detectable if we use any random set of inputs, because firstly, a circuit may have a large number of input sets. Secondly, the Trojan may be present in a region which is rarely activated.

So, in this case if we consider all possible set of inputs, that is, exhaustive set for that particular circuit, there is a chance that at least in one case, we might get an anomaly in the output.

But it is difficult to find the exhaustive input combination of a circuit as a circuit with n inputs would have 2^n possible sets.
So, we made an algorithm to generate exhaustive input sets for a circuit.
Since performing the circuit simulation in Verilog, the input sets in verilog test fixture looks like this:

```
INPUT1=0;

INPUT2=1;

#100;   //this is the time delay
```

So, we had to generate each input set like this. The algorithm for generating this is as follows:

1.Enter the number of inputs (say n)

↓

2. Enter each input one by one in an array

↓

3. create a list of all possible combinations of choosing one or more inputs from the array.(length of the list will be 2^n= number of all possible input sets)

↓

4. iterate through each combination of the list.(say j)

a. iterate through the input array

if input is in j:

value of that input is set 1

else

value of that input is set 0

(in each iteration one input set is created)

↓

5.Stop

We wrote the code in Python language. It is given as follows:

```
from itertools import combinations

a=[]

n=int(input('Enter no. of inputs: '))

print('Enter the inputs:')

for i in range(n):

   a.append(input())

time=int(input('Enter time:'))

for i in (n+1):

   comb=combinations(a,i)

   for j in comb:

      l=list(j)

      for k in a:

         if k in l:

            print(k+'=1;')

         else:

            print(k+'=0;')

      print('#{0};'.format(time))
```

The main problem in this method is excessively large set of inputs . Even in case of a small circuit with 8 inputs , the possible input combinations are 2^8 . So, practically it's nearly impossible to check all input combinations .

The program might crash if the number of inputs is large.

3.**Dependence on Signal Transition Probability**

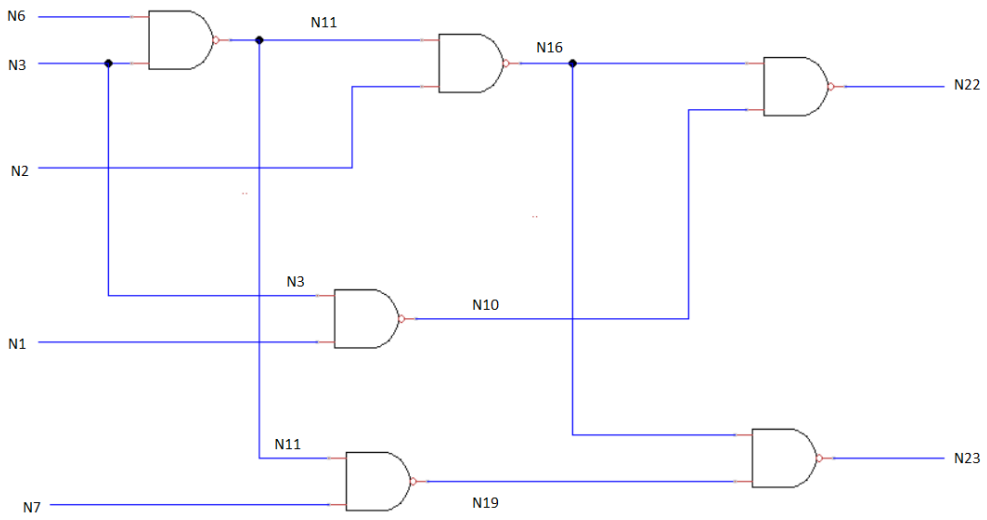
Signal Transition Probability(STP): It is defined as the probability of transition of the signal from one state to other, that is from 0→1 or from 1→0.

Many researchers have experimentally found that detection of Trojan loosely depends on signal transition probability. It is found that if a Trojan lies at a region(wire) with low transition probability, then it is very difficult to detect it. In some cases, it is even undetectable. So, the attackers generally target those wires having less transition probability.

