THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2022, VOL. 45, NO. 1, ISSN 2344-4738, ISSN-L 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS
Article DOI: **https://doi.org/10.35219/eeaci.2022.2.02**

ANALYSIS OF A SEQUENTIAL HARDWARE TROJAN TRIGGERED BY THE TRANSITIONS OF THE RARE INPUT EVENTS

Grigore Mihai Timis, Alexandru Valachi

Technical University "Gh.Asachi"Iasi, Faculty of Automatic Control and Computer Engineering (e-mail: mtimis@tuiasi.ro, avalachi@tuiasi.ro)

Abstract: This paper presents an analysis of a sequential Hardware Trojan(HT) triggered by the rare input events transitions. Hardware Trojans represents a malicious entity who could hijack, leak confidential informations, disable the system etc. Since the Hardware Trojan insertion modifies the functionality of the of the original circuit, it should be treated with maximum importance. The novelty of this paper represents the analysis of the sequential Hardware Trojan (HT) which can be triggered by the rare input events transitions.

Keywords: Hardware Trojan(HT), Sequential Hardware Trojan , Hardware Trojan attacks, Finite State Machine (FSM), Sequential System.

1. INTRODUCTION

The semi-conductor industry has developed so quickly that it involves various companies and countries. Thus, different design phases for an Integrated Circuit (IC) are implemented at geographically distributed locations. Outsourcing of IC design and fabrication has become a popular tendency as well. This tendency has resulted in new security threats such as Hardware Trojan (HT) insertion. A HT can perform many destructive actions such as Denial of Service, leakage of sensible data, damage to functions or performance etc. (Mohammad, et.al., 2010). The existence of HTs has attracted extensive attention. If the chip contained with the HT is deployed in the secure-sensitive fields such as education, military, economy and medical treatment, it may have severe consequences.

Although current research on the detection technology of HTs has made great progress, it is still a challenging problem considering the diversity and unpredictability of HTs. And for different HT detection technologies, attackers can also design corresponding HTs to avoid detection. Maybe we can try to figure out the attackers' mental activities, find the vulnerability of the circuit and design HTs in advance. Doing so will pave the way for HT detection technology and prevent the possible hidden attacks in advance. Therefore, the research on the design and detection of HT is complementary to each other.

As specific literature relates (Su, et.al., 2021),(Li, et.al., 2022),(Farahmandi, et.al., 2019),(Dupuisi, et.al., 2017), the Hardware Trojans (HT) circuits can be usually activated since some specific conditions are meet: power or an output value of a specific logic is activated, an abnormal transition of a digital signal was detected. During the past 10 years, a number of studies on Hardware Trojans design and detection has been developed.

The outline of this paper is as follows: Section II describes the available insertions methods of the Hardware Trojans; Section III outlines the sequential design and sequential Trojans along with the structure of sequential Trojan; Section IV shows the analysis and synthesis of the sequential Hardware Trojan along with a case study example; Section V and VI concludes with a summary and discussion of further development possibilities.

2. . INSERTION METHODS OF HARDWARE TROJAN (HT)

According to (Mohammad, et.al., 2010), (Jain, et.al., 2021), (Tehranipoor, et.al., 2011) Hardware Trojan (HT) insertions can be classified into four categories: HT driver, external, internal trigger.

HT triggers type is usualy associated with an internal event/external/predefined value of a signal. The HT actions can be stored into a DFF sequential system or in system's memory. The driver executes the implementations of the trigger. In (King, et.al., 2008), considering the several attacks, it's implemented the Illinois Malicious Processor.

The existing Hardware Trojans detection techniques has three main categories:

1. side-channel analysis methods that rely on the deviations in area or power consumption to detect Hardware Trojans. However, these methods need a Trojan-free design (golden IC) to be the reference (Haider, et.al., 2019), (Exurville, et.al., 2015), (Tehranipoor, et.al., 2011), (Aghion, et.al., 2009).

