The CSI Journal on Computer Science & Engineering

The CSI Journal on
Computer Science and Engineering
Vol. 14, No. 2, 2017
Pages 21-31
Regular Paper

Improving Effectiveness of Hardware Trojan Detection using Fault Injection

Najmeh Farajipour Ghohroud Shaahin Hessabi

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Abstract

Outsourcing the ICs for manufacturing introduces potential security threats such as hardware Trojans. Trojan inputs usually come from nets with rare activity. Therefore, increasing the probability of rare event activities in the circuit can increase the detection probability. We propose a method to increase the Trojan detection probability using fault injection. The proposed method has no hardware overhead, and has the potential for detection of several types of Trojans that are triggered by rare events. It increases the probability of Trojan activation by up to three times, and achieves Trojan detection rate above 95% with false alarm rate below 3%.

Keywords: Controllability, Fault Injection, Rare Events, Transition Probability, Trojan Detection.

1. Introduction

Most hardware manufacturers outsource the fabrication of their integrated circuits (ICs) to third party foundries in order to reduce the cost of silicon chip fabrication [1]. This increases the vulnerability to malicious activities [2]. Third party foundries may modify the circuit's design or its physical parameters. These modifications are known as Hardware Trojan Horses (HTH). An adversary can insert a Trojan in the design to disable and/or destroy a system, or leak information to the adversary.

HTHs are classified into two categories: parametric and functional. Parametric HTHs modify the characteristics of existing hardware, such as delay, power consumption, etc., while functional HTHs correspond to addition or removal of gates and transistors in the design, which may change its functionality [1]. In this paper, we focus on functional HTHs because they usually have more complicated and damaging behaviour.

HTHs are normally triggered by some internal or external events, or a sequence of such events, to become operative. A smart adversary will try to hide such modification of IC's functional behaviour in a way that makes it very difficult to detect with conventional post–manufacturing test [3]. Therefore, the adversary would ensure that the modified behaviour is triggered under very rare conditions, which are unlikely to occur during test but can arise during the long period of field operation [4].

Trojans are silent most of the time and have small sizes relative to the entire design, with limited contribution to the design characteristics. Therefore, they are most likely connected to nets with low controllability and/or observability [5, 6]. Several Trojan detection methods and design for trust methods [7, 8] have been proposed. Trojan detection methods using transient power analysis [9, 10] require patterns that increase Trojan activity while keeping circuit activity low, to increase Trojan contribution to the circuit power consumption.

Methods based on delay analysis [11, 12, 13] require patterns that generate transitions on Trojan inputs to magnify the delay impact of Trojan on the circuit delay characteristic. In this paper, we propose a method to increase the probability of generating a transition in functional Trojan circuits and improve the effectiveness of existing methods. In contrast to previous work, such as dummy flip-flop insertion [14] or MERO [15], our method has no hardware overhead

and can be applied after chip fabrication. We use physical fault injection to trigger rare events.

1.1. Previous Work

Several Trojan detection methods have been proposed. These methods are classified into three categories: side-channel analysis methods, logic testing methods, and monitoring methods. In the first category, side-channel parameters of the circuit, such as power and delay, are analyzed and the presence of Trojan is examined based on the variations in these parameters [16]. The side-channel parameter can be power consumption [1, 17, 18, 19], current [12] or delay [11, 20]. These methods are effective when the Trojan is large enough to affect these parameters. Measurement noise, process variation and environmental changes may mask the Trojan effects on side-channel parameters.

The objective of the methods in the second category is to activate the Trojan completely [21, 22, 23, 24]. In the case of functional Trojans, when the Trojan is activated, its malicious effect can be observed on the design output. The main problem is the long time for full activation of the Trojan. Since a smart adversary makes the trigger condition of the Trojan very rare, the probability of Trojan activation in a short time is very low. Several design for trust methods have been proposed to produce dummy transitions on nets with rare events. Dummy flip-flop insertion [21] and voltage inversion [22] are two examples of these methods, which have large hardware overhead.

Monitoring systems are proposed to prevent Trojans from damaging the circuit behaviour. An example is the DEFENSE system which is embedded in the functional design of logic circuits to implement real-time security monitors [25].

Our proposed method can be used to enhance side-channel based Trojan detection methods. Since increasing the probability of rare events can cause all or some Trojan inputs to trigger, it can fully or partially activate the Trojan. Even if it is partially activated, it can be detected using side-channel methods. We increase the probability of rare events by laser-induced fault injection. The four main advantages of our method over previous methods are having no hardware overhead, non-necessity of golden model, detection of Trojan location, and generality. In the case of using golden model in our method, the Trojan detection accuracy will be increased.

2. Proposed Method

The adversary tries to insert Trojan in circuit nodes whose inputs trigger rarely and are hard to control [26]. Therefore, increasing transition probability of these rare nets is one way to increase the Trojan activation probability. Using the method presented in this paper, we attempt to increase the transition probability of these nets by fault injection. We theoretically show that when we inject faults on each net of the circuit, the probability of rare events increases considerably. We apply comprehensive gate-level fault injection on some ISCAS benchmarks [27]. Figure 1 shows the result. We report the percentage of increase in probability of rarest event after fault injection. Since the probabilities of rarest events in different benchmarks are very different in range, we show the percentage of probability increase for the sake of clarity in presentation. Absolute probability values of rarest events for these benchmarks are reported in table 1 For example, for C17 circuit the probability of rarest logic 0 before fault injection is 0.25 and using fault injection increases this probability to 0.29, which shows a 16% increase for this event. The rarest event probability can be increased by fault injection by up to 2.8 times (for C1355). For larger circuits the method is more effective.

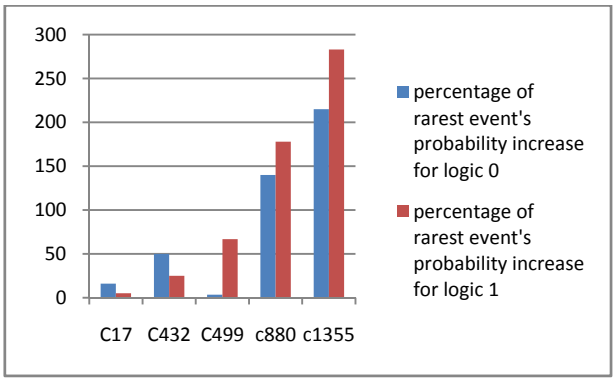


Figure 1. Rare events' probability increase by fault injection on some ISCAS benchmarks

As a second step, we should choose a physical fault injection method which is precise enough to inject a fault in a specific net of the design. There are several physical fault injection methods, such as using heavy ions, protons, neutrons and laser beam. Laser-induced fault injection has some advantages over others, as follows: 1) it can more accurately specify the fault injection location, 2) LET (Linear Energy Transfer) is easily controllable, and 3) implementation costs are affordable [28]. As a result, we chose laser-induced fault injection as our physical fault injection method.

