Integrated Circuit Security - New Threats and Solutions

Miron Abramovici and Paul Bradley

DAFCA
10 Speen St
Framingham, MA 01701
+1-774-204-0056
{miron,paul}@dafca.com

ABSTRACT
In this paper, we present new threats to the security of integrated circuits (ICs) and outline a new solution.

Categories and Subject Descriptors
D.7.3 [Reliability and Testing]:

General Terms

Keywords
Integrated circuits, security, Trojan attacks, tampering attacks, detection, countermeasures.

1. INTRODUCTION
The problem of IC security is extremely important today, since ICs are involved in all critical aspects of our lives. For economic reasons, nearly all ICs are fabricated by foreign foundries and they include intellectual property (IP) cores supplied by many third-party IP providers. In addition, they rely on outsourced design and test services, and use automation tools from many different vendors. Such a design and manufacturing process provides an adversary with many opportunities to insert Trojan-horse logic to sabotage the mission of an IC used in critical applications. We will show that today there are no reliable methods that can guarantee the pre-deployment detection of Trojan intrusions. This fact has large national-security implications: a Trojan attack can create havoc in the basic civilian infrastructure (electric grid, communication, and banking networks), sabotage critical missions, disable weapon systems, or provide back-door access to otherwise highly-secure systems.

2. INTRUSIONS AND ATTACKS
We distinguish between intrusions and attacks. An intrusion denotes a hostile modification of a circuit that occurs before deployment (during design or manufacturing), providing the basis for an attack that may occur later during the normal operation of the deployed IC. Intrusions may modify the design at different stages, such as RTL (register transfer language), gate-level netlist, or GDSII layout. Intrusions may target the functional logic or the infrastructure logic inserted in the design to enhance the testability, the reliability, or the manufacturability of the chip. Intrusions such as focused ion-beam (FIB) circuit modifications target an already manufactured chip.

Not all attacks require prior intrusions. For example, non-invasive tampering attacks, such as subjecting the chip to radiation or operating the chip outside its specified ranges for voltage, temperature, or frequency, can occur without any circuit modifications.

Ideally, we would like to detect the intrusion before the attack occurs. Pre-silicon, we can detect an intrusion during the design verification process. Post silicon, the detection opportunities are during manufacturing test, silicon validation, and system test and validation. Of course, pre-silicon detection would be better than post-silicon detection. However, Trojans may be designed so that their pre-silicon is practically impossible. For example, a Trojan may be inserted in an IP core provided as an RTL model. Such a Trojan may be designed to be activated when the IC is already deployed in the field by a time-bomb mechanism (for example, “disable the core one month after system reset”) or by a booby-trap mechanism (for example, “set the core in test mode after 100 million packets have been processed”). The large delay assures that the Trojan will not be activated during pre-silicon verification (using simulation or emulation) or even during silicon validation. Formal verification methods are not capable of dealing with the complexity of current designs. If the Trojan is never activated during simulation or emulation, it could be identified by analyzing areas with low functional coverage during pre-silicon verification. But typically a complex IC is sent for tapeout with many areas having incomplete coverage, since complete coverage cannot be achieved in practice.

Post-silicon, most proposed Trojan detection techniques analyze different physical characteristics of the IC (such as power consumption, timing variations, layout structures) with respect to a golden-reference model. However, the presence of a Trojan in the RTL model of the device precludes having a golden model; this invalidates the basic assumption which is the cornerstone of most detection methods. Even if an RTL golden model does exist, such a model covers only the functional logic. Insertion of infrastructure logic in the design provides many additional opportunities for inserting hidden Trojans.

Consider a two-stage intrusion that first inserts a Trojan in the layout model using spare gates without any connection to the functional logic, and after manufacturing follows up with a FIB circuit modification that connects the Trojan to the functional
logic. Such a Trojan would be invisible until the FIB intrusion occurs. Destructive reverse-engineering techniques are not applicable since the FIB attack would target only a subset of the fabricated ICs. Today we don’t have any scalable non-destructive technique able to detect modifications introduced by a FIB attack. But even if we had such a technique, it would be extremely difficult to use it effectively on a large number of ICs.

The above analysis makes clear that we cannot guarantee that ICs deployed in the field are Trojan-free. (This does not mean that we should not use pre-deployment detection techniques - they are necessary but not sufficient.) In the following we introduce a new approach detecting post-deployment Trojan or tampering attacks.

Figure 1 illustrates the architecture of a system-on-chip (SoC) with inserted DEFENCE logic. The insertion is done at RTL under user control; the user selects the important signals to be monitored. The instrumented RTL model is processed by the standard customer design flow.

Signal Probe Networks (SPNs) are configured to select a subset of the monitored signals and transport them to Security Monitors (SMs). An SPN is a distributed pipelined MUX network designed to support multiple clock domains. An SM is a programmable transaction engine configured to implement an FSM to check user-specified behavior properties of the signals currently brought to its inputs to be analyzed. The Security and Control Processor (SECOPRO) reconfigures SPNs to select the groups of signals to be checked by SMs and reconfigures SMs to perform the required checks. The configuration of one SM does not interrupt the normal system operation or the checking activity of other SMs. All the configurations are encrypted and stored in the Secure Flash memory.

