On-Demand Transparency for Improving Hardware Trojan Detectability

Rajat Subhra Chakraborty, Somnath Paul and Swarup Bhunia
Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, OH, USA
{rsc22, sxp190, skb21}@case.edu

Abstract: *Malevolent Trojan circuits inserted by layout modifications in an IC at untrustworthy fabrication facilities are difficult to detect by traditional post-manufacturing testing. In this paper, we develop a novel low-overhead design methodology that facilitates the detection of inserted Trojan hardware in an IC through logic testing. As a byproduct, it also increases the security of the design by design obfuscation. Application of the proposed design methodology to an 8-bit RISC processor and a JPEG encoder resulted in improvement in Trojan detection probability significantly. It also obfuscated the design with verification mismatch for 90% of the verification points, while incurring moderate area, power and delay overheads.*

Keywords: Design Obfuscation, Hardware Trojan Detection, Trust in IC.

I. INTRODUCTION

The issue of trust is an emerging problem in semiconductor integrated circuit (IC) security [1]. Economic motivations dictate manufacturing to be performed at a location which is physically remote from the design facility, often in a foreign country. In such scenario, a design layout can be reverse-engineered and tampered in an untrusted fabrication facility by the insertion of malicious "*Trojan*" circuitry that conditionally triggers circuit malfunction and thus can potentially result in catastrophic failures. These Trojan circuits are likely to be triggered under certain very rare input conditions to avoid easy detectability using manufacturing test. They may also be *autonomous* and *time-triggered,* and thus can affect the functionality only after long periods of in-field deployment. Fig. 1 shows a model for Trojan circuits that an adversary can incorporate. We consider two broad types of Trojan circuits. *Combinational Trojans* (Fig. 1a) are Trojan circuits where the triggering logic is a combinational Boolean function of some internal points. On the other hand, *Sequential Trojans* (Fig. 1b) are triggered by a sequence of events on the internal nodes and/or clock signal of a circuit. Detection of such Trojans by conventional post-manufacturing test with reasonable number of test vectors can be extremely challenging. There are two major reasons behind this. First, the adversary will try to make the Trojans trigger at a very rare condition and prevent any malicious alteration being easily observed. This reduces the effectiveness of functional or structural testing in terms of detecting a Trojan. Second, the number of possible Trojan instances is an exponential function of circuit nodes in case of both combinational and sequential Trojans.

Previous works in this area have explored *side-channel* information [2] in the form of de-noised power trace analysis as a way of detecting Trojans [3]. However, effectiveness of the approach in [3] reduces considerably with decreasing Trojan size and increasing process induced parameter variations due to reduction in signal-to-noise ratio. On the other hand, logic testing based Trojan detection can be very robust for detecting small Trojans in a multi-million gate circuit under large process

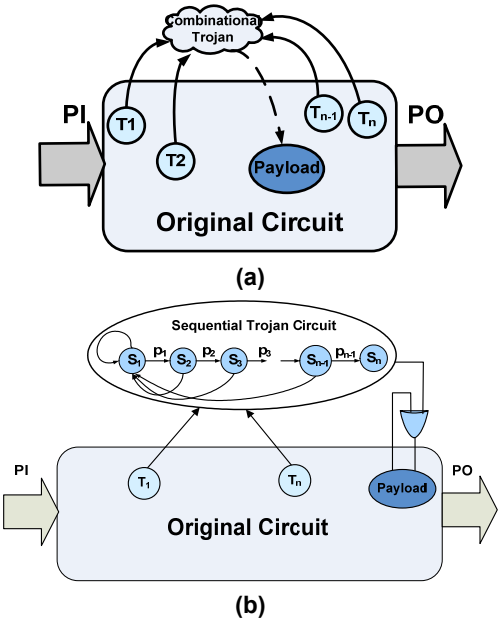

Fig. 1: Models for a) combinational and b) sequential Trojan circuits we have considered.