After choosing the fault injection technique, we should model laser-induced fault injection to evaluate the effectiveness of our proposed method. In [29, 30], authors model the effect of laser irradiation on NMOS and PMOS transistors. They use a pulsed laser at 1064nm wavelength to conduct the Photoelectric Laser Stimulation experiments. The obtained measurements were used to validate and tune the model. As shown in their papers, laser irradiation causes some parasitic currents to flow. They model the laser irradiation effect using some dependent current sources, located between well and substrate, well and source, and well and drain. Employing their models, we can model the laser irradiation effect on the output value of the transistors and logic gates consisting of these transistors.

We modelled laser irradiation on main logic gates (AND, OR, NAND, NOR, NOT, XOR, XNOR) and flip-flops (D-FF, T-FF, JK-FF) using SPICE, and extracted the fault model. The fault models are bit set and bit reset faults, but bit flip fault is not generated using laser irradiation. Since the laser spot size is much smaller than the transistor size (as shown in figures 4 to 7), in each laser irradiation step, only a single fault is injected. Therefore, there is no need to consider multiple fault injection. Utilizing our laser-dependent gate models, we can simulate laser-induced fault injection.

To apply Trojan detection method using laser-induced fault injection, the first step is to detect corners of the wafer

using x-ray image of the chip, and limit the range of laser irradiation to the wafer area. Then, we place the chip under a 2D nanopositioner like N470 from PI Corporation. The laser that will be used is a 100mW laser with the wavelength of 1064nm, 50ns pulse duration and spot sizes from 0.1 to 1 micrometer. The absorption of the silicon at 1064nm wavelength is weak [31]. Our method is based on backside laser illumination. Backside laser illumination is also used in [30] [29]. The precision of the above mentioned nanopositioner is 20 nm. Therefore, it can be used for recent technologies. Since we use 90 nm technology in this paper, this precision is enough for us.

Our approach to detect hardware Trojans in chips has two stages. First, we select a chip and pose it on the test board. We apply the patterns to the inputs in the absence of laser irradiation and record side channel parameters such as power consumption. After applying all predefined input patterns and extracting the side channel trace, the second stage begins. The chip is placed under laser beam and the all patterns which had been applied in the first stage are applied to its inputs.

After applying each pattern, the laser beam scans the chip area completely and the side channel trace is recorded. By comparing the side channel trace before and after irradiation, and finding big differences, we can locate the possible inserted Trojan. Using the coordinates of the laser beam on the chip, we can detect the location of the Trojan. To ensure that this unusual difference in side channel trace is the result of the Trojan behavior, we repeat the laser irradiation locally on the suspected part of the chip. If this unusual event is observed again, the Trojan presence is confirmed.

Two important issues exist in our approach: generation of input patterns that are applied to the circuit, and the process of unusual events detection in side channel trace. These two issues are discussed in the following subsections.

2.1. Input Pattern Generation

It is obvious that we cannot apply all possible input patterns. For circuits with high number of inputs, this process will be very time consuming and practically impossible. Therefore, we should apply a limited number of patterns. Several research works are concentrated on generation of patterns which can trigger almost all parts of the circuit and several methods are proposed to solve this problem.

Although all parts of the circuit cannot be triggered by applying the random patterns, this type of random pattern generation is more efficient than algorithmic methods. Hence, in this paper we randomly generated input patterns. We create a batch of input patterns for a chip and use it in the later stages. The process of random pattern generation is as follows:

a. Using the random generation function in C library, and CPU time as its seed, we generate random integers and convert them to 32-bit binary vectors.

b. Based on the number of input bits (N) of the circuit, we generate N-bit random binary vectors.

c. Repeat steps (a) and (b) until a predefined number of N-bit patterns is produced. Then these patterns are saved in a file and are used in every simulation phase.

Since this method is scalable, it is easily applicable for industrial-size circuits.

2.2. Detection of Unusual Events in Side Channel Trace

In our Trojan detection approach, we should compare the side-channel trace of the circuit before fault injection with its side channel trace after fault injection. In this paper, we use power traces as side-channel trace to achieve the goal of Trojan detection. It is necessary to do this comparison automatically. Therefore, the process of detecting big differences between two power traces should be formulated. This formulation is shown in (1).

$$D(t_{i,j}) = \left(\frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{AFI}(t)dt}{\Delta t} - \mu_{P_{AFI}} \right) - \sqrt{\sum \frac{(\frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{BFI}(t)dt}{\Delta t} - \mu_{P_{BFI}})^2}{\frac{T}{\Delta t}}} \quad (1)$$

Where $P_{BFI}(t)$ and $\mu_{P_{BFI}}$ are the power consumption function and the average power of the circuit which is not exposed to laser irradiation. $P_{AFI}(t)$ and $\mu_{P_{AFI}}$ are the power consumption function and the average power of the circuit during the laser-induced fault injection process. Δt is the time interval in which the comparison is done, and T is the total test time. $[t_{i,j}, t_{i,j}+\Delta t]$ is the time interval in which the laser beam is located on (i, j) coordination of the chip. For each interval, $D(t_{i,j})$ is calculated to show the difference between the two power traces. The first term calculate the difference between average power consumption in time interval $[t_{i,j}, t_{i,j}+\Delta t]$ and the average power consumption during the process, for laser-induced fault injection process. The second term calculate the standard deviation of power consumption before laser-induced fault injection. If $D(t_{i,j})$ exceeds a threshold value, it means that the big difference is detected between two power traces(before and after fault injection), and the possibility of Trojan insertion on that point is high. The accuracy of this Trojan detection approach is highly dependent on the threshold value. In Section3.2, the process of threshold value determination is described.

2.3. Types of Trojan which are Detectable Using Our Approach

Several methods of classifying hardware Trojans based on their features have been proposed. In [32], the authors propose a simple classification of Trojans – combinational (whose activation depends on the occurrence of a special condition at some internal nodes of the circuit) and sequential (whose activation depends on the occurrence of a certain sequence of logic values at internal nodes). In [5], the authors classify Trojans based on three characteristics: physical, activation and action. In this section, we follow the Trojan taxonomy proposed in [4], where the Trojans are classified based on their trigger and payload mechanisms, as shown in figure 2.

