Exploring Hard to Detect Sequential Hardware Trojans

Nilanjana Das
Department of Computer Science & Technology
IIEST Shibpur, India
nilanjanadas.rs2017@cs.iiests.ac.in

Joy Halder
Department of Computer Science & Technology
IIEST Shibpur, India
jhalder.rs2016@cs.iiests.ac.in

Baisakhi Das
Department of Information Technology
Institute of Engineering & Management
Kolkata, India
baisakhi.das@iemcal.com

Biplab K Sikdar
Department of Computer Science & Technology
IIEST Shibpur, India
biplab@cs.iiests.ac.in

***Abstract*—Hardware Trojans embedded in the modern Integrated Chips (ICs) is the most adverse threat due to its stealthy nature and rare triggering occurrence. In this work, we have proposed a novel triggering mechanism of an Hardware Trojan model in the Finite State Machine. In particular, this work illustrates the design and placement of sequential Hardware Trojans, which are rarely activated/ observed during basic testing of a chip. The proposed trojan model along with examples, theoretical analysis of effectiveness and simulation results are evaluated to determine the impact of this Trojan in a sequential circuit. It is shown that careful design and placement of the sequential Trojan can evade existing trojan detection approaches.**

***Index Terms*—Sequential Hardware Trojan, Sequential Circuit, Finite State Machine, Trigger Condition and Trigger Signal, Trojan Detection.**

I. Introduction

Nowadays, Integrated Chips (ICs) have been used in the information infrastructure and systems that contain confidential information, for example, financial information, secret key data. It is necessary for these ICs to have high secrecy. Globalization during the IC manufacturing decreases control of a designer on the fabricated chips. The designer can not entirely handle the whole process of designing as well as manufacturing. Therefore, it is extremely difficult to control the whole process of an IC design. The designing phase is likely to be fabricated by the attackers through the plantation of Hardware Trojans (HTs) [1].

Several untrusted authorities during the manufacturing phase can potentially compromise an IC's functional and/ or parametric behavior such that it can evade conventional IC testing approaches. Such alterations of a design can be introduced at different stages of the IC manufacturing phase, e.g. malicious insertion in an Intellectual Property (IP), modification of netlist or tampering specific files during fabrication. These can have adverse effect during in-field operations, especially in the security-critical applications such as defence, communication and national as well as international infrastructure [1][2].

Trojan circuits disable the normal behaviour of ICs by leaking secret information, or by destroying the normal functionalities of ICs [3][4]. Generally, Trojan circuits have a trigger unit and a payload unit. The trigger unit determines whether the values of the signal lines and states meet the activation conditions, set by the attackers. The activation conditions are usually rare in normal operation. Finally, the payload unit executes the attack.

Several methods to detect Trojans have been proposed [4][5]. The detection methods are divided into two categories. One verifies the behaviors of circuits and the other category analyzes side-channel parameters such as leakage current, dynamic power, path-delay characteristics, or noise. When a Trojan affects the behavior of the circuit rapidly, the detection of Trojan becomes easier. These are verification-based approaches [4]. If size of the Trojan circuit increases, the amount of side-channel information such as the leakage current of the Trojan circuit also increases, so detection of the Trojan becomes easier. These approaches are called side-channel analysis-based approaches [4]. These particular cases where a Trojan circuit affects the behavior of the circuit and the size of Trojan also have a major impact on the detection of Trojan circuits.

Many research works have been done so far based on Combinational Hardware Trojan [6]. Bhunia.et.al describes Sequential Hardware Trojan (HT) in their research work [7] and demonstrated possibility of different types of sequential HTs in which one of the possibility is the Finite State Machine (FSM) based Trojan.