2. functional testing methods that apply input test vectors to detect possible Hardware Trojans (Zhang, et.al., 2013), (Waksman, et.al., 2014), (Flottes, et.al., 2015), (Zhang, et.al., 2017), (Jacobr, et.al., 2014). 3. circuit analysis methods that identify suspicious modules by analysing the function and features of the circuit. Circuit analysis methods have received much attention because they do not require the modification of the original design. This technique focuses on distinguishing inputs which do not affect outputs during testing. In (Hicks, et.al., 2011) is invoked a new technique named Unused Circuit Identification (UCI). This method identify suspicious and potentially malicious circuits at the design phase. Since UCI is sensitive to the Hardware Trojans implementation, it can only detect a small set of Hardware Trojans. In (Waksman, et.al., 2014) is presented FANCI, which applied scalable and approximate Boolean functional analysis to detect suspicious wires. In (Zhang, et.al., 2013) is designed a new verification technique for Hardware Trojans detection, namely VeriTrust.

The main advantage of VeriTrust is that it's insensitive to Hardware Trojans implementation styles. By examining verification corners, VeriTrust automatically identifies potential Hardware Trojans trigger inputs which are not sensitised by verification test cases.

On the basis of FASTrust in (Yao, et.al., 2015), it was extended FASTrust to a multilevel FASTrust verification (ML-FASTrust) framework. It can pick up a series of representative structural and quantified features. Then the analysis is applied to detect Hardware Trojans in 3PIP cores. It analyses Hardware Trojans features on the flip-flop level Control-Data Flow Graph of the circuit. It is able to identify existing explicitly triggered and implicitly triggered Hardware Trojans. Experiments are proven that this method can detect all Hardware Trojans from Trust-Hub, systematic Hardware Trojans design methodology and DeTrust (Zhang, et.al., 2014).

There are new methods of emerging Hardware Trojans detection based on the register-transfer level (RTL) features. In (Kok, et.al., 2019) is proposed the Minimum Redundancy Maximum Relevance feature selection to improve the effectiveness of RTL Trojan features. Also is presented an Hardware Trojans vulnerability analysis for the macro block in the RTL design based on rare triggering nets' estimation. It was extracted circuit features from the RTL source codes and built a training database. In (Muraoka, et.al., 2019) it's proposed the RTL nodes' partition method and novel features, applying the random forest algorithm to detect RTL Trojans effectively. Also a systematic Hardware Trojans design containing two Hardware Trojans coding models are proposed. In (Hicks, et.al., 2011) was designed a Hardware Trojans to ensure that the UCI algorithm will not flag the circuitry between input and output as potentially malicious. The Hardware Trojans are hidden as a 'useful circuit' to evade the UCI algorithm. In (Zhang, et.al., 2014) was presented a systematic Hardware Trojans design methodology, namely DeTrust. DeTrust implements stealthy implicit triggers, firstly it spreads the trigger logic to multiple sequential levels and combinational logic blocks, then it combines the trigger logic with the normal logic. So it can defeat the detection of FANCI and VeriTrust methods. However, it is possible to extend FANCI and VeriTrust to trace and verify signals across multiple sequential levels to defend against DeTrust. Later, in (Su, et.al., 2021) was introduced four crucial properties (d, t, α and l) to determine the stealthiness of Hardware Trojans. Among them, d denotes trigger signal dimension, t denotes payload propagation delay, α refers to implicit behaviour factor and l denotes trigger signal locality. A larger value of these properties means higher stealthiness.

Figure 1 shows the HT insertion flow: The Hardware Trojan designer model the HT behavior using a finite-state machine. Thus, by altering the finite state machine state, the HTH can be inserted into the original system. The HTH finite state machine should have an input trigger but the driver is hidden into the finite state machine structure.

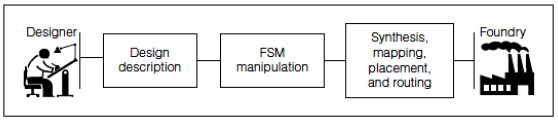

Fig.1. Design Process Flow

In this paper, we describe a possible way to activate a sequential Hardware Trojan using the rare inputs events transitions on a Finite State Machine (FSM) for a sequential system, in this way altering the FSM's states.