All the functional Trojans triggered digitally can be activated by fault injection, especially those that are activated by a single command bit. In addition, during the laser-induced fault injection, the Trojan may be partially activated many times. Therefore, by measuring the side channel parameters such as power consumption, we can observe the Trojan effect on the circuit. In the case of Trojans with sequential trigger circuit, this phenomenon occurs with lower

rate, because the Trojan will be activated if a rare sequence of events occurs. We can overcome this problem by utilizing the predesigned scan chains which are used to test sequential circuits. Applying random patterns to feed scan chain flip-flops, we can fully or partially trigger the Trojan. However, each sequential Trojan has a combinational part to control the sequence occurrence. Injecting faults in this combinational part results in activation of the Trojan.

Trojans whose trigger circuit is analog, fault injection method can be effective. Even for Trojans which are triggered when on-chip sensors report a predefined value, there is a controller circuit which controls the sensor value and compares it with a predefine value or a threshold. Fault injection in this controller circuit can fully or partially activate the Trojan. However, when the payload circuit is analog, it may be very difficult to confirm the Trojan existence using side channel parameters. In this paper, we investigate the existence of Trojans whose trigger and payload circuits are digital circuits. In the next section, we evaluate our proposed method on circuits with digital Trojans (combinational or sequential).

2.4. Threshold Calibration with a Golden Model

Although our method can be used without using golden models, there are some important points which may affect the accuracy of our Trojan detection method. Practice of fault injection reveals that exposing an IC to laser irradiation induces photocurrents. Even a fault induced in a HT-free part of an IC, it will propagate and induce a change in the circuit's side channel trace because it will change its logical activity. Therefore, it is highly possible to detect these side channel changes as Trojan existence by mistake (increasing false alarm rate).

To eliminate the effect of these phenomena on Trojan detection method, we can use some golden models to calibrate the threshold values used for Trojan detection. We can record the side channel profile of these golden models before and after fault injection. The difference between these two trace can give us some information about the threshold value which should be used for Trojan detection. Equation (2) shows how we can calculate the threshold value when we use power trace as the circuit side channel trace.

$$GMD_k(t_{i,j}) = \left(\frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{GM_kAFI}(t)dt}{\Delta t} - \frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{GM_kBFI}(t)dt}{\Delta t} \right) \quad (2)$$

$$\mu_{GMD} = \frac{\left(\sum_{k=1}^{N} \mu_{GMD_k}\right)}{N} \quad (3)$$

$$TID(t_{i,j}) = \left(\frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{TIAFI}(t)dt}{\Delta t} - \frac{\int_{t_{i,j}}^{t_{i,j}+\Delta t} P_{TIBFI}(t)dt}{\Delta t} \right) \quad (4)$$

Where $P_{GM_kBFI}(t)$ and $P_{TIBFI}(t)$ are power consumption functions of k'th golden model circuit and Trojan infected circuit which are not exposed to laser irradiation. $P_{GM_kAFI}(t)$ and $P_{TIAFI}(t)$ are the power consumption function of k'th golden model circuit and Trojan infected circuit during the laser-induced fault injection process. Δt, $[t_{i,j}, t_{i,j}+\Delta t]$ and (i,j) have the same meaning as used in (1). $GMD_k(t_{i,j})$ is calculated to show the difference between the two power traces for k'th golden model. $TID(t_{i,j})$ is calculated to show the difference between the two power traces for Trojan infected circuit. μ_{GMD_k} is the average value of absolute values of $GMD_k(t_{i,j})$ over time and μ_{GMD} is the average value of μ_{GMD_k} values for k different golden models. We use μ_{GMD} and its multiples as threshold value. If the absolute value of $TID(t_{i,j})$ exceeds a threshold value, it means that the big difference is detected between two power traces (before and after fault injection), and the possibility of Trojan insertion on that point is high. In this way, we eliminate the effect of power changes caused by laser-induced fault injection. Details are described in Section 3.2.

2.5. Main Advantages of the Proposed Trojan Detection Approach

Our approach has three advantages over other Trojan detection methods. First, it does not impose any hardware overhead. Second, several types of Trojans which are trigged by rare events (sequential, combinational, or hybrid triggered Trojans) can be detected.

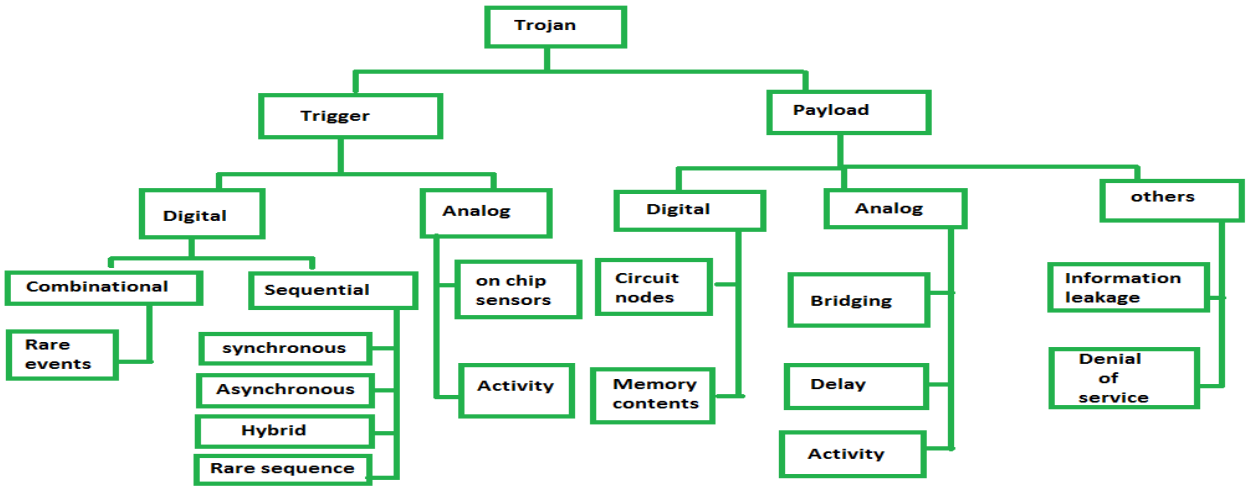


Figure 2. Trojan classification [4]

Third, using laser-induced fault injection, all nodes of the design are accessible, and there is no need for partitioning and isolation. Nevertheless, it is necessary to demonstrate the effectiveness of the proposed method in practice. Therefore, we simulated laser-induced fault injection on the layout of sample designs, and analyzed the results.

3. Experimental Results

3.1. Laser-Induced Fault Injection on a Sample Circuit

In this section, we demonstrate how we can inject faults using laser pulses to trigger rare nets. We designed a sample circuit layout in 90 nm technology, and applied our laser-dependent gates model to its logic gates and flip-flops. Utilizing laser sensitive transistor models, we simulate laser irradiation to the circuit at transistor level using SPICE program. The gate level schematic of the sample design is shown in figure 3. This is a very small design consisting of combinational and sequential elements, but the method can be applied for complicated designs, as well.

