ERSA – INVITED TALK/LECTURE

Addressing the Challenges of Hardware Assurance in Reconfigurable Systems

William H. Robinson, Trey Reece, and Nihaar N. Mahatme
Security and Fault Tolerance (SAF-T) Research Group
Department of Electrical Engineering and Computer Science, Vanderbilt University
Nashville, TN, USA

Abstract - Despite the numerous advantages of nanometer technologies, the increase in complexity also introduces a viable vector for attacking an integrated circuit (IC): a hardware attack, also known as a hardware Trojan. Since such an attack is implemented within the hardware of a design, it is generally undetectable to any software operating on this circuitry. To make matters worse, a hardware attack could be introduced at almost any point in a design’s development cycle, be it through third-party intellectual property (IP) licensed for a design, or through unknown modifications made during the fabrication process. This malicious hardware could act as a kill-switch for a vital device, or as a data-leak for sensitive information. Activation would occur at some predetermined time or by a trigger from a malicious agent. An effective method is required to find such unexpected functionality. This paper describes several key challenges to be addressed in order to provide hardware assurance for trustworthy systems. We examine the platform of field programmable gate arrays (FPGAs) both for their potential vulnerability to threats within third-party IP as well as their capability to accelerate the testing of those modules.

Keywords: Trusted hardware; malicious hardware detection; security; FPGAs; third-party intellectual property (IP)

1 Introduction

Trustworthy computing (with software) cannot exist until there is trustworthy hardware on which to build it [1]. To most designers, one of the advantages to implementing a design in hardware instead of as a software implementation is the secure nature of hardware. The assumption is prevalent that hardware is secure while software can be attacked. Unfortunately, this is a false assumption, created due to a lack of security awareness with increasingly complicated circuits. Advancements in process technology provide designers with the ability to put more transistors on a single silicon die [2] to fabricate increasingly complex designs. Unfortunately, the contents of these chips can be obscured, leading to potential security vulnerabilities within the hardware. A full design could have logical blocks contributed by dozens of different sources, with hundreds of different people contributing to the overall design. In some cases, these designers may have nothing to do with each other, and may come from outside of the company. There exists the threat that malicious agents can compromise the supply chain of integrated circuits (ICs) [3, 4] by inserting hardware Trojans (i.e., tiny circuits implanted in the original design to make it work contrary to the expected way in certain rare and critical situations [5]). In addition, the capital investment required for semiconductor foundries has limited the number of companies who fabricate their own ICs. Many companies have become “fabless” and rely upon overseas foundries to manufacture their designs (Table 1); these designs are then returned as packaged chips. The challenge of detecting malicious hardware requires that the testing methodology identifies unknown functionality within a chip after fabrication.

Table 1: 2011 Top 10 Semiconductor Foundries [6]

<table>
<thead>
<tr>
<th>Rank</th>
<th>Foundry</th>
<th>Location</th>
<th>Sales (USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSMC</td>
<td>Taiwan</td>
<td>14,533M</td>
</tr>
<tr>
<td>2</td>
<td>UMC</td>
<td>Taiwan</td>
<td>3,604M</td>
</tr>
<tr>
<td>3</td>
<td>GlobalFoundries</td>
<td>U.S.</td>
<td>3,580M</td>
</tr>
<tr>
<td>4</td>
<td>SMIC</td>
<td>China</td>
<td>1,319M</td>
</tr>
<tr>
<td>5</td>
<td>TowerJazz</td>
<td>Israel</td>
<td>613M</td>
</tr>
<tr>
<td>6</td>
<td>IBM Microelectronics</td>
<td>U.S.</td>
<td>545M</td>
</tr>
<tr>
<td>7</td>
<td>Vanguard International</td>
<td>Taiwan</td>
<td>516M</td>
</tr>
<tr>
<td>8</td>
<td>Dongbu HiTek</td>
<td>South Korea</td>
<td>483M</td>
</tr>
<tr>
<td>9</td>
<td>Samsung</td>
<td>South Korea</td>
<td>470M</td>
</tr>
<tr>
<td>10</td>
<td>Powerchip Technology</td>
<td>Taiwan</td>
<td>431M</td>
</tr>
</tbody>
</table>