Based on this scenario, this work proposes an Sequential Hardware Trojan model where the trojan mainly attacks the FSM of the sequential circuit. The proposed trojan is implemented using the MOD up-counter. The counter will reach to an unreachable state by the trojan trigger condition. This

978-1-7281-9369-4/20/$31.00 © 2020 IEEE

unreachable state works as the trigger signal which executes the payload operation. The probability of triggering condition is also computed and it is proved that the designed trojan is extremely rare and also random such that it is hard-to-detect with the present conventional detection methodologies. The main focus of the work is to draw attention on sequential circuit based trojan which are more vulnerable than combinational trojan [7]. Three important factors - present state, next state and clock pulse operate a sequential circuit where hampering any of this can make a harm in a circuit. We have designed our HT model based on these parameters and demonstrated it's malicious behaviour.

The remainder of this paper is organized as follows. Section II describes a brief introduction of hardware trojan along with a short description of current detection algorithm. In Section III, the paper presents a sequential circuit based trojan model with MOD-N counter. Section IV evaluates the probability of trigger condition and signal and verified that it is difficult to detect with the detection algorithm. Finally Section V concludes this paper with future work.

II. A Brief on Trojan Circuits

The Hardware Trojans (HTs) are malicious alterations done by some adversaries. The purpose of implementing HT is to harm a secure system by leaking any secret data or destroying the whole system. To avoid any detection method, the HT is triggered rarely but the triggering condition is random in nature. The probability of rare and random properties of HT makes it harder to detect. The HTs can be of two types: Combinational HT and Sequential HT [5].

A. Combinational and sequential hardware trojan

The activation of a Combinational Hardware Trojan depends on the occurrence of a particular condition at certain internal wires of the circuit. On the other hand, activation of Sequential Hardware Trojan depends on the occurrence of a specific sequence of rare logic values at internal wires [7].

The traditional verification and testing scheme detects an HT by triggering it, and hence designing a rare trigger condition makes the HT silent during any testing scheme. Therefore, the efficient trigger modelling becomes the key issue for HT implementation.

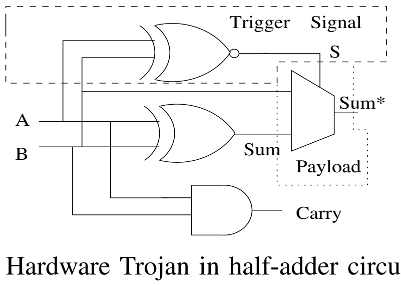

Fig. 1: Hardware Trojan in half-adder circuit [11]

The *Trigger Condition* is an event, manifested in the form of a particular Boolean value of certain internal/ external wires of the circuit, that activates the HT trigger circuitry. The *Trigger Signal or Trigger State*, on the other hand, is a set of physical wire(s) which the HT trigger circuitry asserts in order to activate the payload circuitry once a trigger condition is satisfied. The *Trojan Payload* is the part of the circuit or the functionality affected by the activation of the Trojan trigger condition and signal.

A trojan infused half adder circuit is shown in Figure 1 [11]. Here, $A = B$ is the *Trigger Condition*, the select line S of the multiplexer is the *Trigger Signal* which becomes '1' when $A = B$ (HT is triggered). After the completion of *Trigger Condition* and activation of *Trigger Signal*, the *Trojan Payload* will take place; in this case the payload is $Sum = B$ instead of $Sum = A \oplus B$.

B. Trojan detection method HATCH

Although there are many trojan detection approaches presented in recent days: UCI [8], Veritrust [9], Fancy [10] e.t.c, the trojan models are always one step ahead of the trojan detection approaches. Recently in [11], the authors have proposed *the state of the art* in the hardware trojan detection by an efficient algorithm namely HATCH. The four properties *Trigger Signal Dimension* $d(T)$, *Payload Propagation Delay* $t(T)$, *Implicit Behavior Factor* $\alpha(T)$ and *Trigger Signal Locality* $l(T)$ are defined in [11] to measure the stealthiness of an HT circuit. A Trojan detection scheme, without a well defined scope on the landscape of HTs, fails to provide concrete security guarantees. Therefore, the properties d, t, α, l, thus introduced, determine the stealthiness of a deterministic HT having a set of trigger states T [12].