Using this method, it's providing a way to hide cases when the rare inputs values are triggering the Hardware Trojan, so it increases the effect of the Hardware Trojan. The proposed Hardware Trojan has advantages in terms of stealthiness, general applicability and imperceptibility. The stealthiness is proven in Hardware Trojan's ability to camouflage as a normal circuit, since the trigger inputs have major impact on the FSM's states and output signals.

The generality of the proposed sequential Hardware Trojan means that it can be inserted in any circuits that uses a combination of the rare input values rather than some particular circuits. Mostly depending by the Hardware Trojan's complexity triggering logic, in general it has almost negligible overhead on area and power consumption.

3. SEQUENTIAL DESIGN AND SEQUENTIAL TROJANS

In figure 2 is shown the structure of a sequential digital system with sequential Hardware Trojan inserted. The sequential FF(FlipFlop) elements works together with the combinational logic in a "genuine circuit". Primary inputs and outputs are usually connected to combinational logic.

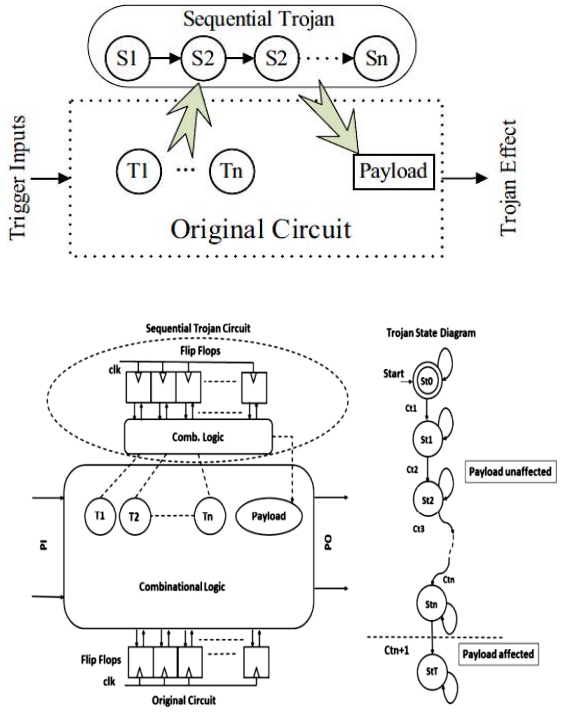

Fig.2. Structure of Sequential Trojan attached to the Original Circuit

Notice that it becomes challenging to generate test patterns to apply appropriate values on flip-flops inside the circuit. In this case, the sequential Hardware Trojan is inserted in a such way as it can have triggers based on rare input events happening in the genuine circuit. Sequential Trojan can be described as a FSM (Finite State Machine), where state transitions are mapped to rarely observed events in the circuit.

The logic to determine next-state usually involves a combination of rare logic values inside the genuine circuit, otherwise, the Trojan stays in the same state. The Trojan output is activated when it reaches a final state (StT as shown in the figure 2) when the payload node is affected and implements malicious behaviour intended by the Trojan. According with (Mohammad, et.al., 2010) this is an internal trigger type.

4. SEQUENTIAL TROJAN HORSE. A CASE STUDY EXAMPLE

We propose to design a digital system which simulates a two color led semaphore who is usually used on the train railway sections.

We consider the following functionality:

Green – *Clear*. The train may proceed subject to any speed restrictions applying to the section of line or to the train itself.

Red – Danger/Stop

The system has three buttons (B_1,B_2,B_{res}) and two leds(Green, Red) as in figure 3.