Furthermore, different points of insertion can also involve different types of Trojans. A Trojan inserted at fabrication might utilize direct physical changes, due to the lack of a digital copy of the Trojan. On the other hand, a Trojan inserted through third-party intellectual property (IP) could pretend to be a type of digital watermark, yet hide additional malicious functionality. The reuse of IP makes it difficult to guarantee the security of a system when the underlying components are untrusted [7]. For example, a
design might include licensed design modules from vendors supplying third-party intellectual property, requiring techniques to ensure the trustworthiness of those modules [8-11]. For reconfigurable systems using field-programmable gate arrays (FPGAs), third-party IP becomes a likely attack vector. Some approaches with FPGAs attempt to isolate modules within the system’s implementation [12], or establish a root of trust within the FPGA fabric [13].

The concept of trust requires an accepted dependence or reliance upon another component or system [14]. In an age where hardware complexity provides the means to hide malicious hardware, the assumption that the hardware is secure can be misleading. Although software attacks are still the most common, a hardware attack emerges within the realm of possibility. Standard verification techniques ensure that a design meets the minimum functional requirements, but new methods of verification are required to guarantee that a design performs its intended function but nothing more. This paper discusses the challenges of developing trustworthy, reconfigurable computing systems. It is crucial for a designer to determine the trustworthiness of the design, as well as what possibilities are available for compromising that design. A solution for hardware assurance likely needs some automation to cover the potential test vector space. Reconfigurable hardware offers the possibility to accelerate the process.

The rest of this paper is organized as follows. Section 2 discusses hardware assurance and the basis for a root of trust. Section 3 provides a perspective on risk management by vendors and designers. Section 4 describes detection methods that have been developed and presented in the literature. Section 5 proposes a potential hardware testbed where field-programmable gate arrays (FPGAs) could be used to accelerate the verification process. Finally Section 6 summarizes the paper and offers some potential directions for future research.

2 Hardware Assurance

![Figure 1: Linkages among hardware and software for secure and reliable computing](image)

Many systems use hardware as the root of trust in order to defend against software-level attacks. Consequently, there is significant research on software assurance. However, viewing the system strictly in terms of hardware and software is a coarse-grained analysis. Understanding the linkages among technology, architecture, communication, and the application domain is critical for development of a trusted system (Figure 1). This section discusses the threat model used and its potential to affect full computing systems. It also describes a taxonomy for understanding malicious hardware and its potential impact on semiconductor intellectual property.

2.1 Threat model

One of the most insidious methods of attacking a circuit is by modifying its hardware in a malicious way. To put it simply, a hardware Trojan is created by discreetly inserting hidden functionality into a hardware design. This insertion can occur at any stage in a production path, and could have devastating effects on the final design. Such Trojans can have a variety of functionality, ranging from denial-of-service functionality that gives designs a controllable kill switch, to hidden data-leaks that can leak sensitive information [14].

One of the earliest papers covering the concept of Hardware Trojans was published by a group of researchers at the University of Champaign-Urbana [15]. This research included the design and test of a variant of the Aeroflex Gaisler LEON 3 [16] processor, called the Illinois Malicious Processor (IMP). The IMP was a fully functional version of the LEON 3 that operated normally in almost all circumstances, with the sole exception of one trigger: the receipt of a specially crafted corrupt network packet. Triggering this functionality would then switch the processor into a new shadow mode where the processor would accept and perform commands sent over the network. The shadow mode allowed an attacker to both compromise and hijack a system running on this processor, regardless of any security measures in the software. Additionally, this modification only required the insertion of 1,341 gates to the existing circuit, which originally contained over 1 million. Detecting such an insertion representing 0.1% of the circuit poses a significant problem. Even in much smaller circuits, the percent impact of hardware Trojans on the total area of a circuit is less than 0.5% [17, 18].