The width of this layout is 37 um and its height is 7 um. We simulated scanning of this layout using laser beam with a step size of 0.2 um. The laser pulse width is 50 ns and its duration on each point is 400 ns. The laser power is 100 mW and its spot size is 1 um.

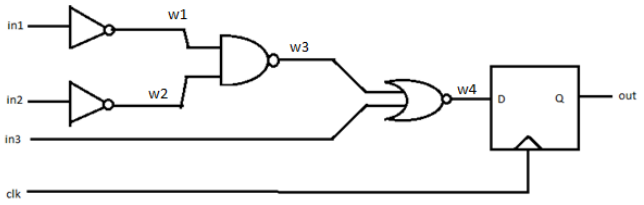


Figure 3. A sample gate level design

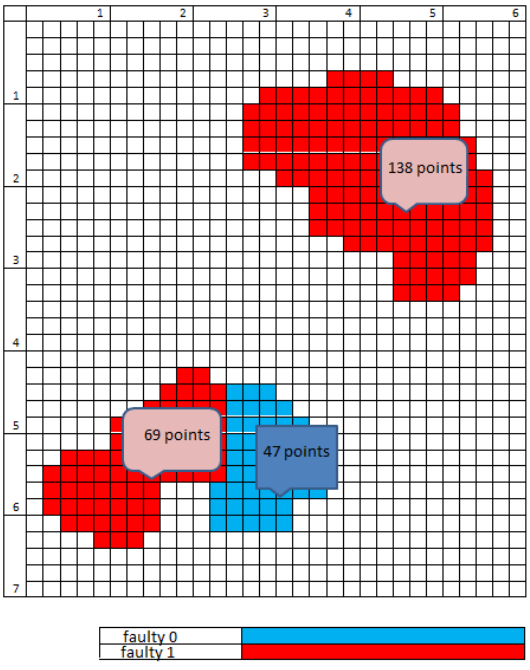


Figure 4. The map of laser induced faults for D flip-flop

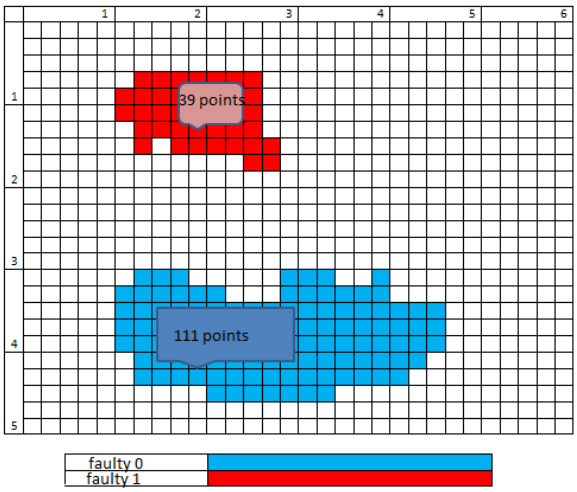


Figure 5. The map of laser induced faults for NOR gate

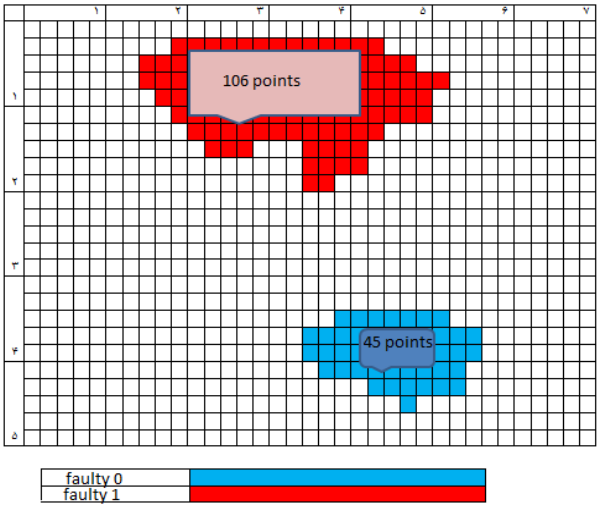


Figure 6. The map of laser induced faults for NAND gate

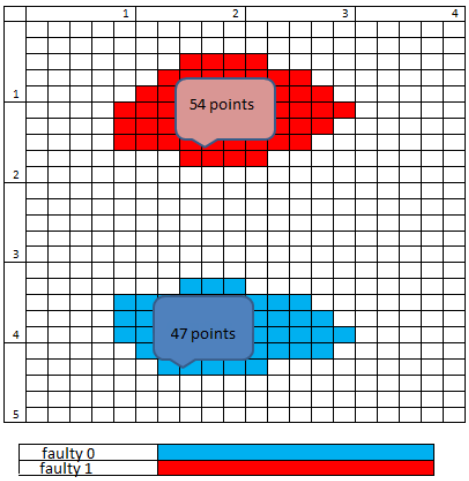


Figure 7.The map of laser induced faults for NOT gate

We extracted the map of laser-induced faults for each gate and flip-flop separately. For example, for D flip-flop, the size of the layout is 6 um in width and 7 um in height. Since the laser irradiation step size is 0.2 um, the total irradiation points in this layout are (6*5)*(7*5) which is

equal to 1050 points, as is shown in figure 4. When the laser irradiation causes the output of the gate to become 1 while the correct output is 0, the irradiation point is marked as red, and when the laser irradiation causes the output of the gate to become 0 when the correct output is 1, the irradiation point is marked as blue. Figures 4 to 7 show the map of laser-induced faults for D flip-flop, NOR gate, NAND gate and NOT gate, respectively.

We applied all possible input patterns to the circuit, and simulated laser scanning for each pattern. Since the circuit has 3 inputs, we should irradiate the laser 29400 times, which takes about 11.5 ms. Using the above maps, we found the number of times the faulty output is generated on each gate or flip-flop to be 3644. Among these internal faults, 2116 cases are manifested on circuits' outputs as an error. Therefore, using laser beam we can inject all types of stuck-at faults on all nets of the design several times. The rarest event in the sample circuit is logic 1 on the circuit's output. Using laser-induced fault injection, we can deliberately set it to logic 1, for 1137 more times. The probability of this event before fault injection is 0.125, and is increased to 0.1637 (↑31%) using fault injection. Since the circuit is very small and the rarest event's probability is relatively high, the effect of this method is not clearly shown in this example. For larger circuits, this method is more effective. Table 1 shows the probability of rarest 0 and 1 before and after fault injection for some ISCAS benchmarks. As shown in the table, for larger circuits in which the rarest event's probability is relatively low, the fault injection method is more effective. Tables 2 and 3 show the number of internal faults injected by laser irradiation, and the fault injection duration for some ISCAS benchmarks using laser resolution steps of 0.2 um and 0.5 um. For larger circuits, the duration of laser-induced fault injection will be increased. However, it is not worrying because unlike the test process where all tests should be applied for each chip, the Trojan detection process should be performed only for a small sample of produced chips. Therefore, the long process duration is acceptable. For increasing the Trojan detection speed, we can use a larger laser resolution step or apply laser fault injection for several chips in parallel.

