Chapter 15
The Future of Trustworthy SoC Design

Prabhat Mishra, Swarup Bhunia, and Mark Tehranipoor

15.1 Summary

This book provided a comprehensive coverage of IP security and trust issues with contributions from academic researchers, SOC designers as well as SoC verification experts. The topics covered in this book can be broadly divided into the following three categories.

15.1.1 Trust Vulnerability Analysis

Chapter 1 highlighted how security of IP can be compromised at various stages in the overall SoC design-fabrication-deployment cycle affecting various parties. This book presented five efficient techniques for trust vulnerability analysis.

- *Security Rule Check:* Chap. 2 presented a framework to analyze design vulnerabilities at different abstraction levels and assessed its security at design stage.
- *Vulnerability Analysis at Gate and Layout Levels:* Chap. 3 described vulnerability analysis for both gate- and layout-level designs to quantitatively determine their susceptibility to hardware Trojan insertion.
- *Code Coverage Analysis:* Chap. 4 presented an interesting case study to identify suspicious signals using both formal and semi-formal coverage analysis methods.
- *Layout Analysis for Probing Attack:* Chap. 5 surveyed existing techniques in performing probing attacks, protection against probing attacks, and presented a layout-driven framework to assess designs for vulnerabilities to probing attacks.

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

P. Mishra et al. (eds.), *Hardware IP Security and Trust*,
DOI 10.1007/978-3-319-49025-0_15

- *Side Channel Vulnerability:* Chap. 6 covered side channel testing using three metrics as well as practical case studies on real unprotected and protected targets.

15.1.2 Effective Countermeasures

The next two chapters provided effective countermeasures against various attacks. The goal of these approaches is to make it hard for the attacker to introduce vulnerability.

- *Camouflaging, Encryption, and Obfuscation:* Chap. 7 reviewed three major security hardening approaches—camouflaging, logic encryption/locking, and design obfuscation—that are applied to ICs at layout, gate, and register transfer levels.
- *Mutating Runtime Architecture:* Chap. 8 presented a mutating runtime architecture to support system designers in implementing cryptographic devices hardened against side channel attacks.

15.1.3 Security and Trust Validation

The final six chapters presented efficient techniques for validation of IP security and trust vulnerabilities.

- *Validation of IP Trust:* Chap. 9 surveyed the existing security validation methods for soft IP cores using a combination of simulation-based validation and formal methods.
- *Proof-Carrying Hardware:* Chap. 10 utilized model checking and theorem proving using proof-carrying hardware for trust evaluation.
- *Trust Verification:* Chap. 11 described three methods to detect potential hardware Trojans by utilizing Trojan characteristics. It also outlined how a stealthy Trojan can evade these methods.
- *Unspecified IP Functionality:* Chap. 12 outlined how to exploit unspecified functionality for information leakage and presented a framework for preventing such attacks.
- *Security Property Validation:* Chap. 13 proposed mechanism for specifying security properties and verifying these properties across firmware and hardware using instruction-level abstraction.
- *Malicious Parametric Variations:* Chap. 14 described how to perform test generation for detecting malicious variations in parametric constraints.

These chapters provided a comprehensive coverage of hardware IP trust analysis as well as effective trust validation techniques to enable secure and trustworthy SoC design.

15.2 Future Directions

Although significant research has been carried out over the past decade for securing IPs, there are still many challenges ahead, especially as the IC and IP supply continuously change. For example, the IP vendors are shifting towards using encryption to protecting their IPs from piracy. This would make logic simulation, trust verification, and integration with other IPs in an SoC more difficult. Furthermore, addressing one security issue may create unwanted new vulnerabilities to some other security concerns. We briefly outline some of the challenges ahead in verifying security and trustworthiness of future IPs.

15.2.1 Security and Trust Verification for Encrypted IPs

Recent trends in IP piracy have raised serious concerns to IP developers. IP piracy can take several forms, for example, an untrusted SoC designer may legally purchase a third party IP (3PIP) core from an IP vendor and then make illegitimate copies of the original IP and sell them under their own name [4]. The SoC designer may also add some extra features to the original IPs to make them look like a new one and then sell them to another SoC designer for making easy profit. To prevent IP piracy, the IP developers are increasingly adopting the IEEE P1735 encryption scheme developed by Design Automation Standards Committee of the IEEE [7]. Most EDA tools also support IEEE P1735 standard which utilizes two levels of encryption to produce an encrypted IP core [8]. IEEE P1735 standard ensures that the IP in plaintext format is never exposed to the SoC integrator while allowing the EDA tools to perform functional simulation, synthesis, etc., of the encrypted IP core.