2.2 Classification of malicious hardware

The structure of a hardware Trojan can vary greatly depending upon intended functionality and payload [19]. A well-placed bug in a critical location can be as detrimental as a secret data-leak in a strong cryptosystem. Some Trojans are triggered via a specific sequence of inputs that are unlikely to occur in standard operation, and other Trojans are continuously active with an indiscernible payload. A taxonomy proposed by Karri et al. (Figure 2) [20] organizes Trojans based on 5 characteristics: (1) the point at which the Trojan enters the design, (2) the abstraction level of the Trojan, (3) the type of triggering which activates the Trojan, (4) the effect/payload of the Trojan, and (5) the location of the Trojan in the design. A similar taxonomy proposed by Wang et al. [21] focuses on three factors: (1) the physical characteristics (i.e., structure), (2) the activation characteristics (i.e., trigger), and (3) the action characteristics...
Additionally, while a large number of attacks fall under the classification of a hardware Trojan, the detection techniques are greatly dependent upon the individual characteristics of such Trojans.

2.3 Impact on semiconductor intellectual property (IP)

Depending on the method through which a Trojan is inserted, possible detection methods vary greatly [22-24]. Semiconductor IP has become a key part of electronics design because it can reduce IC development costs, accelerate time-to-market, reduce time-to-volume, and increase end-product value [25]. (According to Gartner Dataquest, the semiconductor IP market will reach $2.3B in 2014 [26].) Another confounding factor that increases the difficulty of developing countermeasures is that few attacks have been found in the wild. Instead, researchers must rely upon example attacks developed as benchmarks to illustrate the threat of malicious hardware. Unfortunately, these example attacks can often contain unnecessary functionality, making the detection of such an attack significantly easier. To make progress in this research area, it is necessary to understand both the attack and the defense of digital designs [27]. The Trust-Hub research community [28] developed as a forum to host and exchange resources related to hardware security and trust. It has grown to contain a significant number of tools and benchmarks, becoming the largest repository of hardware Trojans available to the public. It is supported by the National Science Foundation (NSF) and continually grows each year as contributors submit further resources.

3 Risk Management in the Supply Chain

When determining the security of a design, the first step is to identify clearly what types of steps in the design flow can be trusted, and what cannot. This determination might change depending upon the types of circuits and their implementations, but typically a vendor will trust its in-house design process and acknowledge the potential vulnerability of external design. Of course, there is the possibility of insider threats [29, 30].

3.1 In-house design

A simplifying assumption made for the purpose of this discussion is that all in-house design can be considered trusted. Under no circumstances does this mean that there are no security leaks, attempted sabotage, theft, or other problems within an organization. In fact, organizations have experienced this type of in-house threat. However, there are effective methods to resolve these threats that can be put into place. It is difficult to sabotage a design secretly if all changes to a digital design are tracked and logged with significant oversight on all changes. To put it simply, in-house design has its own process of verification that acts completely separately from other types of verification. The point of this assumption is to clearly define external attack vectors in order to most effectively block possible attacks. This allows a designer to guarantee that every possible step in a design is covered from attacks.

3.2 External design

After declaring all in-house work as trusted, the next step is to declare all work done outside of an organization as suspect. Any production step in which a design is modified by or in the care of an outside source can represent a possible vector for an attack. For each step, it is important to identify what attacks might be made by a third-party during this opportunity, and determine methods of either preventing or identifying such attacks. For example, a medical device company designing a pacemaker might license a wireless controller block from a vendor marketing third-party intellectual property (IP). It could be disastrous if this controller had malicious functionality hidden by the designer. In such a situation, it is foremost to identify the risk posed by incorporating this untrusted block in a design.
3.3 Vulnerabilities in the supply chain