The $d(T)$ represents the number of wires used by the HT trigger circuitry to activate the payload that exhibits malicious behavior. A large d complicates trigger signal. It implies that HT is harder to detect [11][12].

The $t(T)$ is the number of cycles required to propagate a malicious behavior to the output port of a chip after an HT is triggered. A large t means it takes a long time, after triggering, to activate the malicious behavior. That is, it is less likely the HT will be detected during testing [11][12].

The $\alpha(T)$ represents the probability that given an HT gets triggered. That is, the HT will not (explicitly) manifest malicious behavior, Such an HT shows implicit behavior. Higher probability of implicit malicious behavior means higher stealthiness during testing phase [11][12].

The $l(T)$ shows the spread of trigger signal wires of the HT across the IP core. Small l value implies that the wires are in the close vicinity of each other. For large l value, it is harder to figure out exactly which wires form the trigger signal and, therefore, the HT is harder to detect [11][12].

The main purpose to introduce [11] is that we have mapped our proposed trojan model with these parameters and proved that a trojan with large d, t and l is difficult to detect in real life scenario. Our proposed Hardware Trojan is represented by multiple sets of trigger states T, each having their own d, t and l values. The quadruple $< d; t; \alpha; l >$ defines the achievable region of the HT.

III. Sequential Trojan With MOD-N Counter

A sequential trojan exhibits its malicious outcome after a sequence of rare events, during a long period of general operations- that is, acting as a time-bomb [11]. Generally, a sequential trojan having an increasing length of trigger sequence is exponentially harder to detect [12]. This section reports the sequential trojan model, using a generalized MOD-N up-counter. The MOD-N counter is designed with n flip-flops, where $2^{n-1} < N \leq 2^n$. The $N < 2^n$ is a favourable condition for trojan insertion in our proposed model.

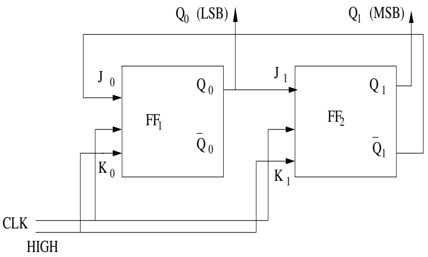

Fig. 2: MOD-3 counter

The $(2^n - N)$ states of the sequential circuit realizing MOD-N counter are out-of-scope count (unreachable state) for this counter. The trojan, explored in this work, can utilize any of the out-of-scope counts. For example, a MOD-3 counter (Figure 2) counts from 00 to 10. A trojan infected counter circuit forcefully can count 11 (out-of-scope count value) for a rare sequence.

A. FSM based trojan using MOD-N counter

A MOD-N up counter counts 0 to (N-1) states. The MOD-N up counter can be modified in such a way that when it is in $(N-1)^{th}$ state, it switches to state N and then returns to state 0 when some trigger condition is fulfilled. A P-bit LFSR is used to generate random patterns of length P and a pre-defined pattern x of the LFSR is considered as the trigger condition. Thus when the counter is in $(N-1)^{th}$ state and the LFSR generates pattern x simultaneously, the counter switches to state N in the next clock cycle rather to switch state 0. In the next subsection, an example using MOD-3 counter is modelled to clarify this idea.

B. An example design with MOD-3 up counter

The basic functionality of the MOD-3 counter is to count from '00' to '10'. The embedded trojan forces the counter to abnormally count from 00 to 11. Table I and Table II describe the present state-next state relationship of the MOD-3 counter and the trojan infused MOD-3 counter as shown in Figure 3. We have used J-K flip flops to configure the MOD-3 counter. The excitation table for the J-K flip flop is given in Table III. Table IV presents the excitation table of the MOD-3 counter of Figure 2. It shows that the Q_1, Q_0 pair never be 11. Table V and Figure 4 present the excitation table and the logic diagram of the trojan infused MOD-3 counter respectively. Two multiplexers are added to the original circuit which help the trigger signal to successfully start the payload operation as shown in the figure. The output of the AND gate connected to the LFSR becomes '1' when the predefined pattern x generates by the LFSR. This output is the select line of both the multiplexers. These multiplexers alter the values of two inputs in Figure 4 when the select line is '1', i.e. in original circuit $J_0 =' \overline{Q}_1'$, $K_1 =' 1'$ where after trigger operation $J_0 =' 1', K_1 =' Q_0'$.