Unfortunately, IEEE P1735 encryption scheme introduces new challenges and complications for IP trust verification and security rule check by restricting the SoC designer to analyze the RTL code. This would prohibit the SoC integrator from applying some of the IP trust verification techniques proposed in this book as well as in the literature. For example, IP trust verification techniques proposed in [1–3, 5, 6, 12] cannot be applied on encrypted IPs as they require analysis of RTL code which needs to be in plaintext format. The academic research community should focus their effort to ensure that IP trust verification is possible even when the IP is given in an encrypted format. One possible solution is to investigate IP trust verification techniques which work on the gate-level netlist. The reason is that most IP providers allow the visibility of the gate-level netlist in unencrypted format according to IEEE P1735 standard.

15.2.2 Security and Trust Verification for Obfuscated IPs

Logic obfuscation approach does not address all the issues associated with IP piracy. The untrusted SoC integrator can add some additional functionality to the obfuscated IP and sell it to other SoC integrators as a firm IP. For example, an untrusted SoC integrator may purchase an encryption engine from an IP provider in encrypted format. Then, the SoC integrator can include hashing functionality with the encryption engine and synthesize them to gate-level netlist (unencrypted format). The untrusted SoC integrator can then illegitimately sell it as a firm IP which can perform encryption and hashing operation to other SoC designers under its own name. To address this issue, academic researchers have proposed to obfuscate and functionally lock the IP [10, 11]. This approach works by placing locking gates (XOR/ XNOR) into the design. The IP produces functionally correct output when it receives the correct chip unlock key. The obfuscation approaches have gained interest in the industry and are expected to be included in design flows in the near future.

The IP obfuscation technique would help address the problem of IP piracy. However, from SoC designer's point of view it would make the IP trust verification extremely challenging. Similarly, logic obfuscation makes it difficult to perform security rule check. To date, no IP trust verification approach has been proposed in the literature to take IP obfuscation into account. Most existing techniques like FANCI [16], rare net identification [13], security rule check [18] for IP trust verification would not work on netlist which is functionally locked. The academic research community should direct their attention and effort to developing new and innovative IP trust verification techniques that can be applied to obfuscated and functionally locked IPs.

15.2.3 Security and Trust Verification for Hard IPs

Another growing concern in the industry is verifying the trust of hard IPs. Most foundries have begun to offer more and more hard IPs to SoC designers. These hard IPs have comparatively lower cost, have a high yield since they have been produced many times before and been tested for high yield, and offer minimal time to market delay. However, these IPs are incorporated in the design at the very last stages of SoC design flow (physical layout). Any analysis done on the layout level would be very complex and time consuming. One possible solution to this problem would be to take inspiration from postsilicon debugging techniques and develop postsilicon security and trust verification techniques.

15.2.4 Security and Trust Verification During SoC Design Flow

It is of utmost importance to identify IP security and trust issues at all levels of abstraction in the SoC design flow and during transition from one stage to the next stage of the design process. In most literature, the 3PIP vendor and the foundry are considered as untrusted [17]. However, there are other entities in the SoC design flow that can maliciously incorporate hardware Trojans in the design or create security vulnerability. For example, many SoC designers outsource the design for test (DFT) insertion task to third party entities. These entities can incorporate malicious circuitry in the design before returning it to the SoC designer. Therefore, it is important for the SoC developers to not only verify trustworthiness of third party IP but also to verify trustworthiness of their design at subsequent levels of abstraction, namely synthesis, DFT insertion, physical layout, etc. Researchers can draw inspiration from different fields in order to address this problem. For example, techniques proposed for IP piracy prevention can be adopted in order to establish trust in the design. The SoC designers can obfuscate and functionally lock their design and restrict other entities in the SoC design flow from making any malicious modifications.

15.2.5 Unintentional Vulnerabilities

Many security vulnerabilities in SoCs could be created unintentionally by CAD tools and by designers' mistakes. CAD tools are extensively used for synthesis, DFT insertion, automatic place and route, etc. However, today's CAD tools are not equipped with understanding security vulnerabilities in SoCs. Therefore, the tools can introduce additional vulnerabilities in the design. For example, during synthesis process, the tool tries to optimize a design for power, area, and/or performance of the design. If there exists any don't-care conditions in the RTL specification, the synthesis tool introduces deterministic values for the don't-care conditions for design optimization. This could facilitate attacks such as fault injection or side channel-based attacks [18]. Another example worthy of mentioning here would be the vulnerabilities introduced by the DFT tool. The inserted DFT can create numerous vulnerabilities by allowing attackers to control or observe internal assets of an SoC. Vulnerabilities in an SoC design can also be introduced by designer's mistakes.