Table 1. Probability of rarest 0 & 1 before and after fault injection for ISCAS benchmarks with 0.5 um laser resolution step

ISCAS circuit	Before fault injection		After fault injection	
	The probability of rarest logic 0	The probability of rarest logic 1	The probability of rarest logic 0	The probability of rarest logic 1
C17	0.25	0.38	0.29(↑16%)	0.4(↑5.26)
C432	0.12	0.08	0.18(↑50%)	0.11(↑37.5%)
C499	0.497	0.003	0.514(↑3.4%)	0.005(↑66.7%)
C880	0.001	0.0009	0.0024(↑140%)	0.0025(↑177.8%)
C1355	0.00057	0.00014	0.00179(↑214%)	0.0004172(↑198%)

Table 2. Simulation results for laser-induced fault injection for ISCAS benchmarks with 0.2 um laser resolution step

ISCAS circuit		Number of laser irradiation points	Number of internal laser-induced faults	Number of uncollapsed faults in the circuit * number of input patterns	Estimated time for laser fault injection
C17	All input patterns	360000	45720 (12.7%)	12*32=384 (1/119 of induced faults)	144 ms
C432	(all functionally allowed input patterns)	9493000	170874 (1.8%)	320*40=12800 (1/13 of induced faults)	3.8 s
	Random input pattern using 0.02% of all patterns	3261769880875	345747607372 (10.6%)	320*13743895=4398046400 (1/78 of induced faults)	15 days
C499	Random input pattern using 0.02% of all patterns	359144973300	28013307917 (7.8%)	404*858993=347033172 (1/81 of induced faults)	40 hours (1.7 days)
C880	Random input pattern using 0.02% of all patterns	202574816870	8528399790 (4.21%)	766*348966=267307956 (1/32 of induced faults)	22 hours
C1355	Random input pattern using 0.02% of all patterns	817332276428	41683946097 (5.1%)	1092*858993=938020356 (1/44 of induced faults)	90 hours (3.75 days)

Table 3. Simulation results for laser-induced fault injection for ISCAS benchmarks with 0.5 um laser resolution step

ISCAS circuit		Number of laser irradiation points	Number of internal laser-induced faults	Number of uncollapsed faults in the circuit * number of input patterns	Estimated time for laser fault injection
C17	All input patterns	57600	5644 (9.8%)	12*32=384 (1/15 of induced faults)	23 ms
C432	(all functionally allowed input patterns)	1518880	24302 (1.6%)	320*40=12800 (1/2 of induced faults)	607 ms
	Random input pattern using 0.02% of all patterns	521883180940	44881953560 (8.6%)	320*13743895=4398046400 (1/10 of induced faults)	57 hours
C499	Random input pattern using 0.02% of all patterns	57463195728	3103012569 (5.4%)	404*858993=347033172 (1/9 of induced faults)	6.38 hours
C880	Random input pattern using 0.02% of all patterns	32411970699	1069595033 (3.3%)	766*348966=267307956 (1/4 of induced faults)	3.6 hours
C1355	Random input pattern using 0.02% of all patterns	130773164228	6146338718 (4.7%)	1092*858993=938020356 (1/6 of induced faults)	14.5 hours

We applied laser-induced fault injection for combinational circuits. However, using scan chain, which is implemented for test purposes, every sequential circuit can be treated as a combinational circuit.

Simulations show that the main idea behind our Trojan detection method is correct in practice and in layout level. Therefore, we can apply laser-induced fault injection on the manufactured design, and use logic testing or side-channel analysis.

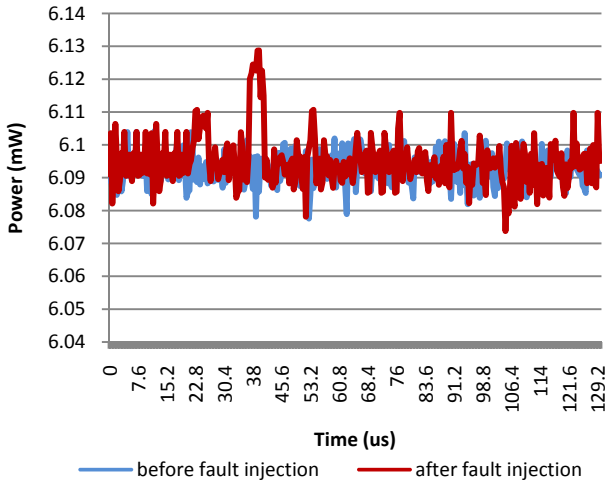
3.2. Trojan Detection Procedure without Using Golden Models

In this section, we evaluate our hardware Trojan detection and diagnosis approach on the ISCAS85 benchmarks. Three versions of the same layouts of some ISCAS'85 benchmarks were generated. Three combinational comparator Trojan circuits with 3, 5 and 10 gates named Trj1, Trj2, and Trj3 were designed and inserted in these benchmarks. The attributes of these Trojans are summarized in table 4.

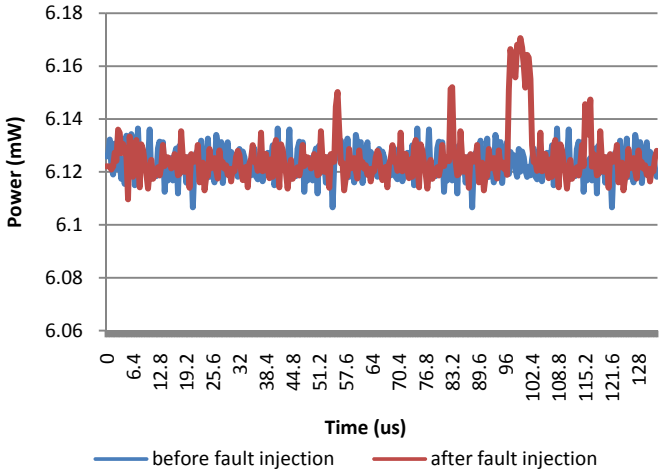
Table 4. Trojan attributes

Trojan	# of gates	Area overhead (relative to C432 circuit)
Trj1	3	3.4%
Trj2	5	4.7%
Trj3	10	9.6%