So, the tester has to identify the circuit space which is prone to Trojan insertion, and test them with large sets of input to find anomaly in the output. Those input sets has to be considered which can trigger/activate those regions.

Experiment1:

In this experiment, we have considered the following circuit for Trojan detection:



Here, Inputs:N6, N3, N2, N1 and N7

Wires: N10, N11, N16, N19

Output: N22,N23

Output of circuit

INPUT					OUTPUT	
N1	N2	N3	N6	N7	N22	N23
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	1	1
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	1
1	1	0	0	0	1	1
1	0	1	0	0	1	0
1	0	0	1	0	0	0
1	0	0	0	1	0	1
0	1	1	0	0	1	1
0	1	0	1	0	1	1
0	1	0	0	1	1	1
0	0	1	1	0	0	0
0	0	1	0	1	0	1
0	0	0	1	1	0	1
1	1	1	0	0	1	1
1	1	0	1	0	1	1
1	1	0	0	1	1	1
1	0	1	1	0	1	0
1	0	1	0	1	1	1
1	0	0	1	1	0	1
0	1	1	1	0	0	0
0	1	1	0	1	1	1
0	1	0	1	1	1	1
0	0	1	1	1	0	0
1	1	1	1	0	1	0
1	1	1	0	1	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	0
0	1	1	1	1	0	0
1	1	1	1	1	1	0

So, we placed a Trojan circuit at wire N16, by taking triggers from N2, N1, N7 and N11 with transition probabilities 0.5, 0.5, 0.5 and 0.1875 respectively. These S.T.Ps can be considered to be high.
We tested the circuit with all the input sets and observed the following:

Output with Trojan at N16

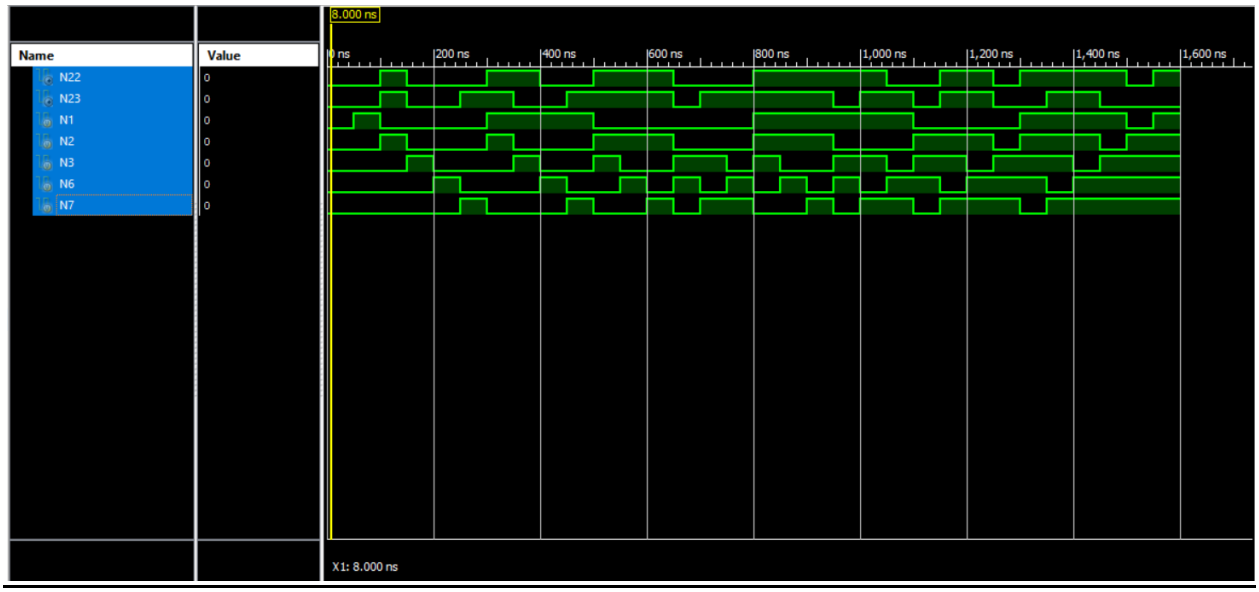
INPUT					OUTPUT	
N1	N2	N3	N6	N7	N22	N23
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	1	1
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	1
1	1	0	0	0	1	1
1	0	1	0	0	1	0
1	0	0	1	0	0	0
1	0	0	0	1	0	1
0	1	1	0	0	1	1
0	1	0	1	0	1	1
0	1	0	0	1	0	1
0	0	1	1	0	0	0
0	0	1	0	1	0	1
0	0	0	1	1	0	1
1	1	1	0	0	1	1
1	1	0	1	0	1	1
1	1	0	0	1	1	1
1	0	1	1	0	1	0
1	0	1	0	1	1	1
1	0	0	1	1	0	1
0	1	1	1	0	0	0
0	1	1	0	1	0	1
0	1	0	1	1	0	1
0	0	1	1	1	0	0

