Chapter 1
Security and Trust Vulnerabilities in Third-Party IPs

Prabhat Mishra, Mark Tehranipoor, and Swarup Bhunia

1.1 Introduction

Our daily life is intertwined in the fabric of Internet-of-Things (IoTs), a fabric in which the number of connected computing devices exceeds the human population. We anticipate over 50 billion devices to be deployed and mutually connected by 2020 [1]. Security and trust are paramount considerations while designing these systems. Majority of these IoT devices as well as embedded systems are designed using both System-on-Chip (SoC) components and software applications. In order to reduce design cost and meet shrinking time-to-market constraints, SoCs are designed using third-party Intellectual Property (IP) blocks [2]. For example, Fig. 1.1 shows a typical SoC with a processor core, a Digital Signal Processor (DSP), memory (MEM), analog-to-digital (ADC) and digital-to-analog (DAC) converters, communication network, peripherals (I/O), and so on. For the ease of illustration, communication network is shown as an IP block instead of connectivity between IPs and peripherals. In many cases, these IPs are designed by different companies across the globe and travel through long supply chain involving multiple vendors, as shown in Fig. 1.2. Security vulnerability of SoCs at different stages of its life cycle is an important concern to SoC designers, system integrators, as well as to the end users. Over the ages, hardware components, platforms, and supply chains have been considered secure and trustworthy. However, recent discoveries and reports on security vulnerabilities and attacks in microchips and circuits violate this assumption [3].

Growing reliance on reusable hardware IPs, often gathered from untrusted third-party vendors, severely affects the security and trustworthiness of SoC computing platforms [3]. An important emerging concern with the hardware IPs acquired from

P. Mishra (✉) • M. Tehranipoor • S. Bhunia
University of Florida, Gainesville, FL, USA
e-mail: prabhat@ufl.edu; tehranipoor@ece.ufl.edu; swarup@ece.ufl.edu

P. Mishra et al. (eds.), *Hardware IP Security and Trust*,
DOI 10.1007/978-3-319-49025-0_1

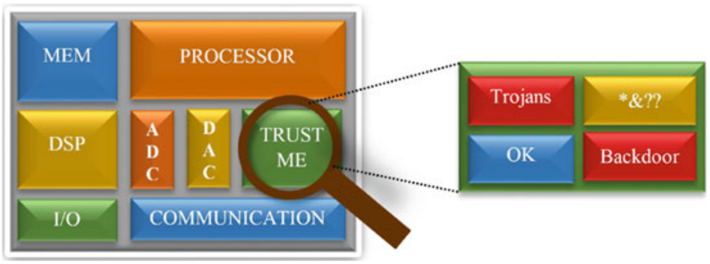

Fig. 1.1 Trusted SoC design using untrusted IPs

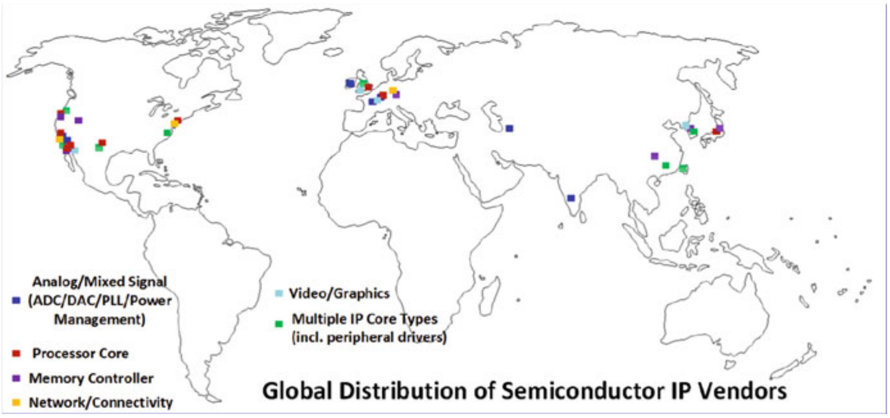
Analog/Mixed Signal (ADC/DAC/PLL/Power Management)
Processor Core
Memory Controller
Network/Connectivity
Video/Graphics
Multiple IP Core Types (incl. peripheral drivers)
Global Distribution of Semiconductor IP Vendors