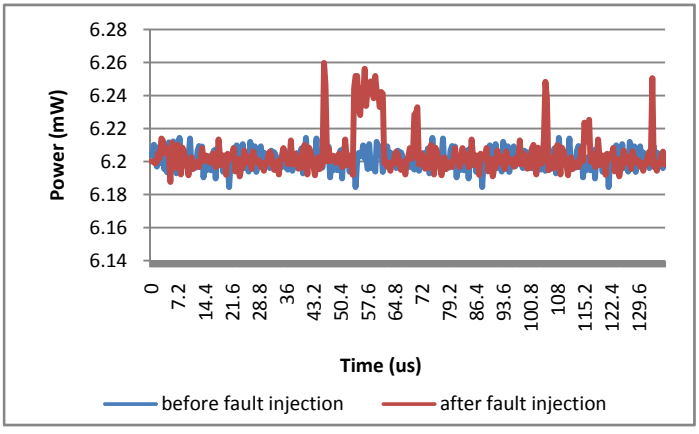
The original designs were synthesized using Synopsys Design Compiler™ [a] with 90-nm technology [33]. Using random pattern generation technique, we create a batch of input patterns for each benchmark. We apply these input patterns to all three versions of the benchmarks (which consist of three different Trojans Trj1, Trj2 and Trj3) two times. In the first attempt, the circuit is not exposed to laser-induced fault injection, while in the second one, after applying each pattern to the inputs, the laser beam scans the circuit and several faults are injected in different points of the circuit. The duration of applying each pattern in these two flows is the same. In each flow, the power trace is extracted and the big differences in the traces are found to locate the hardware Trojan in the circuit. Figure 8 shows power traces of these two flows for C432 benchmark containing three different Trojans.



(a)



(b)



(c)

Figure 8. Power trace of C432 benchmark in presence of three different Trojans. a) Trj1, b) Trj2 & c) Trj3

As observed in the figures, fault injection in some points of the circuit increases the power consumption unusually. These points are the possible Trojans locations. As described in Section 2.2, Equation (1) calculates the difference between two traces. If $D(t_{i,j})$ exceeds the predefined threshold value, the big difference is found and the possible inserted Trojan is detected. We set the threshold value based on the standard deviation value of power trace of the circuit before fault injection. If the threshold value is very low, some usual events in the trace may be reported as unusual, and the false positive rate will increase. On the other hand, if the threshold is very high, unusual events may be reported as usual, which increase the false negative rate. Figure 9 shows the effect of threshold value selection on false positive and false negative rates of the proposed method, respectively.

Since true negative rates and true positive rates are complements of false positive rates and false negative rates, they are not shown in the figure. As observed, for threshold value of 4σ, the true negative and true positive rates are above 97%, and false positive and false negative rates are below 3%. Therefore, this value is a candidate value to be used in the procedure of Trojan detection. There is a tradeoff between false positive and false negative rates. As shown in figure 9(a) and 9(b), increasing the threshold value increases the false negative rate and decreases the false positive rate. From a security perspective, false negative rate is more important than false positive rate. Therefore, we may choose a threshold value which results in minimum false negative rate.

To eliminate the effect of currents induced by the laser irradiation and fault injection, we repeat the process of Trojan detection using threshold value calculated by (3). If the absolute value of $\mathrm{TID}(t_{i,j})$, which is calculated using (4), exceeds the threshold value the big difference is found and the possible inserted Trojan is detected.

We use μ_{GMD}, $2\mu_{GMD}$, $3\mu_{GMD}$ and $4\mu_{GMD}$ as threshold values and do the Trojan detection process for each of them. Figure 10 shows the effect of threshold value selection on false positive and false negative rates of the proposed method using golden models, respectively. Since true negative rates and true positive rates are complements of false positive rates and false negative rates, they are not shown in the figure.

As observed, for threshold value of 2μ, the true negative and true positive rates are above 97%, and false positive and false negative rates are below 3%. Therefore, this value is a candidate value to be used in the procedure of Trojan detection. Since for threshold value of 2μ, false negative rate is zero, it is a good candidate in a security perspective. Using golden models we remove the effect of laser induced currents on power trace and achieve more accuracy. To examine the outcome of the proposed method for Trojan-free circuit, its power trace is extracted before and after fault injection which is shown in figure 11. Figure 12 depicts the difference between these two traces. Now this difference is compared with threshold levels. As it can be observed, only at three points the values are above 2μ which is selected as the appropriate threshold level to be used in Trojan detection process. This is despite the fact that for the smallest Trojan with only three gates, the number of spikes is more than 15. Also, as observed in figures 8(a), 8(b) and 8(c), in Trojan infected circuits, the duration of spikes which are produced after fault injection in Trojan locations, is more than 4us. However, duration of spikes in the trace of the Trojan free circuit is less than 500ns. Therefore, if the difference between two power traces of a circuit contains short duration spikes, we can ignore these short spikes and mark such circuits as Trojan free.

4. Conclusions and Future Work

This paper reports an analysis of the laser-induced sensitized nodes of different CMOS gates. It is shown that laser-induced fault injection can increase the transition probability of rare events without hardware overhead. This phenomenon can improve the probability of Trojan activation, and can improve the effectiveness of Trojan detection methods. Therefore, using this accurate fault injection method, we can improve the Trojan coverage of current Trojan detection methods. The results show that using laser-induced fault injection, the probability of Trojan activation can be increased up to three times. We achieved the Trojan detection rate above 95% with 3% false alarm rate.

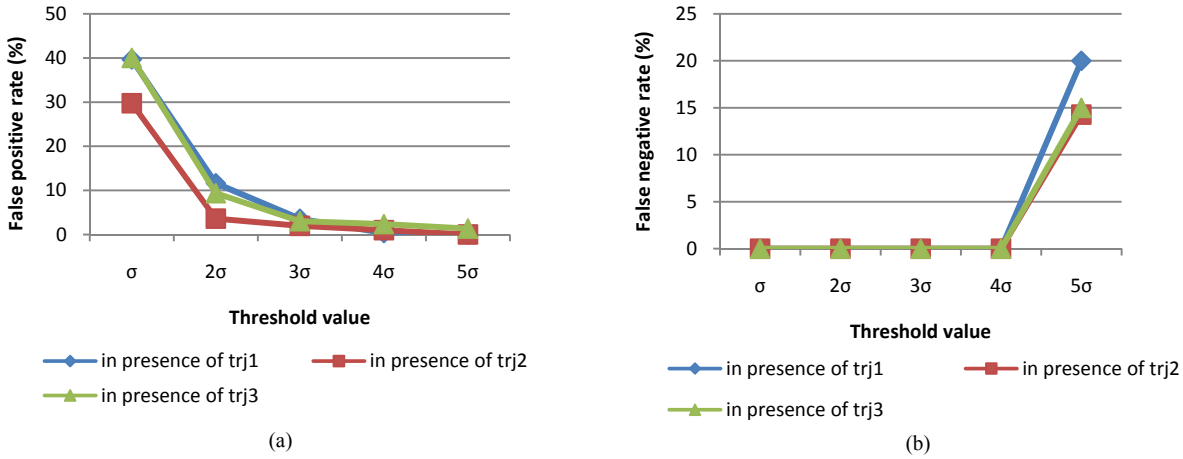


Figure 9. False detection rates of Trojan detection method applied on C432 benchmark in presence of three different Trojans. a) False positive rate and b) False negative rate

In spite of considerable advantages of this method, the full scanning of the chip by laser beam for each input pattern is time consuming for large designs. Proposing methods for shortening the duration of Trojan detection phase can be studied in future work. Laser irradiation on random locations of the chip reduces the duration of Trojan detection phase. Therefore, investigation of the effectiveness of this type of laser-induced fault injection on Trojan detection rate can be investigated in future work. Although in this work we focus on functional Trojans, targeting parametric Trojans will be part of our future work.