TABLE I: State transition of MOD-3 counter

$PresentState$	$NextState$
00	01
01	10
10	00
dd	dd

TABLE II: Trojan infused MOD-3 counter

Present State	Next State
00	01
01	10
10	11
11	00

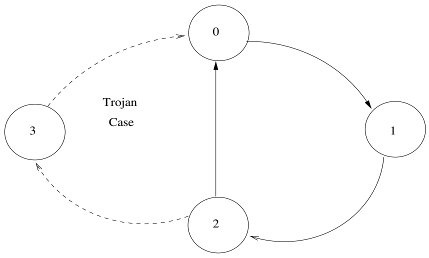

Fig. 3: Trojan on MOD-3 counter

A crucial property of trigger function is that it must be random in nature such that it will appear in rare cases and the probability of it's activation must be greater than 0. A trigger function with activation probability zero, in some cases, through out a life span of the digital circuit, will not be considered as a stealthy trojan as the trigger function may remain dormant. The pseudo random number generator (PRNG) can best suite in this case to make a trojan activation probability greater than 0 and also performing as random as possible to make the trojan condition rare in nature [11]. In the current scenario, the particular trojan condition will appear in the LFSR. The particular random pattern (x) fixed by the adversary will initialize the first activation property of the respective trojan.

To make the trojan undetectable, we have used two trojan conditions that should be satisfied simultaneously. The trigger condition initializes the trojan activation procedure. In this case, both the LFSR and present state work as trigger condition. Each of this case alone is inadequate to fulfil the trigger condition. The LFSR generated random patterns pass through an AND gate such that the output of the AND gate becomes one when the predefined pattern x appears. After successful completion of this step, trigger signal activates. Here, select lines of two multiplexers work for the trigger signal.

TABLE III: Excitation table of J-K Flip-Flop

Q_n	Q_{n+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

TABLE IV: Excitation table of MOD-3 Counter

P.S		N.S		J_1	K_1	J_0	K_0
q_1	q_0	Q_1	Q_0				
0	0	0	1	0	d	1	d
0	1	1	0	1	d	d	1
1	0	0	0	d	1	0	d
1	1	d	d	d	d	d	d

In Figure 4, the trigger condition is satisfied when following two conditions occur simultaneously:

(i) The 4 bit LFSR generates pattern 'x' which is '1111'.
(ii) The present state of the MOD-3 counter is '10'.
The trigger signal is satisfied when the select line becomes '1'.

IV. Probability of Trigger Condition and Trigger Signal

The triggering condition of the Trojan signal should be rare and random simultaneously. In the current implementation, we employ a P-bit LFSR which generates L distinct patterns, clearly $2^P \geq L$. Let x is the LFSR generated particular pattern for the trigger condition. The n-flip flop MOD-N counter is used to design the circuit, where $2^n > N$. Then we can define:

1. The probability of a particular pattern x appeared in the LFSR is $P(X) = \frac{1}{L}$,

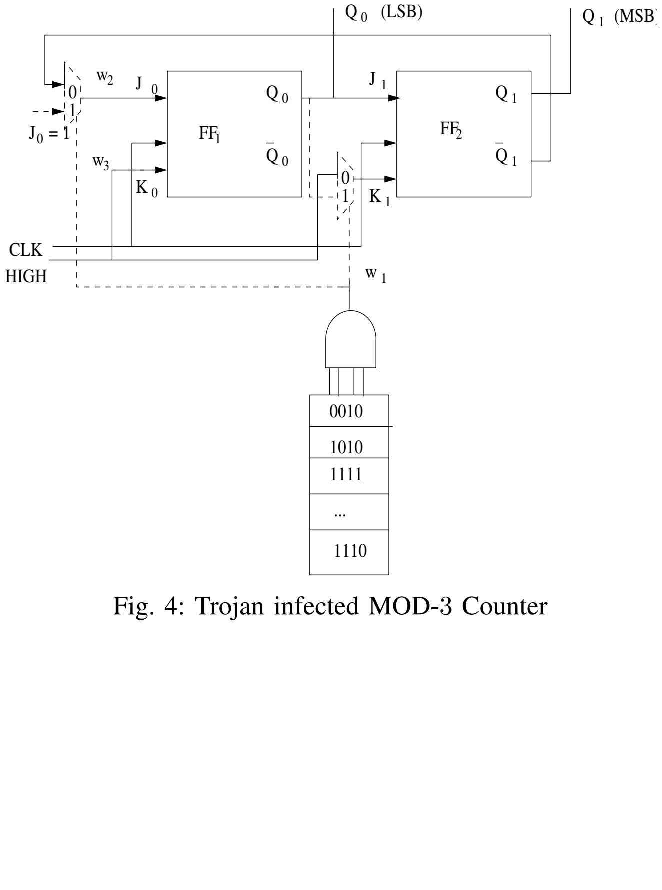

Fig. 4: Trojan infected MOD-3 Counter

TABLE V: Excitation table of trojan infected MOD-3 counter

P.S		N.S		J_1	K_1	J_0	K_0
q_1	q_0	Q_1	Q_0				
0	0	0	1	0	d	1	d
0	1	1	0	1	d	d	1
1	0	1	1	d	0	1	d
1	1	0	0	d	1	d	1

2. The probability that the counter is in the $(N-1)^{th}$ state is $P(Y) = \frac{1}{N}$.

The proposed trojan circuit fails to trigger when in the same clock cycle (i.e. state):

Case 1:
(i) The LFSR generates the pattern x but
(ii) The MOD-N counter is not in the $(N-1)^{th}$ state.

or

Case 2:
(iii) The MOD-N counter is in $(N-1)^{th}$ state but
(iv) The LFSR generated pattern is other than x.

For the above two cases, the MOD-N counter works normally. The trojan circuit works in such a way that in spite of the particular pattern x generated from LFSR, the counter still counts correctly as the state is not matched with the predefined trigger state condition. Therefore, in this situation trojan circuits can evade any detection method as the working principle of the counter is not violated by the trojan circuitry. This makes the trigger condition more challenging to detect with the present conventional trojan detection methods.
Hence, the required probability for the above condition, where the trigger circuitry is activated but the original circuit is still doing its job, is

$$P(T_f) = P(X) \cdot P(\overline{Y}) = \tfrac{1}{L} \cdot (1 - \tfrac{1}{N}).$$

A. Condition to show explicit behavior

The reported trojan circuit can show an explicit behavior when in the same clock cycle (i.e. state):

(i) The LFSR generates the pattern x and
(ii) The MOD-N counter is in the $(N-1)^{th}$ state.

Hence the probability of the explicit behavior for the proposed model is

$$P(E) = P(X) \cdot P(Y) = \tfrac{1}{L} \cdot \tfrac{1}{N}.$$

For example, for a MOD-3 counter with a 4-bit LFSR, which generates 15 distinct patterns

$$P(T_f) = \tfrac{2}{45} \text{ whereas } P(E) = \tfrac{1}{45}.$$

B. Stealthiness of the hardware trojan

As mentioned in the earlier subsections that the two conditions- the LFSR generates the pattern x and the MOD-N counter is in the $(N-1)^{th}$ state, are to be simultaneously satisfied to enable the trojan. From Figure 4, it is observed that the select line (w_1), the present state (q_1, q_0) or say, wire (w_2) and (w_3) are needed to trigger the trojan. When (w_1) is