1	1	1	1	0	1	0
1	1	1	0	1	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	0
0	1	1	1	1	0	0
1	1	1	1	1	1	0

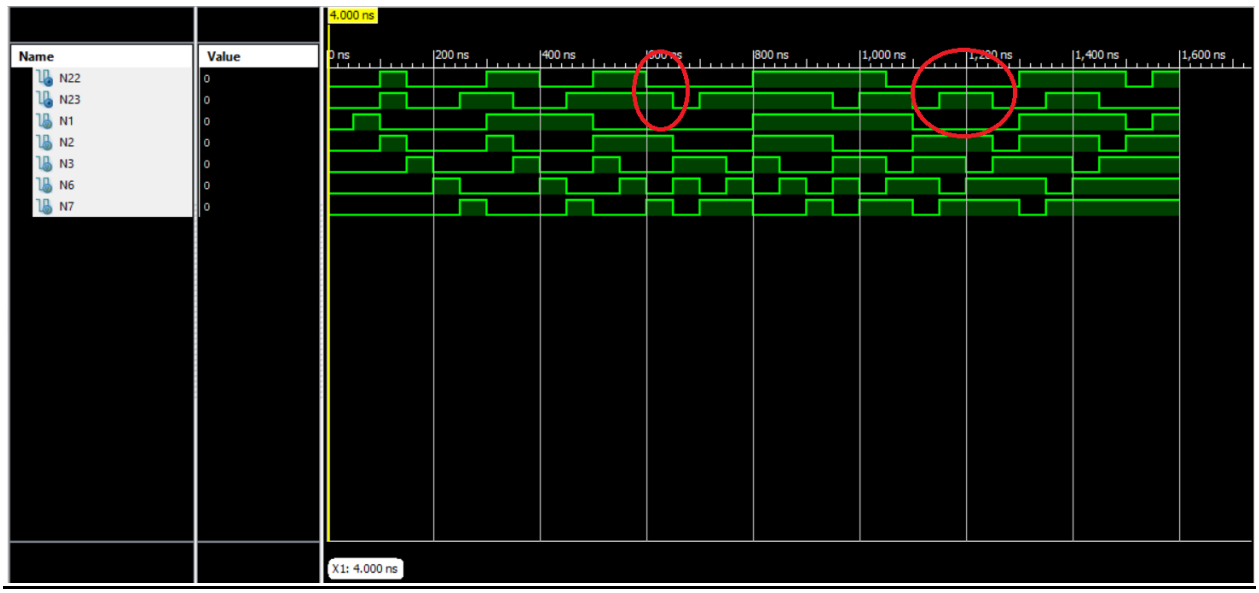
Result Thus, we found anomalies at three input sets(marked in red) which implies that the Trojan is detected.

The anomaly is clearly visible in the behavioral of the two circuits:

Normal Behavioural



Behavioural when Trojan at N16



Experiment2:

In this experiment, we have used the same circuit but, we placed the Trojan at N19 taken triggers as N11,N10,N11,N16 with S.T.Ps 0.1875,0.1875,0.1875 and 0.085. These S.T.Ps are comparatively lower than the previous case.