Fig. 1.2 The long and globally distributed supply chain for hardware IPs makes them increasingly vulnerable to diverse trust/integrity issues

external sources is that they may come with deliberate malicious implants to incorporate undesired functionality (e.g., hardware Trojan), undocumented test/debug interface working as a hidden backdoor, or other integrity issues. For example, a close inspection of an IP (listed as “TRUST ME” in Fig. 1.1) shows malicious logic (hardware Trojan), potential backdoor as well as unknown (unspecified) functionality. Untrusted IPs in an SoC can lead to security consequences for the end user. For example, a hardware Trojan in a processor module can leak the secure key or other protected data to the adversary. In fact, a hardware Trojan in the memory IP would be able to access the secure key from the processor IP if the attacker can carefully analyze the state transitions and able to reach a protected state in the processor IP from any state in the memory IP. It is important to note that an IP from a trusted company may have malicious implants if the IP was outsourced during any of the design stages such as synthesis, verification, test insertion (DFT), layout, fabrication, manufacturing testing, and post-silicon debug. Depending on the type of IP vulnerability, an adversary can not only exploit hardware Trojans, but can also perform side-channel attack, probing attack, FSM vulnerability-based attack, and so on. This book provides a comprehensive coverage of SoC vulnerability analysis and presents effective countermeasures and validation techniques to design trusted SoCs using untrusted third-party IPs.

This chapter is organized as follows. Section 1.2 describes the current practice of SoC design and validation methodology. Section 1.3 outlines security and trust vulnerabilities in third-party IPs. Section 1.4 presents how to design trustworthy SoCs using potentially untrusted third-party IPs. Finally, Sect. 1.5 provides the organization of the remaining chapters.

1.2 Design and Validation of SoCs

System-on-Chip (SoC) integrates both hardware and software components into an Integrated Circuit (IC) that can perform the desired functionality. The hardware components can be processors, co-processors, controllers, converters (analog-to-digital and digital-to-analog), FPGA, memory modules, and peripherals. Similarly, the software components can be real-time operating systems and control programs. A typical SoC can perform digital, analog as well as mixed-signal computations. Figure 1.1 shows a block diagram of a typical SoC design. The example SoC consists of processor, DSP, memory, converters, communication network, and peripherals. SoCs are widely used in designing both IoT devices and embedded systems in a wide variety of domains including communication, entertainment, multimedia, automotive, avionics, and so on. The remainder of this section outlines SoC design methodology followed by SoC validation techniques.

Figure 1.3 shows a typical top-down SoC design methodology. It starts with a design specification, followed by hardware–software co-design (includes partitioning, code generation for software models, and synthesis, layout, and fabrication for hardware models), and integration and verification. Some semiconductor companies use the platform-based SoC design, which is a variation of the traditional top-down methodology. In platform-based methodology, desired behavior (specification) is mapped to the target platforms. Some parts of the design behavior may be available as IPs, whereas the remaining parts need to be implemented in hardware or software. Due to increasing SoC design complexity coupled with reduced time-to-market constraints, semiconductor companies have opted for IP-based SoC design methodology. This enables them to put together all the required components very quickly (compared to top-down design methodology) followed by integration and verification as outlined in Fig. 1.4.

SoC validation is a major challenge due to combined effects of increasing design complexity and reduced time-to-market. Recent studies report that SoC validation consumes up to 70 % resources (cost and time) of overall SoC design methodology [4]. This cost computation includes system-level validation of architectural models, pre-silicon validation as well as post-silicon validation of both functional behaviors and non-functional requirements. Simulation is widely used form of SoC validation using random, constrained random as well as directed test vectors. Since the total