'1', (w_2) and (w_3) are '1' and '0' respectively, the MOD-3 counter is in an abnormal state '11'. Here, the dimension (d) of the trigger condition (Section II-B) is 3- that is, three wires are required to activate the trojan. The clock cycle also takes part as it is a sequential circuit. When the trigger conditions are fulfilled in the clock cycle t, the adverse effect is observed at the clock cycle $(t+1)$- that is, after 1 clock cycle of the trigger condition, the abnormal state appears. Therefore, (t) is 1 in this case. The l, as defined in Section II-B, is infinite here because the wires responsible for the trojan activation are distributed over the circuit. The implicit case is not considered and not applicable for such trojan cases. Therefore, the trojan reported in this work is solely based on three parameters d, t and l. It can be claimed that with this large number of trigger dimension and the consideration of clock cycle in trojan design make the trojan vulnerable and hard to detect in real life scenario.

C. Evaluation of the reported trojan

For evaluation, we consider a MOD-N counter and a P-bit LFSR. The considered LFSR is supposed to generate 2^P-1 random patterns by selecting proper generating function which gives maximum length of random patterns. We have considered following three cases to determine the pre-defined pattern used for activating the trojan:

Case A: When the LFSR generated pattern is all 1s. The probability of getting triggered is $\frac{1}{(2^P-1)\cdot N}$.

Case B: When the LFSR generated patterns contain two consecutive 1s and rest of the bits are 0s (e.g.. 11000 for P=5). In this case, the triggering probability is $\frac{P-1}{(2^P-1)\cdot N}$.

Case C: When the LFSR generated pattern contains two dispersed 1s and rest of the bits are 0s (eg. 01010 for P=5). The probability of triggering the trojan is $\frac{(P-2)(P-1)}{2(2^P-1)\cdot N}$.

Table VI reports theoretical probability ratios of the trojan activation while considering P = 4, 6 and 8 and for MOD-3, MOD-7 and MOD-10 counters respectively. Each column of Table VI represents the probability of trojan activation for a specific MOD-N counter and P-bit LFSR for all considered cases mentioned earlier.

It can be observed from the table that the probability of activation of the trojan is rare in each of the cases of A, B, and C for all the three MOD-counters. Further, if the LFSR generated pattern considered for the trigger condition is all 1s (Case A), the trojan can be more stealthy (as shown in the first row of the table).

D. Experimental Observations

The simulation is performed using the Xilinx ISE Project Navigator 14.7, of Spartan6 family, with XC6SLX45 device and CSG324 package for implementing the proposed trojan model. The Golden circuit and the malicious circuit both are designed in Verilog. We evaluate the FPGA implementation using Windows 10 Pro 64-bit operating system with configuration: Intel core i7 processor, 8GB RAM and 800GB secondary storage. The synthesis tool used is XST and the simulator is ISim.

Let, $\overline{P} : 2^P - 1$, the number of states reachable by the LFSR (we have considered maximum-length LFSR for each considered P value).

x_1 : The starting state (pattern) for the LFSR.

x : The state (pattern) selected for the trojan condition.

y : The distance (in number of steps) between x_1 and x. If a LFSR starts from state x_1, then it will generate the pattern x after y clock cycle(s).

Following situations are observed during the simulation:

1) If $\overline{P}\%N = 0$
    a) y is not divisible by N, then the trigger condition will never satisfied. This situation can be examined for a 4-bit LFSR with generating function $x^4 + x^3 + 1$ and MOD-3 counter; if the starting pattern is '0101' and the trigger pattern is '1111'.
    b) y is divisible by N, then the trigger condition will occur only once. This situation can be examined for a 4-bit LFSR with generating function $x^4 + x^3 + 1$ and MOD-3 counter; if the starting pattern is '1011' and the trigger patter is '1111'.
2) If $\overline{P}\%N \neq 0$, the trigger condition will occur in at least one time in $(\overline{P} \cdot N)$ clock cycles.

Therefore, the triggering condition in the proposed trojan model is infrequent and highly depends on the initial pattern of the LFSR.