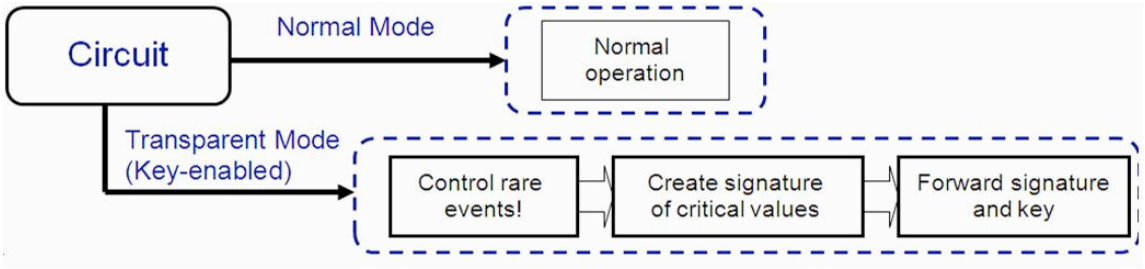

Fig. 2: Modes of operation in the proposed *on-demand transparency* scheme.

variations. However, to detect a Trojan using logic testing, one needs to generate an input condition to activate the trigger function that causes the alteration in logic value at an internal node (referred as *payload*) and simultaneously observe the effect at primary output. It can be very challenging to generate such input conditions since a trigger function will likely consist of multiple rare conditions on the internal nodes (or a sequence of them) and the payload can have poor observability.

In this work, we propose a design methodology called *On-Demand Transparency* that can facilitate logic testing based Trojan detection. The basic idea is to define a special mode of operation, referred as *Transparent Mode*, in a multi-module system where a specific signature is generated at the primary output upon application of a user-defined *key* at the input. Tampering of a node in *any* of the constituent modules by Trojan insertion causes the signature to be different from the expected one, thus detecting the existence of a Trojan. The modification of the design is based on a judicious choice of nodes which are susceptible to Trojan attacks. The choice is guided by the nodes controllability and observability values.

It can be noted that the proposed design modification methodology also helps to obfuscate the design by altering its interface (number of I/O ports), gate-level structure and I/O functional behavior. By choosing the appropriate nodes for design modification, we can achieve high levels of security while incurring only moderate area, power and delay overheads. Next, we explain the methodology in more details.

II. PROPOSED METHODOLOGY

The proposed methodology is based on the assumption that the adversary has information about the functionality and logic structure of the IC before modifying it. In this scenario, a likely approach to tamper the IC would be to insert a Trojan in such a way that the Trojan would affect the normal functionality of the system under very rare input or temporal conditions. An example is a setup where a Trojan is triggered only on the overflow of an arithmetic operation and it affects the "write enable" of a memory array. Under this assumption, the adversary is most likely to choose the *least controllable* nodes of the design to trigger the Trojans and the *least observable* nodes of the design as payloads. In this way, the probability of the Trojans getting triggered and simultaneously observed at output ports during regular functional testing gets minimized. Thus, in the proposed design methodology we create a special mode of system operation (as shown in Fig. 2) where we target efficiently controlling the low-controllability nodes and observing the low-observability nodes in the design to manifest the existence of a Trojan.

In the proposed design approach, a few extra input ports called the "key ports" are added to each module, so that on the application of a particular "key" pattern at

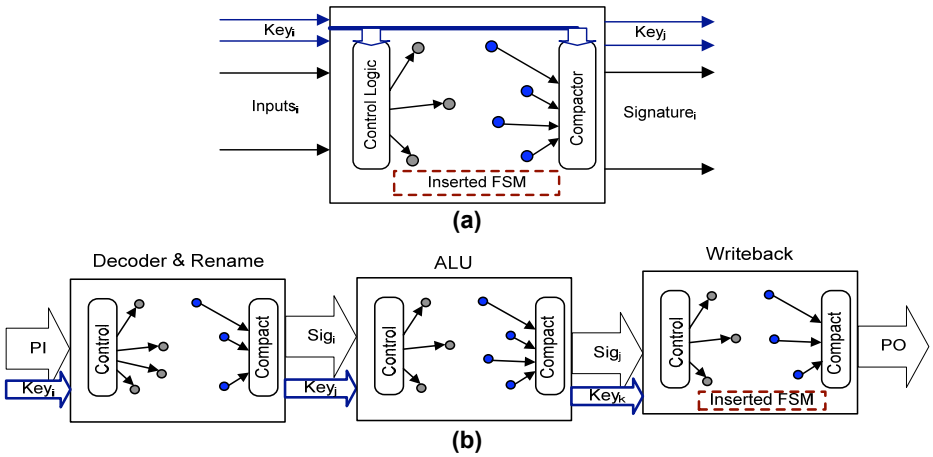

Fig. 3: Proposed design method for a multi-module design: (a) block and (b) system level diagram illustrating the design modification.