One reason that external resources are considered universally untrusted is because of the difficulty in tracking the source of an external resource in the supply chain, regardless of the accompanying documentation. This has been a significant issue with defense contractors in the past few years, with regards to actual physical chips often purchased from reputable vendors or resellers. For example, a fiasco involving the United States Navy was made public in 2010, when a company called VisionTech was charged with selling over 59,000 microchips that contained hidden kill-switch functionality. This functionality would allow an attacker to disable whatever was running on these chips, including missiles, communication equipment, and other military vehicles. For years, this company had been importing counterfeit chips from China, and marketing them to defense companies as military grade microchips [31]. The list of these companies included: (1) BAE systems, which provided Identification Friend-or-Foe (IFF) systems to the U.S. Navy, and (2) Raytheon Missile systems, which supplied chips for use on F-16 fighter planes. Unfortunately, VisionTech is not the only reseller to buy cheap microchips from overseas and sell them domestically. Another similar example of corruption in the supply chain is the 2005 example of United Aircraft and Electronics, a company in which the operator was sentenced to 188 months in prison for false certification of aircraft parts sold [32]. Another 2002 example was the case of United Space Alliance, a company which bid and received a $24 million contract with NASA to supply military grade 8086 microprocessors for use with the space shuttle computers. This company then proceeded to purchase used computers off eBay and pull commercial-grade 8086 processors off the motherboards [32]. Commercial-grade chips would almost certainly have difficulties operating in the adverse environments required by the space shuttle computers. Unless it is possible to completely track the life of a resource, then that resource should be considered suspect. Since verification of an external resource is generally a simpler task than a full forensic investigation of the history of a resource, verification is the preferred method of determining whether something can be considered trusted.

4 Detection Methods

The majority of the existing methods proposed for identifying malicious hardware use the fabricated device; they can be classified into two types: (1) methods that detect changes on the transient current response drawn from extra circuitry on the chip [33-35], and (2) methods that detect timing differences due to the additional circuitry on the chip [36, 37]. A golden chip must be used as the trustworthy baseline in order to measure the deviation by a suspected chip. These methods assume that a trustworthy chip has already been identified, but do not address the issue of how to identify that chip in the first place. There are also some approaches that have attempted to encode signature information (i.e., a watermark) into the design to prevent unwanted piracy of ICs [38-40] or use side-channel measurements to determine the signature of a design [33, 41]. In addition, fault injection could be used to provide hardware assurance [42].

4.1 Physical testing

After the fabrication stage, the individual packaged chips are subjected to a large amount of testing in order to make sure that the designs work as intended. This step can be very involved, depending upon the complexity of the chip. This can require expensive testing equipment and a significant investment of time in order to fully verify a circuit. While this step can be done entirely in-house, outsourcing it to save costs would introduce an opportunity for an attacker to replace chips with compromised ones. Generally, the test vectors chosen will be completely trusted. The test sequences can be chosen entirely in-house, and can be supplied entirely from a known trusted ATPG algorithm. Physical testing typically requires a golden copy of the design and sensitive measurement equipment. Even then, there are still challenges due to the potential of process variation that masks the response [43]. Another method of testing/authentication involves the use of physical unclonable functions (PUFs) to provide challenge/response pairs for a design’s implementation [44, 45]. In order for a Trojan to remain hidden, there are three main characteristics that directly contribute to the difficulty of identification. If even one of these characteristics is lacking, then the difficulty in detecting the Trojan will be reduced.

Small Size: As Trojans can be constructed using a fraction of a percent of the components in the overall circuit, they can be quite small and still attain the desired functionality. However, the larger the Trojan grows, the more circuitry is added to the circuit, thus affecting its functionality. Even if the Trojan is not triggered, some inputs can activate smaller sections of the Trojan, changing the power consumed by the chip. Some techniques involve partially activating the Trojan circuitry in order to make it easier to detect [46, 47]. Additional circuitry is also more likely to displace the existing circuitry, compromising the second desired characteristic of hardware Trojans.

Low Displacement: When inserting a Trojan, it can be necessary to relocate existing circuitry, in order to make room for malicious components. However, such displacement of existing components can have a significant effect on side-channel measurements, making it possible to detect the malicious circuitry [22, 33, 35, 36]. In some cases, a very small Trojan added to a circuit could have a significant effect on the timing response of a circuit, especially if an automatic place-and-route function is implemented. In this case, manual placement of the Trojan circuitry in the layout can minimize the displacement of existing circuitry and help the Trojan to remain covert.

Resistance to unintended triggering: The last characteristic necessary for a Trojan to remain undiscovered is simply for it to be difficult to trigger accidentally. It does not matter how large the Trojan is, or how artfully placed the
components are if the Trojan is found during routine testing, such as standard logical verification. If the Trojan is always on and lacks a trigger, then the payload needs to be something discreet that does not appear on standard tests. For example, the Trojan in the modified LEON3 processor [15] was triggered via a uniquely crafted network packet, which would normally be treated as corrupt. Such a possible input would likely never be tested, simply because it is impossible to test every possible input on every possible state. However, this inability to test every possible input is what makes hardware Trojans effective as malicious attacks.