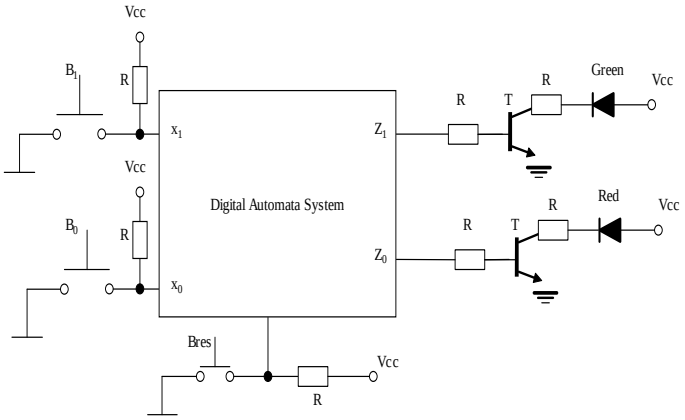

Fig.3 Digital Automata System. Two led semaphore.

These three buttons will provide three inputs into the Digital Automata System:

-Push button B_1 will provide x_1 input with the following functionality: when the button is pushed(ON) the input x_1=1, when button is released(OFF) the input x_1=0.

-Push button B_0 will provide x_0 input with the following functionality: when the button is pushed(ON) the input x_0=1, when button is released(OFF) the input x_0=0.

-B_{res} will provide a system hard reset so both leads will be off and the digital automata system will provide outputs z_1=0, z_0=0.

If the output signal z_1=1 it means that the green led will be on, if the output signal z_0=1 means that the red led will be on. These two led will be ON after the completion of the following sequence, (1):

$$B_1B_0 : OFF\ \_\ OFF \rightarrow OFF\ \_\ ON \rightarrow ON\ \_\ ON \rightarrow$$
$$\rightarrow ON\ \_\ OFF \rightarrow OFF\ \_\ OFF \Rightarrow$$
$$\Rightarrow GreenLed = ON, \text{Re}\, dLed = OFF$$

$$B_1B_0 : OFF\ \_\ OFF \rightarrow ON\ \_\ OFF \rightarrow ON\ \_\ ON \rightarrow$$
$$\rightarrow OFF\ \_\ ON \rightarrow OFF\ \_\ OFF \Rightarrow$$
$$\Rightarrow GreenLed = OFF, \text{Re}\, dLed = ON \quad (1)$$

Based on the sequence from figure 4, it can be drawn the golden model for the FSM fluence graph, figure 5. It can be observed that the green led will be ON when the flow reached the state S5, similarly the red led will be ON when the flow reached the state S9. In the other intermediate states, those two leds will be OFF.

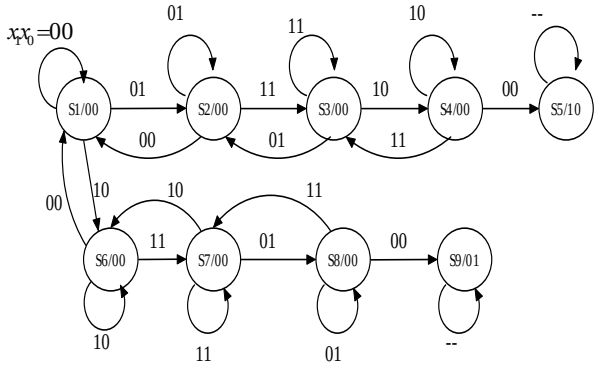

Fig.4. FSM - golden model

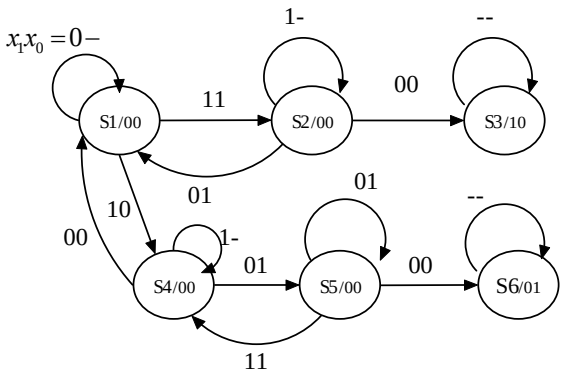

Fig.5. Optimised FSM

As can be observed, the FSM's golden model is not contains any sequential Hardware Trojan malicious states. In order to optimise the number of the states, based on the equivalent states, the optimised golden model fluence graph becomes as in figure 5.