We observed the following:

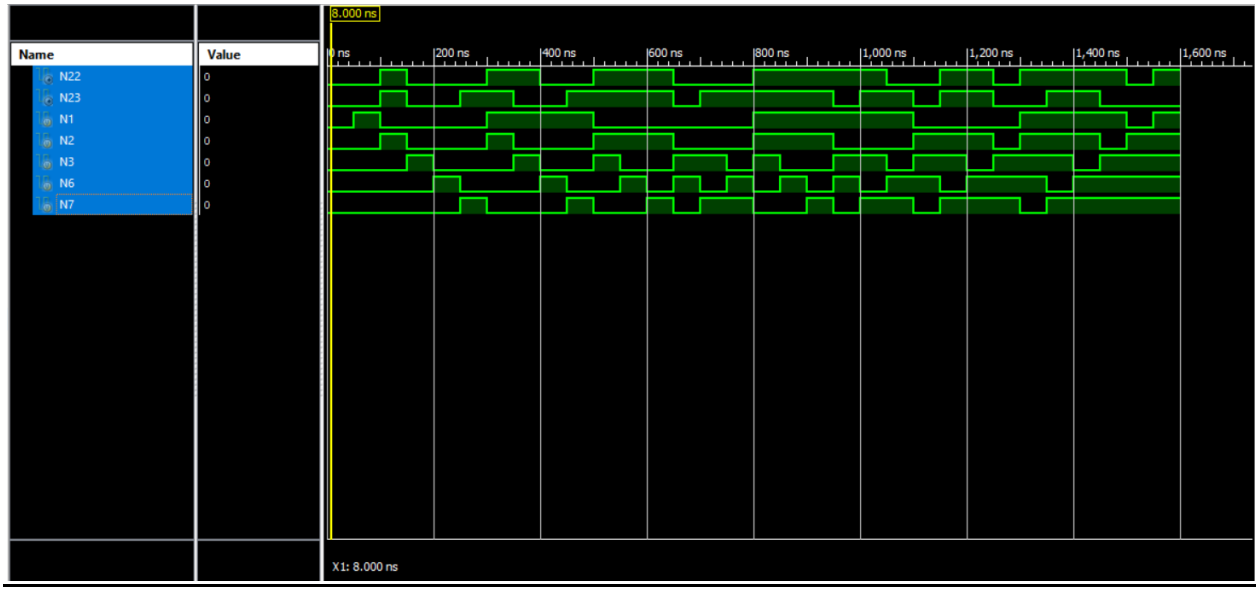
Output of C17 circuit

INPUT					OUTPUT	
N1	N2	N3	N6	N7	N22	N23
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	1	1
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	1
1	1	0	0	0	1	1
1	0	1	0	0	1	0
1	0	0	1	0	0	0
1	0	0	0	1	0	1
0	1	1	0	0	1	1
0	1	0	1	0	1	1
0	1	0	0	1	1	1
0	0	1	1	0	0	0
0	0	1	0	1	0	1
0	0	0	1	1	0	1
1	1	1	0	0	1	1
1	1	0	1	0	1	1
1	1	0	0	1	1	1
1	0	1	1	0	1	0
1	0	1	0	1	1	1
1	0	0	1	1	0	1
0	1	1	1	0	0	0
0	1	1	0	1	1	1