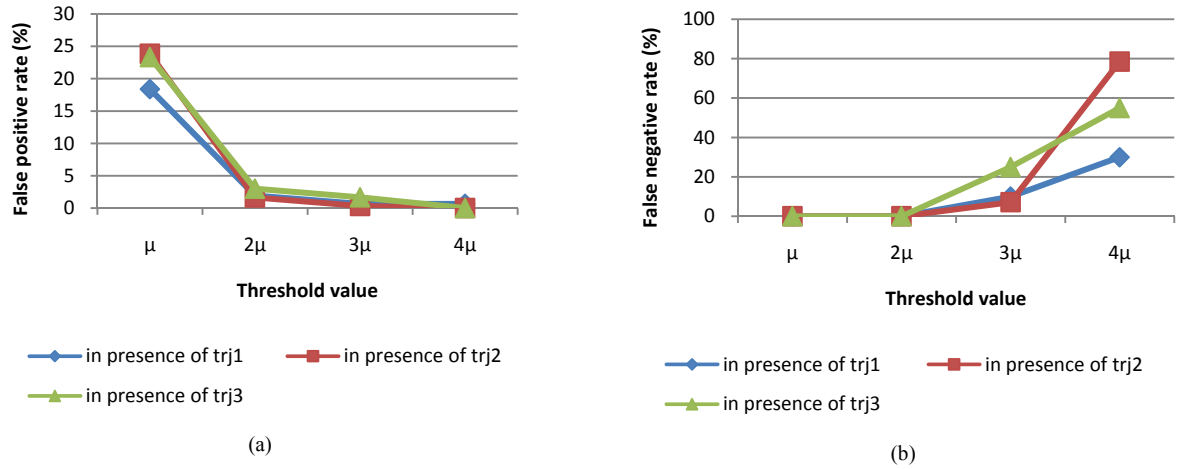


Figure 10. True detection rates and false detection rates of Trojan detection method using golden models applied on C432 benchmark in presence of three different Trojans. a) False positive rate & b) False negative rate

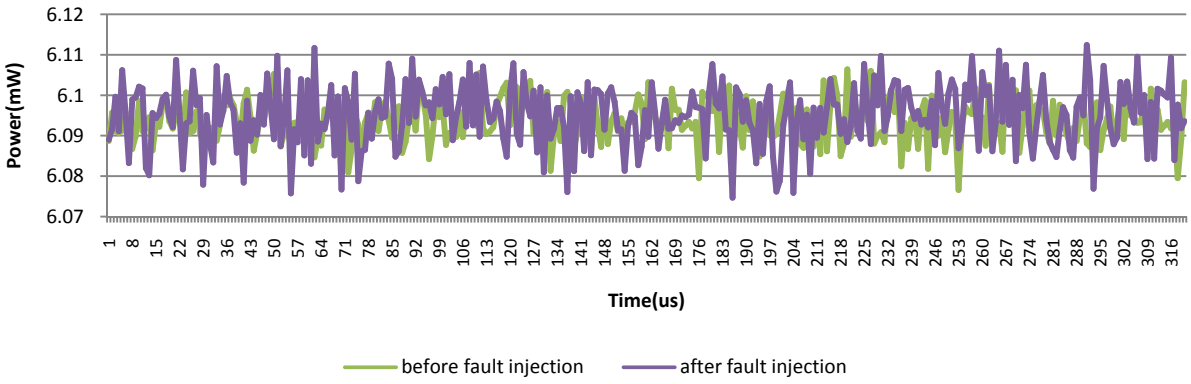


Figure 11. Power trace of C432 benchmark for Trojan free circuit

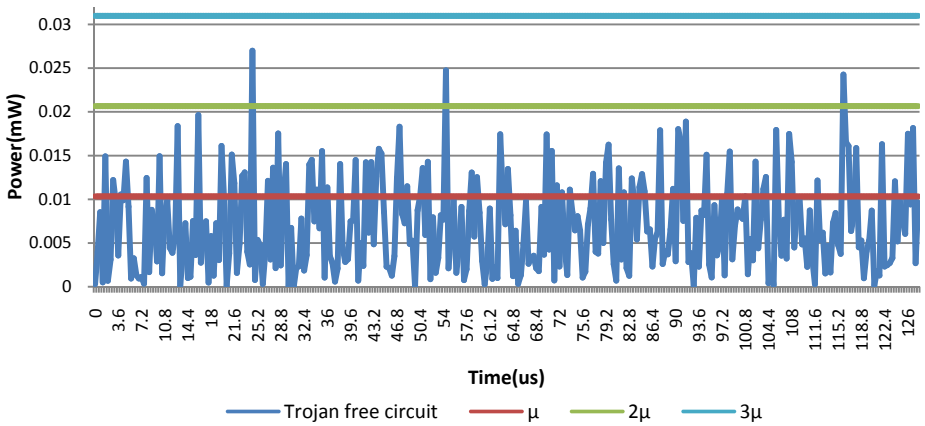


Figure 12. Difference of two power traces of C432 benchmark for Trojan free circuit

Reference

[1] M. Tehranipoor, and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design and Test,* vol. PP, no. 99, pp. 10-25, 2013.

[2] P. Subramanyan, and et al., "Reverse Engineering Digital Circuits Using Structural and Functional Analyses," *IEEE Transactions on Emerging Topics in Computing,* vol. 2, no. 1, pp. 63-80, 2014.

[3] "DARPA BAA06-40: TRUST for Integrated Circuits," Defense Advanced Research Projects Agency, [Online]. Available: https://www.fbo.gov/index?s=opportunity&mode=form&id=db4ea611cad3764814b6937fcab2180a&tab=core&_cview=. [Accessed 18 08 2015].