The hardware implementation using flip-flop CBB J-K and logic gates with an implementation cost C=40 is shown in figure 6.

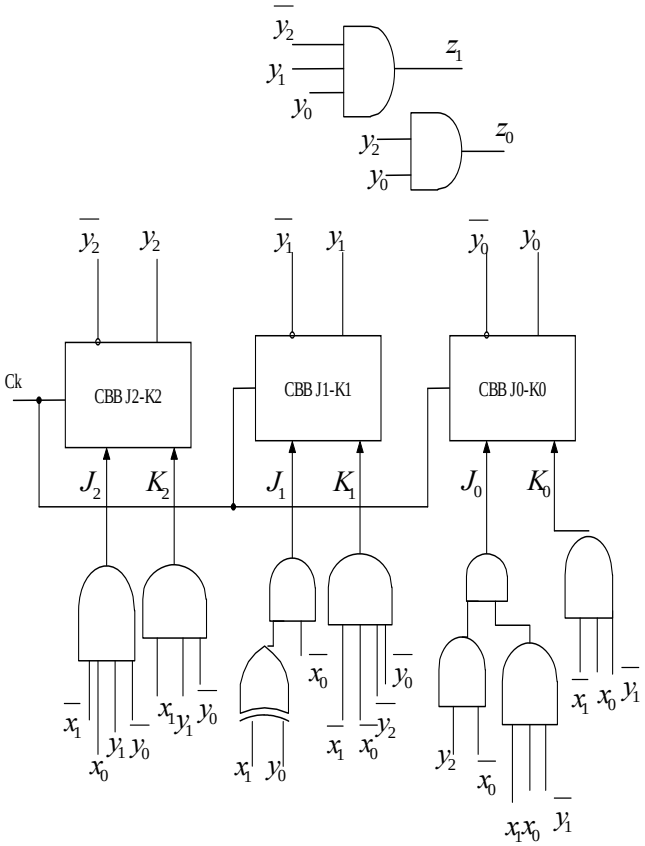

Fig.6. Hardware Implementation with J-K type circuits

Based on figure 6, the logic equations which describe the FSM's golden model are (2):

$$J_2 = \overline{x_1} \cdot x_0 \cdot y_1 \cdot \overline{y_0}$$
$$K_2 = x_1 \cdot y_1 \cdot \overline{y_0}$$
$$J_1 = \overline{x_0} \cdot (x_1 \oplus x_0)$$
$$K_1 = \overline{x_1} \cdot \overline{x_0} \cdot \overline{y_2} \cdot \overline{y_0} \quad (2)$$
$$K_0 = \overline{x_1} \cdot x_0 \cdot \overline{y_1}$$
$$Z_1 = \overline{y_2} \cdot y_1 \cdot y_0$$
$$Z_0 = y_0 \cdot y_0$$

4.1. Trojan Horse activation

In the following section we will consider that the sequential Hardware Trojan is inserted in a such way as it can have triggers based on rare inputs events happening in the genuine circuit, thus the sequential Trojan can be described as a hijacked FSM(Finite State Machine), where state transitions are mapped to rarely observed inputs events in the circuit.

This scenario can be repeated anytime based on the triggering of the rare inputs events.

Since the push buttons can oscillate when are pressed/depressed, these kind of events in the golden model circuit can be speculated by the FSM sequential Hardware Trojan.

Based on the buttons B_1 B_0 oscillations with rare values $00(01,10,00,11) \rightarrow 11(00,01,10,11) \rightarrow$, thus $- > 01(11,10,00,01) \rightarrow 10(00,01,11,10)$ Hardware Trojan will exploit this vulnerability and will hijack the FSM to state S_7 instead the normal state S_6.