V. Conclusion

In this paper, we have explored an trojan model for the sequential circuits. The proposed approach tries to reach unreachable states in the FSM used for defining the trojan circuit. Theoretical computation reveals that the probability of trigger condition of the defined trojan is extremely rare and the trigger condition is purely random in nature. This ensures that the trojan is difficult to detect with the existing trojan detection algorithms. Simulation results also establish that the triggering condition is highly rare and depends on the architecture of the circuit elements used for modelling the trojan. The evaluation of the vulnerability, to break the popular encryption algorithms, of the trojan reported in this work can be the future research as Advanced Encryption Standard (AES) and other conventional encryption schemes use counters and FSMs.

References

[1] S. Bhunia, M. S. Hsiao, M. Banga and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1229-1247, Aug. 2014. doi: 10.1109/JPROC.2014.2334493

[2] X. Wang, M. Tehranipoor and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions,"in Hardware-Oriented Security and Trust, 2008. Host 2008. IEEE International Workshop on . IEEE, 2008, pp. 15-19.

TABLE VI: Theoretical probability of trojan activation

Case	P = 4			P = 6			P = 8		
	N = 3	N = 7	N = 10	N = 3	N = 7	N = 10	N = 3	N = 7	N = 10
A	0.0222	0.0095	0.0067	0.0052	0.0022	0.00158	0.0013	0.0005	0.0003
B	0.0667	0.0285	0.2000	0.0264	0.0113	0.00793	0.0091	0.0039	0.0027
C	0.0667	0.0285	0.2000	0.0529	0.0226	0.01587	0.0273	0.0117	0.0082

[3] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, ”Trojan detection using ic fingerprinting,” in IEEE Symposium on Security and Privacy 2007 (SP ’07) , May 2007, pp. 296-310.

[4] M. Tehranipoor and F. Koushanfar, ”A Survey of Hardware Trojan Taxonomy and Detection,” in IEEE Design & Test of Computers, vol. 27, no. 1, pp. 10-25, Jan.-Feb. 2010. doi: 10.1109/MDT.2010.

[5] Rajat Subhra Chakraborty, Seetharam Narasimhan and Swarup Bhunia, ”Hardware Trojan: Threats and Emerging Solutions,” in IEEE International High Level Design Validation and Test Workshop, 2009.

[6] Ziqi Zhou, Ujjwal Guin, V. D. Agrawal, ”Modeling and Test Generation for Combinational Hardware Trojans”, IEEE 36th VLSI Test Symposium (VTS), 2018.

[7] X. Wang, S. Narasimhan, A. Krishna, T. Mal-Sarkar and S. Bhunia, ”Sequential hardware Trojan: Side-channel aware design and placement,” 2011 IEEE 29th International Conference on Computer Design (ICCD), Amherst, MA, 2011, pp. 297-300. doi: 10.1109/ICCD.2011.6081413

[8] C. Sturton, M. Hicks, D. Wagner, and S. T. King, “Defeating UCI: Building stealthy and malicious hardware,” in Proc. IEEE Symp.Secur. Privacy, 2011, pp. 64–77.

[9] J. Zhang, F. Yuan, L. Wei, Z. Sun, and Q. Xu, “Veritrust: Verification for hardware trust,” in Proc. 50th Annu. Des. Autom. Conf., 2013, pp. 1–8.

[10] A. Waksman, M. Suozzo, and S. Sethumadhavan, “FANCI: Identification of stealthy malicious logic using boolean functional analysis,” in Proc. ACM SIGSAC Conf. Comput. & Commun. Secur., 2013, pp. 697–708.

[11] S. K. Haider, C. Jin, M. Ahmad, D. M. Shila, O. Khan and M. van Dijk, ”Advancing the State-of-the-Art in Hardware Trojans Detection,” in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 1, pp. 18-32, 1 Jan.-Feb. 2019. doi: 10.1109/TDSC.2017.2654352

[12] S. K. Haider, C. Jin, and M. van Dijk, “Advancing the state-of-the-art in hardware trojans design,” arXiv:1605.08413, 2016. [Online]. Available: http://arxiv.org/abs/1605.08413