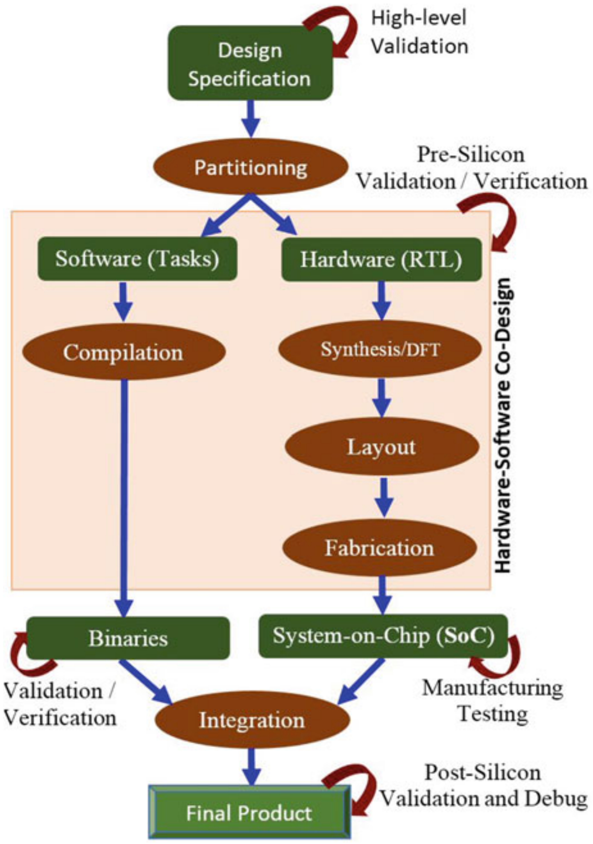

Fig. 1.3 Traditional top-down SoC design methodology

number of possible input combinations (test vectors) can be exponential, simulation-based approaches cannot guarantee completeness of verification. On the other hand, formal methods, such as model (property) checking, equivalence checking, and satisfiability (SAT) solving, can prove correctness but cannot handle large designs due to state-space explosion. SoC verification experts use an efficient combination of simulation-based techniques and formal methods. In a typical industrial setting, full SoC is simulated using efficient test vectors, whereas some of the critical components (e.g., security IPs and complex protocols) are verified using formal methods. Unfortunately, these validation techniques are not designed to detect security and trust vulnerabilities of SoCs [5–8].

1.3 Security and Trust Vulnerabilities in Third-Party IPs

Hardware IPs acquired from untrusted third-party vendors can have diverse security and integrity issues. An adversary inside an IP design house involved in the IP design process can deliberately insert a malicious implant or design modification

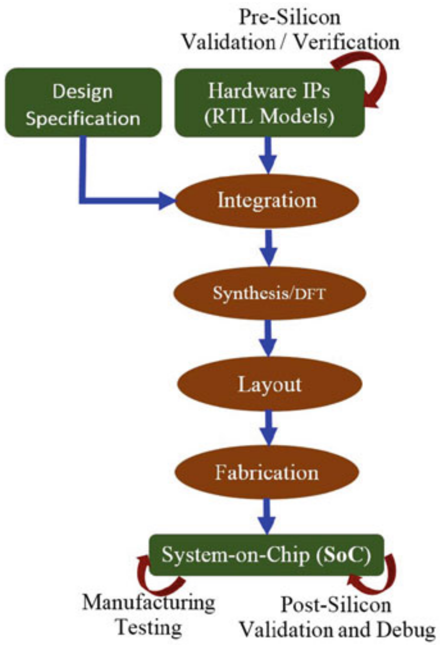

Fig. 1.4 IP-based SoC design methodology

to incorporate hidden and undesired functionality. Such additional functionalities can serve broadly two purposes for an adversary: (1) it can cause malfunction in the SoC that integrates the IP; and (2) it can facilitate information leakage through a hardware backdoor that enables unauthorized access or can directly leak secret information (e.g., cryptographic key, or internal design details of an SoC) from inside a chip. Wide array of hardware Trojan designs including different forms of combinational and sequential Trojans investigated by researchers, with payloads causing malfunction or information leakage, would form a possible arsenal for an adversary. An information leakage or hidden backdoor can give an attacker illegal access to important systems or valuable data inside SoC during in-field deployment of a system. Such an access can potentially be gained remotely through a hardware backdoor. On the other hand, sudden malfunction of an electronic component triggered by a malicious modification can allow an attacker to cause catastrophic consequences in critical infrastructure. Both types of effects constitute critical concerns with respect to national and personal security.