This will lead to state S_9 where both leds are ON (11), so this action represents a semaphore dangerous behaviour since it may generate severe consequences.

Thus, the hijacked digital system will contain also the malicious FSM state added by the Hardware Trojan, named as payload, like in figure 7.

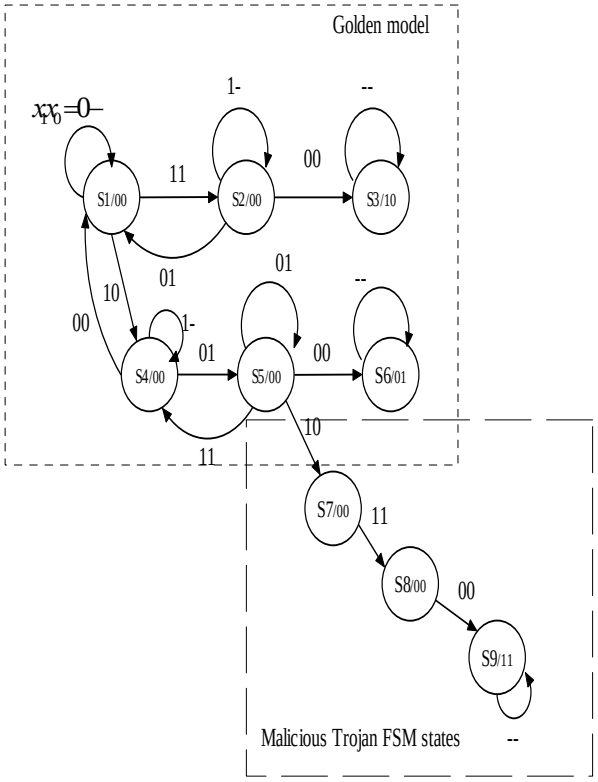

Fig.7. Added the malicious FSM Trojan states to the golden model

Figure 7 presents one of the multiple possibilities to hijack the finite state machine's behavior, thus if the FSM's state flows from state S6 to state S7 due of wrong input value 10, this will lead to state S9 where both leds will be ON.

Assuming that this digital system is used on a real railway system, this behavior could lead to dangerous functioning cases. The equations for the hijacked FSM system are described in (3).

$$D_2 = y_{2,n+1} = \overline{x_1} \cdot x_0 \cdot \overline{y_1} \cdot y_0 + x_1 \cdot y_2 + \overline{x_0} \cdot y_2 + y_2 \cdot y_0 + y_2 \cdot \overline{y_1}$$
$$D_1 = y_{1,n+1} = x_0 \cdot y_1 + x_1 \cdot \overline{y_2} \cdot y_1 + \overline{x_1} \cdot y_2 \cdot y_1 + \overline{x_1} \cdot \overline{x_0} \cdot \overline{y_2} \cdot y_1 + y_1 \cdot y_0$$
$$D_0 = y_{0,n+1} = y_1 \cdot y_0 + y_1 \cdot \overline{y_0} \cdot \overline{x_0} + x_1 \cdot \overline{x_0} \cdot y_0 + \overline{y_2} \cdot y_0 + x_1 \cdot \overline{y_2} \cdot \overline{y_0}$$
$$Z_1 = \overline{y_2} \cdot y_1 \cdot y_0 + y_2 \cdot \overline{y_1} \cdot \overline{y_0}$$
$$Z_0 = y_2 \cdot y_1 \cdot y_0 + y_2 \cdot \overline{y_1} \cdot \overline{y_0} = y_2 \cdot (\overline{y_1 \oplus y_0})$$
(3)

The hardware implementation using three flip-flop CBB D and logic gates with an implementation cost C=80 is shown in figure 8.

It can be observed that the hijacked Trojan implementation is using more logic gates compared with the golden model one. This scenario can be detected by a Trojan Detection algorithm but this will be presented in a future research paper.