Traditionally, the design objectives are driven by cost, performance, and time-to-market constraints, while security is generally neglected during the design phase. In [9], authors have shown that if a finite state machine is designed without security into consideration, it can introduce vulnerabilities by facilitating fault injection attack. It is of paramount importance to identify these security vulnerabilities during hardware design and validation process. However, given the growing complexity

of modern SoC designs, it is extremely difficult, if not impossible, to manually identify these vulnerabilities. This calls for the development of CAD tools which can automatically evaluate the security of a design in reasonable time without significantly impacting time-to-market. Also to avoid some common security problems in early design stages, security-aware design practices must be developed and used as guidelines for design engineers.

15.2.6 Multi-Security Objectives Design

The evaluation time for IP security and trust verification and finding vulnerabilities in an IP need to be scalable with the design size in order to have low impact on time-to-market. It is therefore important to identify which security analysis methods and countermeasures need to be applied given the application of the target SoC. For example, for a crypto IP, side channel vulnerability analysis and mitigation are recommended. However, for non-crypto IP the side channel leakage may not pose any threat and therefore, side channel vulnerability analysis and mitigation are not required for such IPs. However, if an SoC designer blindly starts applying side channel leakage countermeasures for all IPs, it would not only cause unnecessary area overhead but also could have negative impact on Trojan detection approaches which rely on side channel evaluation. In summary, addressing one security issue may negatively impact another security feature, hence during design process multi-security objectives must be considered. To perform this analysis, a designer needs to select the security vulnerabilities he/she may be concerned about for the target SoC, and the tool should perform an optimization process to ensure highest security for all vulnerabilities.

15.2.7 Metrics and Benchmarks

Finally, a major limitation for the performance evaluation of any IP security and trust verification technique is the inadequate number and quality of benchmark circuits. Benchmarks and metrics are necessary components for establishing baseline for comparison between different techniques developed by the research community. The Trojan benchmark circuits developed in [14, 15] are not adequate to help address some of the challenges raised by the emerging IP trust issues, such as encrypted IPs. There is currently no encrypted IP with Trojan in [15]. Therefore, it is important for the research community to design new and innovative metrics and Trojan benchmark circuits that incorporate these features.

References

1. F. Farahmandi, Y. Huang, P. Mishra, Trojan localization using symbolic algebra, in *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2017
2. 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
3. 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
4. U. Guin, Q. Shi, D. Forte, M.M. Tehranipoor, FORTIS: a comprehensive solution for establishing forward trust for protecting IPs and ICs. ACM Trans. Des. Autom. Electron. Syst. (TODAES) **21**(4), 63 (2016)
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. M. Hicks, M. Finnicum, S.T. King, M.M.K. Martin, J.M. Smith, Overcoming an untrusted computing base: detecting and removing malicious hardware automatically, in *Proceedings of IEEE Symposium on Security and Privacy* (2010), pp. 159–172
7. IEEE Approved Draft Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP) (2014)
8. Microsemi 2014, *Libero SoC Secure IP Flow User Guide for IP Vendors and Libero SoC Users* (2014). http://www.microsemi.com/document-portal/docview/133573-libero-soc-secure-ip-flow-user-guide
9. A. Nahiyan, K. Xiao, K. Yang, Y. Jin, D. Forte, M. Tehranipoor, AVFSM: a framework for identifying and mitigating vulnerabilities in FSMs, in *Design Automation Conference*, 2016
10. M.T. Rahman, D. Forte, Q. Shi, G.K. Contreras, M. Tehranipoor, CSST: preventing distribution of unlicensed and rejected ICs by untrusted foundry and assembly, in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* (2014), pp. 46–51
11. J.A. Roy, F. Koushanfar, I.L. Markov, EPIC: ending piracy of integrated circuits, in *Proceedings of the on Design, Automation and Test in Europe* (2008), pp. 1069–1074
12. H. Salmani, M. Tehranipoor, Analyzing circuit vulnerability to hardware Trojan insertion at the behavioral level, in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* (2013), pp. 190–195
13. H. Salmani, R. Karri, M. Tehranipoor, On design vulnerability analysis and trust benchmarks development, in *Proceedings of IEEE 31st International Conference on Computer Design (ICCD)*, pp. 471–474 (2013)
14. H. Salmani, M. Tehranipoor, R. Karri, On design vulnerability analysis and trust benchmark development, in *IEEE International Conference on Computer Design (ICCD)*, 2013
15. Trust-HUB, http://trust-hub.org/resources/benchmarks
16. A. Waksman, M. Suozzo, S. Sethumadhavan, FANCI: identification of stealthy malicious logic using Boolean functional analysis, in *Proceedings of the ACM Conference on Computer and Communications Security* (2013), pp. 697–708
17. K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, M. Tehranipoor, Hardware Trojans: lessons learned after one decade of research. ACM Trans. Des. Autom. Electron. Syst. **22**(1), article 6 (2016)
18. K. Xiao, A. Nahiyan, M. Tehranipoor, Security rule checking in IC design. Computer **49**(8), 54–61 (2016)