In addition to deliberate malicious changes in a design, IP vendors can also unintentionally incorporate design features, e.g., hidden test/debug interfaces that can create critical security loopholes. In 2012, a breakthrough study by a group of researchers in Cambridge revealed an undocumented silicon level backdoor in a highly secure military-grade ProAsic3 FPGA device from MicroSemi (formerly Actel) [9]. Similarly, IPs can have uncharacterized parametric behavior (e.g., power/thermal) which can be exploited by an attacker to cause irrecoverable damage to an electronic system. In a recent report, researchers have demonstrated such

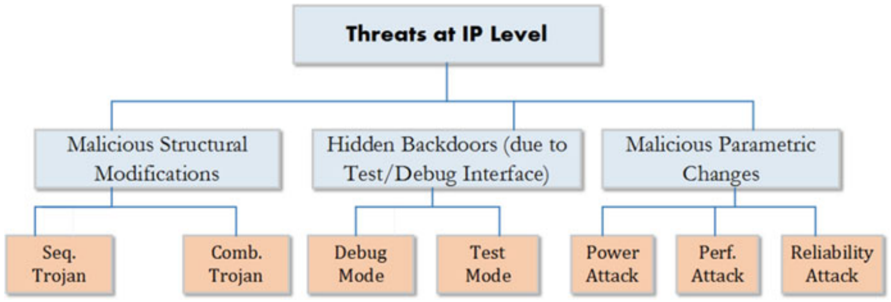
Threats at IP Level
Malicious Structural Modifications
Hidden Backdoors (due to Test/Debug Interface)
Malicious Parametric Changes
Seq. Trojan
Comb. Trojan
Debug Mode
Test Mode
Power Attack
Perf. Attack
Reliability Attack

Fig. 1.5 The taxonomy of trust issues for hardware IPs

an attack where a malicious upgrade of a firmware destroys the processor it is controlling by affecting the power management system [10]. It manifests a new attack mode for IPs, where firmware/software update can maliciously affect the power/performance/temperature profile of a chip to either destroy a system or reveal secret information through appropriate side-channel attack, e.g., a fault or timing attack [11]. Hence, there is a critical need to perform rigorous and comprehensive analysis of trust/integrity issues in hardware IPs and develop effective low-cost countermeasures to protect SoC designers as well as end users against these issues.

Digital IP comes in several different forms: (1) soft IP (e.g., register transfer level IP), (2) firm IP (e.g., gate level IP), and (3) hard IP (e.g., layout level IP). All forms of IPs are vulnerable to the above-mentioned trust issues. Figure 1.5 shows the major classes of trust/integrity issues for hardware IPs. First, an adversary can implant combinational and sequential Trojans, which are triggered by internal node conditions. It is expected that an intelligent adversary would make the trigger condition of these Trojans rare, to evade detection during regular functional testing [12]. A combinational Trojan is activated by a trigger condition, which is a Boolean function of internal node values. On the other hand, a sequential Trojan is activated by a sequence of rare conditions and is realized with a state machine comprising of one or more state elements. In general, these Trojans can have two types of payloads (which indicate the effect of Trojan): either malfunction or information leakage, as mentioned earlier. The latter can be accomplished by leaking secret information either through output ports as logic values or through side-channel information leakage, i.e., adding single bit of information at a time to the power signature to leak the key from a crypto IP [13]. Both types of Trojans can have varying complexity, which are expected to correlate well with the triggering probability. A more complex combinational (or sequential) Trojan is likely to use more nodes (trigger inputs) to realize a trigger condition and hence a more rare activation condition.

The second class of trust/integrity issues would relate the apparently benign test/debug interface which makes IPs vulnerable to diverse security attacks during field operation. For example, a scan chain in crypto IP that interfaces with key-dependent logic, if accessible postfabrication in an SoC, can enable mounting a scan-based controllability/observability attack leading to leakage of the secret key. Similarly, debug interface in an SoC that allows post-silicon validation, debug, and failure analysis is well-known vulnerability for diverse attacks. A recent incident points to such a vulnerability [9]. A study revealed an undocumented silicon level backdoor in a highly secure military-grade reconfigurable device from MicroSemi. This backdoor had a key or passcode as Microsemi puts it, which was extracted in affordable time and effort by the researchers using a highly sensitive differential power analysis technique. With this backdoor, an attacker has access to different chip components like non-volatile memory (NVM), array bitstream, etc. An attacker can utilize this to remotely reprogram as well as disable the chip even if a user locks it with his/her own user key. Microsemi has confirmed the presence of this structure as a test/debug interface with JTAG, controllable, and usable only by the manufacturer. The concern is that they cannot be fixed by software antivirus and hence those chips which are deployed remain susceptible. It is important to analyze if such a test/debug interface creates either direct (logic-based) or indirect (side-channel) vulnerability in terms of information leakage.