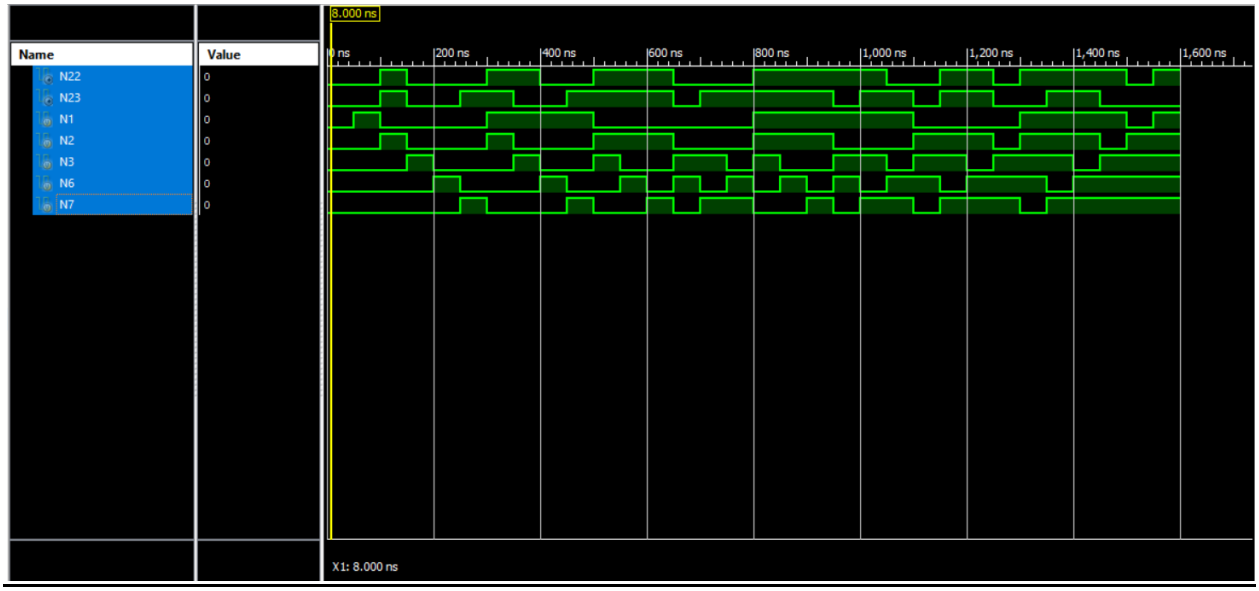
0	1	0	1	1	1	1
0	0	1	1	1	0	0
1	1	1	1	0	1	0
1	1	1	0	1	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	0
0	1	1	1	1	0	0
1	1	1	1	1	1	0

So, there was no anomaly, and thus the Trojan was left undetected.

Normal Behavioral



Behavioral when Trojan at N19



Inference:

Since, the transition probabilities of the triggers used in Experiment1 was quite higher than that of Experiment2, hence , our experimented output support that nodes with higher transition probability can trigger the Trojan easily.

CONCLUSION

Due to the increasing importance of information security and counter security threats , in Hardware Trojan detection fields people have presented various methods to deal with these attacks . In this paper we have studied different types of Trojans and various methods for detection . Mainly , we have studied method based on logic testing to detect Trojan inside a circuit.

The Trojan concern is, however, far from being overcome. Adversaries aware of the main detection methods may develop more sophisticated Trojans able to be undetectable by the known techniques. Nevertheless, while more and more methods are proposed, more difficult is to an attacker to design a Trojan undetectable by all these methods. Consequently, the development of new innovative techniques is the key factor to make difficult possible attacks and thus increases the ICs trustworthiness against Trojans.

REFERENCES

1. Creation and detection of hardware Trojans using non-invasive off-the-shelf technologies : **Catherine Rooney , Amar Seeam and Xavier Bellekens ,2018**
2. Hardware Trojans : Threats and Emerging Solutions : Rajat Subhra Chakraborty , Seetharam Narasimhan , Swarup Bhunia
3. S. Jha and S.K. Jha, "Randomization Based Probabilistic Approach to Detect Trojan Circuits", 11th IEEE High Assurance Systems Engineering Symposium, 2008
4. Huffmire, T.; Irvine, C.; Nguyen, T.D.; Levin, T.; Kastner, R.; Sherwood, The Handbook of FPGA Design Security; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
5. Mitra, S.; Wong, H.S.P.; Wong, S. Stopping hardware Trojans in their tracks. IEEE Spectr. **2015**. Available online: https://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks
6. Yeh, A. Trends in the global IC design service market. DIGITIMES Res. **2012**. Available online: https://www.digitimes.com/news/a20120313RS400.html?chid=2