Table I: Design overhead

Circuit / Parameter	8-bit RISC CPU	JPEG Encoder
Area (%)	5.2	4.5
Power (%)	13.5	12.8
Delay (%)	5.2	5.0
Extra I/O Ports	9 (in 67)	9 (in 56)

these ports, an inserted state machine in the design is activated and after a certain number of clock cycles, the module enters a special operating mode, which we call the "*transparent mode*". During this mode, the key enables the inserted "control logic" circuitry in the module to force the least controllable nodes in the circuit to their corresponding low controllable values, in effect simulating the occurrence of the "rare events" that are likely to trigger a Trojan. To observe the effect of a Trojan at primary outputs, the logic values at the low-observable nodes in the circuit are compacted into a signature and the signature is propagated to the next module. We use the *X-COMPACT* compaction scheme [5] to observe large number of nodes simultaneously and limit the number of extra output ports to be introduced to the modules. The values at the controlled nodes are used to produce the "*key*" for the next module which is triggered to "transparent" mode by the "*key*". In the next module, the signature of the previous module is combined with that of its own and passed forward. Thus, application of a particular key at the special "key ports" at the input effectively makes the design "*transparent*", as the effect of the applied key is propagated from one module to the next till it reaches the primary output. The appearance of the expected "*signature*" at the primary output by application of the key at the primary input provides confidence about trust of the IC under test. After the proposed design modifications, the gate level netlists of the different modules are re-synthesized. The resynthesis is performed with the same set of design constraints as original synthesis. Fig. 3 shows the design modifications at block (Fig. 3a) and system level, where we consider three stages of a processor pipeline (Fig. 3b).

The proposed technique also helps to *obfuscate* and *authenticate* the design by adding extra ports, state elements and gates to the design and then resynthesizing it. The modification also makes reverse engineering of the design significantly more difficult as observed by our initial analysis. It can thus protect hardware intellectual property (IP) cores. The output signature obtained upon application of the key at primary input can be used to authenticate a design.

III. SIMULATION RESULTS

We applied the design methodology to two example designs. The first design we considered is an 8-bit RISC CPU [4], while the second is a JPEG Encoder [6]. Each of these designs consisted of four separately synthesizable modules. We used *Synopsis Design Compiler* for the synthesis with the 250 nm LEDA standard cell library. Table I shows the design overhead for the modified design in terms of area, power, delay and increase in the number of I/O ports compared to the original number of ports. We subsequently used *Synopsis Formality* to check how successful our technique was in obfuscating the design. For both the designs, *Formality* reported mismatch for 90% of the verification points. For the test cases, we performed random insertion of 20 combinational Trojan circuits of varying size and complexity and tested the effectiveness of proposed approach through logic simulations. The simulations reported mismatch of the master signature in 98% of the cases where the Trojans were present. We also applied 1000 random patterns to both designs and observed that <10% of Trojans were detected by random vectors.

IV. CONCLUSION AND FUTURE WORK

We have presented a low-overhead design methodology to improve the Trojan Detectability of a complex multi-module circuit using logic testing. The methodology is based on incorporating a key-enabled additional mode of operation, when low-controllability nodes are controlled and low-observability internal nodes are observed at primary outputs. This methodology also effectively obfuscates the design and makes reverse-engineering of a design difficult. Future work would involve investigating the effectiveness of the approach for combinational and sequential Trojans in complex System on Chip (SoC), developing an automatic synthesis approach to incorporate required design modifications and improving the design overhead.

REFERENCES

[1] DARPA BAA06-40, "Trust for Integrated Circuits", *http://www.darpa.mil/BAA/*

[2] P. Kocher, J. Jaffe and B. Jun, "Introduction to Differential Power Analysis and Related Attacks", available online at *www.cryptography.com/resources/whitepapers/DPATechInfo.pdf*

[3] D. Agrawal et al, "Trojan Detection using IC Fingerprinting", *IEEE Symposium on Security and Privacy*, 2007.

[4] J. Clayton, "The risc16f84: an 8-bit CPU," [Online]. Available: http://www.opencores.org/projects.cgi/web/risc16f84

[5] S. Mitra, and K.S. Kim, "X-Compact: An Efficient Response Compaction Technique," *IEEE Tran. on CAD*, vol. 23, no. 3, pp. 421–432, Mar. 2004.

[6] "The JPEG Encoder Soft Core," [Online]. Available: http://www.opencores.org/projects.cgi/web/jpeg/