The third class of Trojan relates to hidden input/operating conditions that can bring an IP outside safe operating points, such as attacks where an IP violates the peak power or temperature constraint. Since many third-party IPs may not be characterized well with their parametric behavior, it makes them vulnerable with respect to attacks where an attacker tries to exercise input (possibly rare) conditions to violate non-functional constraints. Such a condition can impose serious threat where the firmware/software/input can be used to directly attack the hardware, as reported in [10]. This book presents analysis of major types of attacks at the IP level and their implications in SoC security and trust.

1.4 Trustworthy SoC Design Using Untrusted IPs

Recent research efforts have targeted IP trust assurance. These efforts can be classified into two broad categories: (1) test and verification solutions, where a set of test patterns aim at identifying unknown hidden Trojans in an IP; and (2) design solutions, where SoC designers follow a pre-defined protocol with the IP providers. Existing works on test/verification solutions have considered both statistical and directed test generation. The scalability of the approach to large designs or various forms of IPs and Trojans has not been extensively studied. A salient work in the second category is based on embedding proof-carrying-code (PCC) into a design, where the SoC designer provides a set of proofs to an IP vendor for embedding into the IP, which can then be checked by the SoC designer for integrity verification. While such a technique can be effective to provide high level of assurance to SoC designer on certain properties and components of an IP, it cannot provide comprehensive assurance in all structural components of an IP. An attacker of the IP can evade the protection of PCCs by incorporating a Trojan in logic blocks not covered by them. Furthermore, it may incorporate considerable design overhead due to integration of the PCCs and imposes a significant requirement and binding for the SoC designer to interact with an IP vendor during the IP design process.

Due to their stealthy nature and practically infinite Trojan space (in terms of trigger conditions, forms, and sizes), a cleverly inserted Trojan in an IP integrated into a large SoC is highly likely to evade conventional post-silicon testing. Furthermore, malicious hardware can easily bypass traditional software-implemented defense techniques as it is a layer below the entire software stack. Similarly, an apparently benign debug/test interface or uncharacterized parametric behavior under rare input conditions can be exploited by an adversary during field usage. Existing security analysis and trust validation at IC level do not readily extend to hardware IPs used in an SoC due to the following reasons.

- Different Threat Model: Trust/integrity issues at IP level fall into a fundamentally different class than those at IC primarily because third-party IPs do not come with golden reference models. It prevents performing conventional sequential equivalence check (SEC) to identify an anomaly. Most often these IPs come with simply a functional specification (often incomplete) and optionally high-level block diagrams. On the other hand, for IC, either a set of golden ICs or a golden design at some abstraction level is used as a reference.
- Hidden Test/Debug Interface: Unlike ICs, where test/debug interfaces that go into a design are available to test engineers during trust validation, IPs can include hidden test/debug interface that can create major security issue, as discovered recently.
- Parametric Vulnerability: Power/temperature/performance behavior of an IP may not be adequately verified by an IP vendor, which can create unanticipated vulnerability, as mentioned earlier.

Fortunately, an IP of any form provides an important advantage with respect to trust verification. It is not really a black box and one can exploit the structural/functional information about the constituent logic blocks (e.g., datapath) to efficiently identify diverse trust issues. Three major types of IP trust validation solutions, as illustrated in Fig. 1.6, are detailed below.

Functional Analysis The functional analysis can be performed in two steps. First, it can employ a statistical random testing approach, which aims at activating arbitrary Trojan instances. Published results in this area show that such an approach can be highly effective for small Trojans, particularly combinational ones [14]. It tries to bias a random test generator toward possible malicious implants. However, such an approach generally is not effective for larger Trojans whose activation conditions are random-resistant. Statistical random tests will be complemented with directed tests.