4.2 Third-party IP

As third-party IP is supplied from an external source, there is no baseline with which to compare the IP to in order to identify differences. Instead, it becomes necessary to identify possibly suspicious behavior in a design. This means that the IP design needs to be thoroughly analyzed for possible malicious functionality. Thus, the most significant vector to attacking a circuit during the design stage comes through the inclusion of third-party IP in a design. Most organizations cannot afford to re-invent solutions every time a common component is used, and therefore rely on IP vendors that supply design-modules to perform the desired functionality. The organization can save money and time while avoiding the issue of creating the design from scratch. Designers will instead assemble licensed design modules in order to meet the design specification, often treating the third-party IP as black boxes. These unknown designs can easily make their way unmodified into a final design, allowing for an effective vector for compromising a circuit.

Suppose that a designer were to license a cryptographic circuit for use within a design. The cryptographic block’s encryption could be easily undermined if it were to possess an extra hidden key. While it would appear to function correctly under normal use, someone with knowledge of the hidden key could easily circumvent any security provided by the cryptographic block within the final design. Another risk with third-party IP is that there are a plethora of vendors supplying designs for every possible function, with very little oversight. Vendors come and go, often only possessing an online presence. It would not be difficult for a malicious agent to create a fake vendor persona, and supply malicious design modules at a below-market fee. Compounding the problem is the continuous issue of stolen IP design modules. Vendors sometimes have their IP stolen and resold by other vendors, or even just stolen by designers wanting to use the IP for free. Unfortunately, this has led to a culture of obfuscation and suspicion, making it difficult to get clean, non-obfuscated code in order to identify possible attacks.

5 Accelerated Testing with FPGAs

Although FPGAs exhibit vulnerabilities to the insertion of malicious hardware, they do offer the potential to assist with detecting threats within a design. FPGAs could be used in fault injection campaigns to identify suspected behavior within a design. The potential test vector space is very large, when considering: (1) the number of input vectors (2) the number of fault locations, and (3) the current state for a particular cycle of operation. Emulation in hardware would require less time than using traditional simulation tools [42]. FPGA hardware can also be used to perform the testing in an automated manner. Figure 3 shows a test setup to measure the power drawn for a design under test (DUT). The DUT is a Xilinx BASYS2 FPGA development board, and the I/O is supplied by an Altera DE2 FPGA development board.

6 Summary and Future Work

Unfortunately, detecting malicious hardware within a reconfigurable computing system is an exceedingly difficult task. Inactive Trojans can have an exceedingly small impact on a circuit in terms of area and power, and Trojans are statistically unlikely to be triggered on accident. Stealth is also a key requirement of malicious hardware. A reliance on third-party IP offers a direct path for the insertion of malicious hardware. The very nature of reconfigurability with FPGAs opens the door for security vulnerabilities. Despite the evident need for detecting such changes to a circuit design, there is currently no simple solution to this problem. Many
methods wait until after a chip is fabricated. One alternative is to take samples of the lot for extensive analysis. However, examining the die is becoming increasingly more difficult as transistors decrease in size. Even with an expensive imaging procedure, it would not be possible to test every chip ordered, as imaging may require the destruction of the chip. Other techniques involve detecting changes in the electric current drawn from extra circuitry on the chip, or detecting timing differences due to the additional circuitry on the chip. These methods rely upon the characterization of a golden copy in their comparison, but this trustworthy copy is not available if the original design was compromised, or the parameters could be masked due to process variation on the IC. This paper described the key research challenges for identifying malicious hardware and the state-of-the-art for detection and verification. Yet, there are still opportunities for research contributions as new application domains emerge. For example, in FPGA-based software-defined radio, a designer must defend against malicious modification during initialization and runtime [48]. In wireless sensor networks, the need security emerges for access/discovery, routing, and information [49]. Hardware/software codesign [50] also offers the potential to include security within the overall design framework to address the linkages among technology, architecture, communication, and applications for trustworthy reconfigurable systems.

7 Acknowledgment

This work was supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244) Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia and United Technologies.

8 References