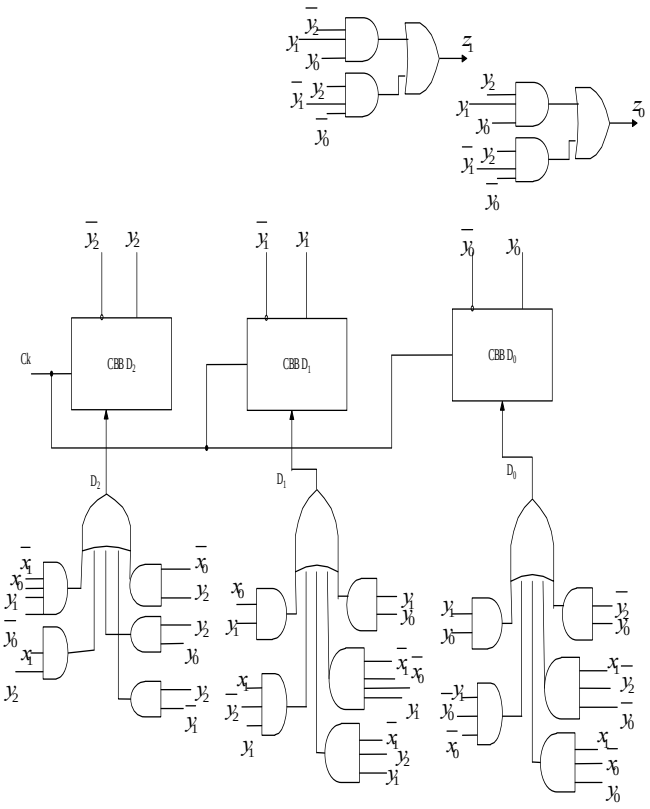

Fig.8. Hardware Implementation with D type circuits

Moreover, the advantage of the FSM-based Trojan is that they can be designed to be arbitrarily complicated with same amount of resource and can re-use both combinational logic and flip-flops (FF's) of the original circuit for FSM-hosting. In some cases, the FSM-based Trojan can have state transitions leading back to the initial state, thus causing the final Trojan state to be reached only if the entire state sequence is satisfied in consecutive clock cycles.

5. CONCLUSIONS AND FUTURE WORK

In this paper we focused on sequential Trojans triggered by a sequence of rare events. The novelty of this paper represents the analysis of the sequential Hardware Trojan (HT) which can be triggered by the rare input events transitions.

Through examples, analysis, and results, we have shown that Trojans inserted in foundries can have several impact on chip's functionalities, manufacturing & costs. The Hardware Trojan design and placement approaches presented are effective for both FPGA and ASIC platforms. Since it's trigger under extremely rare conditions, it is also difficult to detect these Hardware Trojans in functional testing.

6. REFERENCES

Mohammad Tehranipoor and Farinaz Koushanfar, "A survey of hardware trojan taxonomy and detection", IEEE Design & Test of Computer, 27:10-25,2010.

Syed Kamran Haider, Chenglu Jin, Masab Ahmad, Devu Shila, Omer Khan, Marten van Dijk, "Advancing the State-of-the-Art in Hardware Trojans Detection", IEEE Transactions on Dependable and Secure Computing (Volume: 16, Issue: 1, Jan.-Feb. 1 2019).

Zie Zhang; Feng Yuan; Lingxiao Wei; Zelong Sun; Qiang Xu "VeriTrust: Verification for hardware trust" 50th ACM/EDAC/IEEE Design Automation Conference (DAC), 29 May-7 June 2013, IEEE.

A. Waksman, Matthew Suozzo, S. Sethumadhavan, "FANCI: identification of stealthy malicious logic using boolean functional analysis", Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, Published 2013, IEEE.

Matthew Hicks, Cynthia Sturton, David Wagner, Defeating UCI: Building Stealthy and Malicious Hardware, Proceedings of the 2011 IEEE Symposium on Security & Privacy.

