Implicit Hardware Trojan: Principles and Enabling Methods

Nilanjana Das, Mattis Hasler, Sebastian Haas
Barkhausen Institut, Dresden, Germany
{*first name.last name*}*@barkhauseninstitut.org*

***Abstract*—The research community of hardware security has worked hard to develop effective defenses against a range of hardware Trojans (HTs). However, HT design research has consistently advanced faster than HT detection research. In this paper, we describe the concept of implicit hardware Trojan (IHT) design, its properties, enabling methods, and adversarial effects. The IHT advances the field of HT design research. We propose different kinds of multiple IHTs (MIHTs) and classify them in terms of considered logic gates used to design the MIHTs. Results from experiments demonstrate the IHT design's effectiveness and deceit.**

***Index terms* – Implicit Hardware Trojan (IHT), Multiple IHT (MIHT), HT detection, rare input pattern.**

I. Introduction

System on chip (SoC) designers are compelled to embrace third-party electronic design automation (EDA) tools and intellectual property (IP) cores due to resource limitations and time to market pressures. The hardware Trojans (HTs) being implanted into the IC supply chain by malicious entities has increased due to the third-party globalization of the semiconductor industry [1]. The efficiency of an HT detection methods solely depends on the available HT benchmarks. However, the available Trust-Hub [2] benchmarks are detectable because of their backdated trigger and payload mechanisms [3][4]. It is important to search for a new variation of HT design so that to reinforce the available HT detection methods. In this paper we elaborately describe the implicit HT (IHT) which is first introduced by [4]. An IHT is a malicious redundant circuitry embedded in a core, which propagates the malicious outcome for a rare trigger condition when the trigger signal is activated. We examine a set of logic gates and explain the implementation of IHT on those gates. We show that the IHT cannot be detected with the existing HT detection methods. Finally, we propose the concept of multiple IHTs, its classifications, and demonstrate one of them experimentally.

II. Implicit hardware trojan (IHT)

An HT generates malicious outcome only when the trigger condition is satisfied and the trigger signal is activated. For IHT the trigger signal can be activated multiple times, but the HT affected output propagates only when a particular trigger condition (rare) is encountered. Following example demonstrates the basic architecture of IHT.

Fig. 1a, A is the golden circuit (in green colour) and B and a 2-to-1 multiplexer (in red colour) are the inserted HT circuit. If the select line of the multiplexer S is active, the output of the golden circuit becomes a malicious one propagated from B. If we consider A as a 2-input OR gate and B as a 2-input XOR gate, Fig. 1b shows all the possible outcome of the circuit from Fig. 1a when S is active. For three input patterns "00, 01, and 10" (in green colour), there is no change in the output O although S is active. For input pattern "11" (in red colour) the output is changed by which the explicit result will propagate from the circuit. If we consider the input pattern "11" as the rare one, for "00, 01 and 10" the HT circuit works as an IHT. In these cases, the trigger signal S is active, but there is no change in the output of the circuit due to inserted IHT. Figs. 1c, 1d, and 1e demonstrates the same idea considering A and B as NOR and XNOR, AND and XNOR, and NAND and XOR, respectively.

The efficiency of a HT circuit depends on how it propagates the malicious outcome [3] without being detected. For an IHT, the trigger condition mainly creates the explicit occurrence of the HT circuit rare. This situation can be explained clearly with the help of two lemmas.

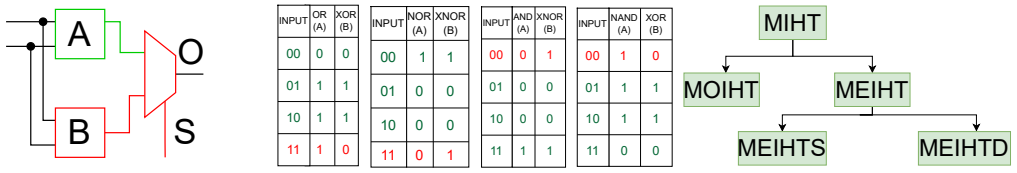
We consider a n-input circuit with 2^n number of possible input patterns which is divided into two disjoint sets X and Y. X contains the input patterns which appear frequently in run time, and Y includes the input patterns that appear rarely[1]. Hence, it is evident that $X \cup Y = 2^n$ and $X \cap Y = \phi$, and we assume $|X| \geq 1$, $|Y| \geq 1$, and $|X| >> |Y|$.

Lemma 1: The set X contains only the input patterns for which the original output is similar with the IHT generated output irrespective of the state of trigger signal.

Lemma 2: The set Y contains only the input patterns those are rare and at least one of these input patterns is responsible for generating the malicious outcome.

Lemma 1 ensures that if an input pattern appears frequently, it will not affect the original outcome. As X contains most of the possible input patterns and the trigger signal can be activated anytime for the input patterns that belongs to set X, it will not be suspected by the HT detection algorithms. Lemma 2 states that, from the set of rarely appeared input patterns, at least one pattern will trigger the malicious outcome. Therefore, the explicit condition must be rare enough to evade the HT detection algorithms. From Lemmas 1 and 2, we conclude that: the IHT can evade the detection algorithms by frequent activation of the trigger signal, and the malicious outcome will

[1]With the primary knowledge of inserted trojan, it is possible to determine Y at the design time.