[4] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme," in *Design, Automation and Test in Europe (DATE'08)*, Munich, 2008.

[5] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, Anaheim, CA, 2008.

[6] M. Banga, and M. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans," in *22nd International Conference on VLSI Design*, New Delh, 2009.

[7] L. Bossuet, X. Thuy NGO, Z. Cherif, and V. Fischer, "A PUF Based on a Transient Effect Ring Oscillator and Insensitive to Locking Phenomenon," *IEEE Transactions on Emerging Topics in Computing,* vol. 2, no. 1, pp. 30-36, 2014.

[8] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and Reverse-Engineering Resilient PUF Authentication and Key-Exchange by Substring Matching," *IEEE Transactions on Emerging Topics in Computing,* vol. 2, no. 1, pp. 37-49, 2014.

[9] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in *IEEE Symposium on Security and Privacy (SP'07)*, Berkeley, CA , 2007.

[10] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'08)*, San Jose, CA, 2008.

[11] J. Li, and J. Lach, "At-speed Delay Characterization for IC Authentication and Trojan Horse Detection," in *," in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, Anaheim, CA, 2008.

[12] Y. Jin, and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprint," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, Anaheim, CA, 2008.

[13] X. Zhang, A. Ferraiuolo, and M. Tehranipoor, "Detection of Trojans Using a Combined Ring Oscillator Network and Off-Chip Transient Power Analysis," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 9, no. 3, 2013.

[14] H. Salmani, M. Tehranipoor, and J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'09)*, Francisco, CA, 2009.

[15] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: A Statistical Approach for Hardware Trojan Detection.," in *Cryptographic Hardware and Embedded Systems (CHES 2009)*, Springer, 2009, pp. 396-410.

[16] Nisha Jacob, Dominik Merli, Johann Heyszl, and et al., "Hardware Trojans: Current Challenges and Approaches," *IET Computers & Digital Techniques, Special Issue on Hardware Security,* vol. 8, no. 6, pp. 264-273, 2014.

[17] R. Rad, J. Plusquellic, and M. Tehranipoor, "A Sensitivity Analysis of Power Signal Methods for Detecting Hardware Trojans Under Real Process and Environmental Conditions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 12, pp. 1735-1744, 2010.

[18] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware Trojan Detection and Isolation using Current Integration and Localized Current Analysis," in *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFTVS08)*, Boston, MA, 2008.

[19] C. Marchand, and J. Francq, "Low-level Implementation and Side-Channel Detection," *IET Computers & Digital Techniques, Special Issue on Hardware Security,* vol. 8, no. 6, pp. 246-255, 2014.

[20] S. Jha, and S. K. Jha, "Randomization Based Probabilistic Approach to Detect Trojan Circuits," in *the 11th IEEE High Assurance Systems Engineering Symposium*, Nanjing, 2008.

[21] M. Banga, and M. Hsiao, "VITAMIN: Voltage Inversion Technique to Ascertain Malicious Insertions in ICs," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'09)*, Francisco, CA, 2009.

[22] R. S. Chakraborty, and S. Bhunia, "Security against hardware Trojan Through a Novel Application of Design Obfuscation," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'09)*, San Jose, CA, 2009.

[23] Y. Jin, N. Kupp, and Y. Makris, "DFTT: Design for Trojan Test," in *17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Athens, 2010.

[24] S. Drzevitzky, U. Kastens, and M. Platzner, "Proof-Carrying Hardware: Towards Runtime Verification of Reconfigurable Modules," in *International Conference on Reconfigurable Computing and FPGAs (ReConFg'09)*, Quintana Roo, 2009.

[25] L. Kim, J. Villasenor, and C. K . Koc, "A Trojan-Resistant System-On-Chip Bus Architecture," in *TheIEEE Military Communication (MILCOM'09)*, Boston, 2009.

[26] S. Narasimhan, and S. Bhunia, "Hardware Trojan Detection," in *Introduction to Hardware Security and Trust*, New York, Springer, 2012, pp. 339-365.

[27] "Benchmark circuits," [Online]. Available: http://www.pld.ttu.ee/~maksim/benchmarks/. [Accessed 27 2 2016].

[28] S. Buchner, K. Kang, W. J. Stapor, A. B. Campbell, A. R. Knudson, P. Mc Donald, and S. Rivet, "Pulsed Laser-Induced SEU In Integrated Circuits: A Practical Method For Hardness Assurance Testing," *IEEE Transactions on Nuclear Science,* vol. 37, no. 6, pp. 1825-1831, 1990.

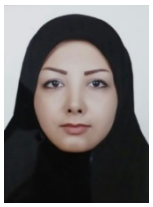
[29] A. Sarafianos, O. Gagliano, V. Serradeil, and M. Lisart, "Building the Electrical Model of the Pulsed Photoelectric Laser Stimulation of an NMOS Transistor in 90nm Technology," in *International Reliability Physics Symposium (IRPS)*, Anaheim, CA, 2013.

[30] A. Sarafianos, O. Gagliano, M. Lisart, and V. Serradeil, "Building the Electrical Model of the Pulsed Photoelectric Laser Stimulation of a PMOS Transistor in 90nm Technology," in *International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, Suzhou, 2013.

[31] A. Sarafianos, and et al., "Robustness Improvement of an SRAM Cell Against Laser-induced Fault Injection," in *dfts*, 2013.

[32] M. Banga, and M. S. Hsiao, "A Region Based Approach for the Identification of Hardware Trojans," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, Washington, DC, USA, 2008.

[33] "User Manual," Synopsys Inc., [Online]. Available: http://www.syn-opsys.com. [Accessed 18 09 2015].



Najmeh Farajipour Ghohroud received the B.Sc. degree in computer engineering from Iran University of Science and Technology, Tehran, Iran in 2007. She received the M.S. degree in computer engineering from University of Tehran, Tehran, Iran in 2010. She is working toward the Ph.D. degree at the Department of Computer Engineering at Sharif University of Technology.
Her research interests include hardware security and trust, and embedded system design.
Email: farajipour@ce.sharif.edu

Shaahin Hessabi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 1986 and 1990, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, Ontario, Canada in 1995. He joined Sharif University of Technology in 1996, and is currently an associate professor at the Department of Computer Engineering.
His current research interests include System-on-Chip and Network-on-Chip, and VLSI design and test. He has published more than 100 refereed papers in the related areas. Dr. Hessabi has served as the program chair, general chair, and program committee member of various conferences.
Email: hessabi@sharif.edu

Paper Handling Data:

Submitted: 03.06.2017
Received in revised form: 10.09.2017
Accepted: 24.02.2018
Corresponding author: Dr. Shaahin Hessabi,
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.

[a] Design Compiler™ Is a Trademark of Synopsys, Inc.