The security checks are application-dependent and circuit-dependent. The DEFENSE logic performs two types of checks. The first category is a set of user-specified security violations such as:

- An attempt to access to a restricted address space
- A control signal supposed to be inactive is activated
- Denial of service
- A core responds to a request addressed to another core
- A core whose clock is turned off has output changes
- A core enters a mode of operation which is illegal in the current system state

The second category of checks consists of the general correctness properties of the system behavior. The rationale for this is that an activated Trojan will make the system operate in an incorrect way. These checks may be the same that were performed in pre-silicon verification. For example, these may include the assertions used in simulation to verify the correct implementation of the standard communication protocols used on chip (AMBA, PCI, etc.), or the behavior of a specific block.

DAFCA provides tools to define the “personalities” for the reconfigurable instruments. To define a check the user specifies the FSM to be implemented by a security monitor. All checks are prepared and verified pre-deployment in a secure environment and their corresponding configurations are pre-loaded in the Secure Flash. The chip manufacturer does not know the contents of the flash.

In a powered-off chip, the reconfigurable logic is “blank” (unprogrammed) and thus its function is perfectly concealed from attackers trying to reverse engineer the device. Similarly, the control logic of SECOPRO is configured (at power-on) from the Secure Flash, so its function is also invisible to an attacker. As shown in Figure 1, the security checkers are not accessible from either the functional logic or from the embedded software. Similarly, SECOPRO is invisible for the other on-chip application processors.

SECOPRO runs one group of concurrent checks at a time for a specified interval and executes all the groups in a continuous loop. The configurations for SPNs and SMs are retrieved from the Secure Flash. The configuration process does not interrupt the normal operation of the system. The groups of checks may be

3. A NEW DEFENSE MECHANISM
3.1 Detecting Trojan Attacks

Clearly we need infrastructure logic to implement post-deployment security checks during the normal operation of the system. The main problem is that there are a large number of needed checks and implementing all of them in hardware may have a prohibitive cost. Security requires that checkers are invisible to an attacker and to the embedded software. Today there is no efficient solution satisfying these requirements.

Our approach involves adding reconfigurable Design-For-Enabling-Security (DEFENSE) logic to the functional design to implement real-time security monitors. The DEFENSE infrastructure consists of distributed instruments that can be repeatedly configured to dynamically implement different security checks to constantly monitor the system operation to detect unexpected or illegal behavior. Reconfigurability allows a large number of checks to be implemented by time-sharing the same hardware. Similar concepts are used in a commercially-available solution provided by DAFCA (www.dafca.com) for in-system silicon validation and debug, already in use by the largest semiconductor manufacturers[1].
overlapping so that more important checks are run more often. A self-check of the DEFENSE platform is also run periodically to assure its integrity.

3.2 Detecting Tampering Attacks

By design, the DEFENSE infrastructure is ideally suited to detect post-manufacturing tampering attacks. Unauthorized modifications to the software, unauthorized access to memory, and other hardware-based tampering methods used to compromise the system can be detected with this infrastructure. Moreover, this infrastructure is ideal for monitoring system performance characteristics. Any tampering or denial-of-service attack that leads to modifications in the software or hardware “signature” will be detected and can trigger appropriate countermeasures.

3.3 Countermeasures

When an attack is detected, the first step should be to deploy countermeasures such as disabling a suspect block or forcing a safe operational mode. The DEFENSE platform can implement countermeasures by controlling specified signals. This is illustrated in Figure 1 by the Signal Control block that enables SECOPRO to override the value of a signal. For example, if a core exhibits illegal behavior, SECOPRO may isolate that core by disabling its clock, powering it off, resetting it continuously, or forcing safe values on its outputs.

These represent only basic countermeasures that need to be integrated in a system-level solution for surviving detected attacks. System-level recovery may combine techniques such as provision of fail-safe states, spare logic to replace misbehaving logic, and returning to last safe checkpoint. Other options include disabling the entire system and/or wiping out critical data. These topics, however, are beyond the scope of this paper.

4. CONCLUSIONS

We introduced the reconfigurable DEFENSE platform for post-deployment detection of Trojan attacks in ICs deployed in critical application. The technology is a natural extension of a commercially-available silicon-proven platform used for in-system silicon validation and debug.

The Security Monitors can perform a large number of complex on-line security checks. These checks detect Trojan or tampering attacks and do not rely on a golden reference model. The DEFENSE platform can support basic countermeasures to respond to a detected attack. Our approach is application-independent and technology-independent.

The DEFENSE logic is invisible to the application and system software and therefore protected from software-based attacks. Being reconfigurable, its function is not visible to reverse engineering efforts that analyze the circuit. Unlike FPGA-like hard macros, the soft reconfigurable logic cannot be distinguished from the functional ASIC logic. Also the configurations are encrypted and stored in the Secure Flash. Hence the DEFENSE logic is practically invisible to an attacker. We can extend these techniques to also obfuscate the functional logic to make the entire IC practically impossible to reverse engineer.

5. REFERENCES