Structural Analysis It is important to ensure that an IP block performs exactly as intended in the specification (nothing more, nothing less), which requires evaluation of both state machine and datapath. To check the integrity of the state transition function, one can employ a self-referencing approach [15], which compares the temporal behavior of a finite state machine (FSM) at two different time instants

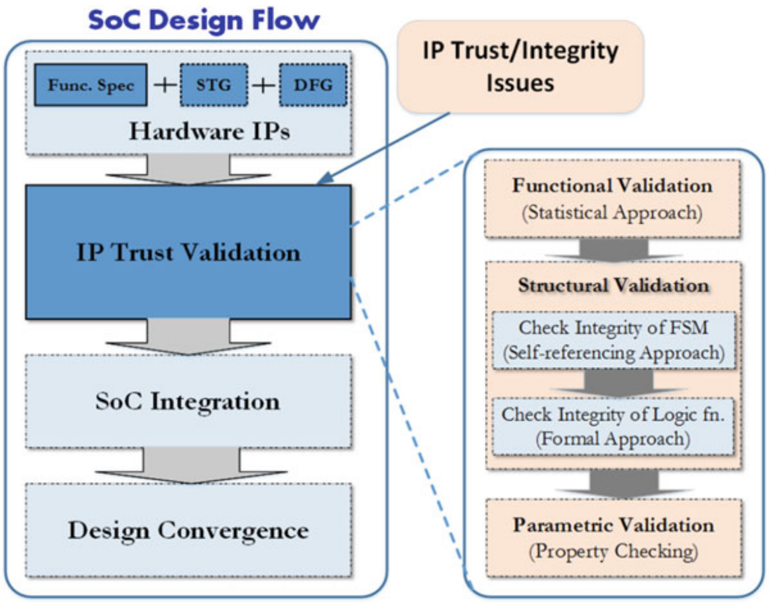
SoC Design Flow
Func. Spec
STG
DFG
Hardware IPs
IP Trust Validation
SoC Integration
Design Convergence
IP Trust/Integrity Issues
Functional Validation (Statistical Approach)
Structural Validation
Check Integrity of FSM (Self-referencing Approach)
Check Integrity of Logic fn. (Formal Approach)
Parametric Validation (Property Checking)

Fig. 1.6 Different classes of IP trust validation approaches which can potentially be integrated for higher level of confidence

to identify malicious modification of the FSM. The basic idea is that a sequential Trojan (e.g., a counter Trojan) in a design would trigger uncorrelated activities in the state machine and hence can be isolated. Self-referencing will be complemented with a formal verification approach. Formal verification methods make use of mathematical techniques to guarantee the correctness of a design with respect to some specified behavior. Unfortunately, existing methods often fail for large IP blocks due to state-space explosion during equivalence checking unless the structure of specification and implementation are very similar. Hence, new scalable formal methods for structural trust verification of IP need to be developed [7].

Parametric Analysis It is also important to check various properties of IPs including peak power and peak temperature for trust assurance. An attacker can generate a power virus (through software/firmware/input control) that can maximize the switching activity in the IP. In case the peak power violates the threshold, the designers may need to re-design the IP block to reduce the peak power. An attacker can develop mechanisms to produce a temperature virus that can generate sustained peak power to produce peak temperature. In case the temperature virus can increase the IP temperature beyond threshold, it can lead to reliability, security, and safety concerns.

Secure by Design As an alternative or complementary approach to IP level trust validation, judicious design solutions can be used to protect SoCs from malicious behavior of untrusted IPs. Such a solution would often require collaboration between SOC design house and IP vendors. The central idea for such secure by design solution is to design an IP with specific provable properties whose trustworthiness can be easily established during SoC integration. An SoC designer would ask an IP vendor to embed a set of pre-defined verifiable security assertions into an IP which can be checked—e.g., using formal methods, to verify the trust level of the IP. Any violation of the security assertions/checks would indicate the IP being non-compliant and hence potentially malicious. Major challenges with this approach include derivation of appropriate security checks for different functional blocks in a design, the design overhead due to insertions of security checks, and finally, difficulty to safe-guard the entire design. Besides, such an approach requires change in the current business model where IPs are sold to chip manufacturers as commodity products with little or no customization.

1.5 Book Organization

This book is organized as follows. Chapters 2–5 present several trust vulnerability analysis techniques. Chapters 6–7 provide effective countermeasures against various forms of attacks. Chapters 8–14 survey efficient techniques for testing and validation of hardware IP security and trust. Finally, Chap. 15 concludes the book.