(a) (b) (c) (d) (e) (f)

Fig. 1: Illustration of IHT: (a) basic IHT circuit, (b)-(e) Logic gates and IHT alternatives, (f) classification of MIHT

appear too rarely to be detected.

To demonstrate the idea of Lemmas 1 and 2, we consider the circuit in Fig. 1a by setting A as a 2-input OR gate and B as a 2-input XOR gate as described in Fig. 1b. A possible arrangement of X and Y are: X=$\{00, 01, 10\}$ and Y=$\{11\}$, respectively, which ensures all the aforementioned cases. Similarly, for the case in Fig. 1d, a possible arrangement of X and Y are: X=$\{01, 10, 11\}$ and Y=$\{00\}$, respectively.

III. Multiple Implicit Hardware Trojans (MIHTs)

In this section we propose the concept of multiple IHTs (MIHTs) embedded in an IP core. In this case, more than one logic gate will be replaced by the implicit logic circuit as mentioned in Figs 1b-1e[2]. Based on the selected logic gates, MIHTs can be classified into two categories as follows. (i) Multiple Homogeneous IHTs (MOIHTs): The altered logic gates are alike, e.g., all selected OR gates are replaced by XOR gates (Fig 1b). In this case, the set of rare patterns Y remains the same along the IP core. (ii) Multiple Heterogeneous IHTs (MEIHTs): The altered logic gates are different, e.g., in an IP core one OR gate is replaced by XOR gate (Fig 1b), and one AND gate is replaced by XNOR gate (Fig 1d). In this case, the set of rare patterns Y for each logic gate to be replaced may or may not be equal. The set of rare patterns for the IP core Y_{IP} will be union of each set of rare patterns Y of each altered logic gates, that is $Y_{IP}= \cup_{i=1}^{n} Y_i$, where Y_i is the set of rare patterns of i^{th} altered logic gate. The MEIHTs can be further classified into two categories based on the set of rare patterns described below. Fig 1f reports the overall classification of MIHTs. (i) Multiple Heterogeneous IHTs with Same rare conditions (MEIHTSs): In this case, the altered logic gates (different) have the same set of rare input patterns. For example, one OR gate is replaced by XOR gate (Fig 1b) and one NOR gate is replaced by XNOR gate (Fig 1c), and both have the same rare pattern "11". (ii) Multiple Heterogeneous IHTs with Different rare conditions (MEIHTDs): In this case, the altered logic gates (different) have different set of rare input patterns. For example, one OR gate is replaced by XOR gate (Fig 1b) and one NAND gate is replaced by XOR gate (Fig 1e), and they have different rare patterns "11" and "00".

IV. Experimental results and discussion

We implement MEIHTS (Fig. 1f) on ISCAS-85 benchmark circuit C432 (27-channel interrupt controller). In C432 an OR gate and a NOR gate are replaced by an XOR gate and an XNOR gate respectively for the MEIHTS implementation. The simulation is performed five times by increasing the number of input patterns with a step of 50. The ts1 and ts2 denote the trigger signals used for MEIHTS design. From Fig. 2 it is observed that the activation rate of ts1 and ts2 are 63% and 69% respectively when 50 input patterns are passed. The activation rates of ts1 and ts2 are decreased when the number of input patterns is increased. This proves that both the ts1 and ts2 are activated frequently but there is no effect on the original outcome. It is seen that for a particular input pattern, the malicious outcome is spotted. For the second simulation, one rare input pattern is passed intentionally for which the original output is replaced by the malicious outcome. Considering the wires with trivial activation rates, it will be impossible to detect the trigger signals ts1 and ts2 due to their higher activation rate.

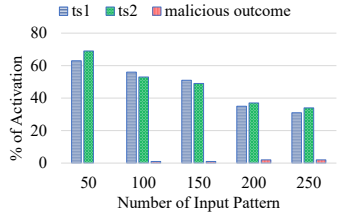


Fig. 2: Activation rates of ts1 and ts2 of MEIHTS on C432

V. Conclusion and Future Direction

This paper describes the concept of implicit HT (IHT), proposes idea and classification of multiple IHTs (MIHTs). It is observed that MIHT evades the HT detection methods by activating the trigger signals frequently, and shows malicious behavior only when the rare trigger condition is met. Consideration of other possible cases of IHTs and MIHTs, and corresponding detection strategies are left as the part of future direction.

References

[1] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE design & test of computers*, 27(1):10–25, 2010.

[2] M. Tehranipoor et al. Trusthub." [online]. available: http://trust-hub.org.

[3] S. Haider et al. Hatch: Hardware trojan catcher. *IACR Cryptol. ePrint Arch.*, 2014:943, 2014.

[4] J. Zhang, F. Yuan, and Q. Xu. Detrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware trojans. In *Proc. of ACM SIGSAC CCS*, pages 153–166, 2014.

[2]Note that IHT/ MIHT can be implement in several ways. In this paper we have limited our discussion into the mentioned basic gates only.