Song Yao; Xiaoming Chen; Jie Zhang; Qiaoyi Liu; Jia Wang; Qiang Xu; Yu Wang; Huazhong Yang, FASTrust: Feature analysis for third-party IP trust verification, Published in 2015 IEEE International Test Conference (ITC), 06-08 October 2015, DOI: 10.1109/TEST.2015.7342417.

Jie Zhang, Feng Yuan, Qiang Xu, "DeTrust: Defeating Hardware Trust Verification with Stealthy Implicitly-Triggered Hardware Trojans", CCS '14: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications SecurityNovember 2014 Pages 153–166.

ML Flottes, S Dupuis, PS Ba and abd B Rouzeyre, "On the limitations of logic testing for detecting Hardware Trojans Horses", International Conference on Design & Technology of Integrated Systems in Nanoscale Era IEEE,pp.1-5,2015.

Chee Hoo Kok; Chia Yee Ooi; Michiko Inoue, Net Classification Based on Testability and Netlist Structural Features for Hardware Trojan Detection, Published in 2019 IEEE 28th Asian Test Symposium (ATS), 10-13 December 2019, DOI: 10.1109/ATS47505.2019.00020.

Kenichi Kawaguchi Chie Iwasaki Michiaki Muraoka, A RTL Partitioning Method with a Fast Min-Cut Improvement Algorithm, Semiconductor Research Center, Matsushita Electric Industrial Co., LTD.
3-1-1, Yagumo-Nakamachi, Moriguchi, Osaka 570

Yu Su; Haihua Shen; Renjie Lu; Yunying Ye, A Stealthy Hardware Trojan Design and Corresponding Detection Method, Published in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 22-28 May 2021, DOI: 10.1109/ISCAS51556.2021.9401770

W. Hu, L. Zhang, A. Ardeshiricham, J. Blackstone, B. Hou, Y. Tai, et al., "Why you should care about don't cares: Exploiting internal don't care conditions for hardware Trojans", pp. 707-713, 2017.

Ayush Jain; Ziqi Zhou; Ujjwal Guin, "Survey of Recent Developments for Hardware Trojan Detection", 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 22-28 May 2021, DOI: 10.1109/ISCAS51556.2021.9401143, IEEE.

I. Exurville, L. Zussa, J.B. Rigaud and B. Robisson, "Resilient Hardware Trojans Detection based on Path Delay Measurement", In International Symposium on Hardware-Oriented Security and Trust (HOST'15), pp. 151–156, 2015.

M Tehranipoor, H. Salmani, X. Zhang, X. Wang, R. Karri, J. Rajendran and K. Rosenfeld, "Trustworthy Hardware: Trojan Detection and Design-for-Trust Challenges", In IEEE Computer, pp. 66–74, 2011.

Ovidiu Ursaru, Cristian Aghion, Mihai Lucanu, Liviu Tigaeru, "Pulse width Modulation Command Systems Used for the Optimization of Three Phase Inverters", Advances in Electrical and

Computer Engineering Journal. Suceava, Romania, 2009, pag.22-27.

N. Jacob, D. Merli, J. Heiszl and G. Sigl, "Hardware Trojans: current challenges and approaches", in IET Computers & Digital Techniques, 8(6):264–273, 2014.

Dejian Li, Qizhi Zhang, Hardware Trojan Detection Using Effective Property-Checking Method Electronics 2022, https://doi.org/10.3390/electronics11172649

Farimah Farahmandi; Prabhat Mishra, FSM Anomaly Detection Using Formal Analysis, Published in 2017 IEEE International Conference on Computer Design (ICCD), DOI: 10.1109/ICCD.2017.55

Sophie Dupuis, Marie-Lise Flottes, Giorgio Di Natale and Bruno Rouzeyre, Protection against Hardware Trojans with Logic Testing: Proposed Solutions and Challenges Ahead, DOI 10.1109/MDAT.2017.2766170, IEEE Design

Samuel T. King, Joseph Tucek, Anthony Cozzie, Designing and implementing malicious hardware, LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent ThreatsApril 2008 Article No.: 5Pages 1–8, Published:15 April 2008