- *Chapter 2* presents a framework to analyze design vulnerabilities at different abstraction levels and assess its security at design stage.
- *Chapter 3* describes vulnerability analysis for both gate- and layout-level designs to quantitatively determine their susceptibility to hardware Trojan insertion.
- *Chapter 4* presents an interesting case study to identify suspicious signals using both formal and semi-formal coverage analysis methods.
- *Chapter 5* surveys existing techniques in performing probing attacks, the problem of protection against probing attacks, and presented a layout-driven framework to assess designs for vulnerabilities to probing attacks.
- *Chapter 6* reviews three major security hardening approaches—camouflaging, logic encryption/locking, and design obfuscation—that are applied to ICs at layout, gate, and register transfer levels.
- *Chapter 7* presents a mutating runtime architecture to support system designers in implementing cryptographic devices hardened against side-channel attacks.
- *Chapter 8* surveys the existing security validation methods for soft IP cores using a combination of simulation-based validation and formal methods.
- *Chapter 9* utilizes model checking and theorem proving using proof-carrying hardware for trust evaluation.
- *Chapter 10* describes three methods to detect potential hardware Trojans by utilizing Trojan characteristics. It also outlined how a stealthy Trojan can evade these methods.

- *Chapter 11* outlines how to exploit unspecified functionality for information leakage and presented a framework for preventing such attacks.
- *Chapter 12* proposes mechanism for specifying security properties and verifying these properties across firmware and hardware using instruction-level abstraction.
- *Chapter 13* describes how to perform test generation for detecting malicious variations in parametric constraints.
- *Chapter 14* covers side-channel testing using three metrics, along with practical case studies on real unprotected and protected targets.
- *Chapter 15* concludes the book and provides a future roadmap for trustworthy SoC design.

References

1. D. Evans, Cisco white paper on the internet of things: how the next evolution of the internet is changing everything (April 2011). www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf. Accessed 8 Nov 2016
2. M. Keating, P. Bricaud, *Reuse Methodology Manual for System-on-a-Chip Designs* (Kluwer, Norwell, MA, 1998)
3. S. Bhunia, D. Agrawal, L. Nazhandali, Guest editors' introduction: trusted system-on-chip with untrusted components. IEEE Des. Test Comput. **30**(2), 5–7 (2013)
4. M. Chen, X. Qin, H. Koo, P. Mishra, *System-Level Validation: High-Level Modeling and Directed Test Generation Techniques* (Springer, New York, 2012)
5. Y. Huang, S. Bhunia, P. Mishra, MERS: statistical test generation for side-channel analysis based Trojan detection, in *ACM Conference on Computer and Communications Security (CCS)* (2016)
6. X. Guo, R. Dutta, P. Mishra, Y. Jin, Scalable SoC trust verification using integrated theorem proving and model checking, in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2016), pp. 124–129
7. F. Farahmandi, Y. Huang, P. Mishra, Trojan localization using symbolic algebra, in *Asia and South Pacific Design Automation Conference (ASPDAC)* (2017)
8. X. Guo, R. Dutta, Y. Jin, F. Farahmandi, P. Mishra, Pre-silicon security verification and validation: a formal perspective, in *ACM/IEEE Design Automation Conference (DAC)* (2015)
9. S. Skorobogatov, C. Woods, Breakthrough silicon scanning discovers backdoor in military chip, in *CHES 2012*. Published in Lecture Notes in Computer Science, vol. 7428 (2012), pp. 23–40
10. Ellen Messmer, RSA security attack demo deep-fries Apple Mac components, Network World (2014). http://www.networkworld.com/article/2174737/security/rsa-security-attack-demo-deep-fries-apple-mac-components.html
11. P.C. Kocher, Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems, in *CRYPTO* (1996), pp. 104–113
12. F. Wolff, C. Papachristou, R. Chakraborty, S. Bhunia, Towards Trojan-free trusted ICs: problem analysis and a low-overhead detection scheme, in *Design Automation and Test in Europe (DATE)* (2008)
13. L. Lin, W. Burleson, C. Parr, MOLES: malicious off-chip leakage enabled by side-channels, in *International Conference on CAD (ICCAD)* (2009)

14. R. Chakraborty, F. Wolff, S. Paul, C. Papachristou, S. Bhunia, MERO: a statistical approach for hardware Trojan detection, in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)* (2009)
15. S. Narasimhan, X. Wang, D. Du, R. Chakraborty, S. Bhunia, TeSR: a robust temporal self-referencing approach for hardware Trojan detection, in *4th IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2011)