High-Level Methods for Hardware IP Protections: Solutions, Trends, and Challenges

Christian Pilato
Politecnico di Milano - Dipartimento di Elettronica, Informazione e Bioingegneria
christian.pilato@polimi.it

***Abstract*—Due to the globalization of the semiconductor supply chain, the security threats for the production of an integrated circuit (IC) and its intellectual property (IP) are becoming more and more critical for many fab-less design houses. Conversely, the protections for these threats are expensive, especially when introduced in the last stages of the design flow. In this paper, we discuss the approaches, the trends, and the associated challenges that can be applied in the early stages of the design, i.e., before logic synthesis. On one hand, these approaches can operate on more semantic information and offer more protection. On the other hand, they have more effects on the overall design and need to somehow "predict" the effects on the final implementation.**

I. Introduction

The increasing costs of the manufacturing process for deep sub-micron technologies are pushing many semiconductor companies to outsource the integrated circuit (IC) fabrication to third-party foundries [1], as shown in Figure 1. This trend has several advantages. *Fab-less* companies can focus their investments on the design process, while the foundries can mitigate the costs by serving multiple customers. However, globalizing the semiconductor supply chain can also lead to security issues [2], [3]. A malicious actor that has access to the IC design can reverse engineer the functionality, steal the intellectual property (IP), and sell illegal IC copies [4]. Many design companies are thus looking for security countermeasures for hardware IP protection. Existing techniques often operate on the gate-level netlist, which is the result of logic synthesis, or directly at the manufacturing level. For example, logic locking and split manufacturing are popular techniques applied at this level. However, this approach has two major drawbacks. First, it requires an intimate knowledge of the target technology and the design, limiting the application to small chips. For example, the designers must have access to the technology libraries and the full IC specification to understand where to apply the protections. Second, logic synthesis performs several optimizations that embed semantic information into the design. For example, a multiplication for a constant can trigger optimizations that can reduce it even to a simple left shifter. While this optimization improves area and timing, it clearly reveals the operation that is executed. An alternative approach that is gaining attention is to operate at higher levels of abstraction, i.e., before logic synthesis.

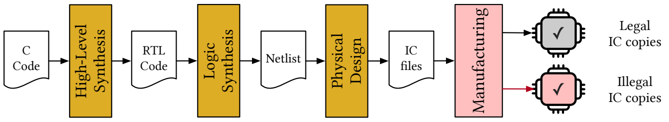

Fig. 1. IC design flow with security threats

These high-level approaches allow designers to operate on and protect more semantic information. In this context, *high-level synthesis* (HLS) plays a critical role in the generation of the hardware description to be synthesized. Protection methods can be directly applied to the chip's specification (e.g., the input C code), during the HLS steps, or directly on the resulting register-transfer level (RTL) description. However, not all of them are compatible with current industrial EDA flows, demanding innovations in tools, metrics, and tool flows.

In the rest of this paper, we first describe the threat models that are usually considered for hardware IP protection (Section II). Then, Section III presents the current solutions and trends, while Section IV discusses open challenges.

II. Threat Model

The *threat model* defines the goals and capabilities of the attackers. In this paper, we consider attackers that aim at reverse engineering a critical functionality to steal the corresponding IP. The entire IC or part of it can be later copied into an illegal design, fabricated, and sold. This process creates significant economic damage to the original design house.

We assume the attackers can access the IC design files and can reverse engineer the functionality from GDSII to gates and then up to RTL [5], [6]. Once they have the RTL description, they can perform extensive simulations or re-synthesize the design to see the effects of logic optimizations. They may have access also to a working version of the chip (called *oracle*), for example, because they found it on the (black) market. The oracle chip can be used to analyze the I/O relationships [7].

III. Overview of Current Solutions and Trends

In the following, we discuss hardware IP protection methods that can be implemented at higher levels of abstraction, i.e., ranging from component specification to system-level design.

A. Watermarking

Watermarking is a *passive* protection method, i.e. it helps identify but cannot actively prevent an illegal IC copy [8]. It operates by embedding a secret *signature*, i.e., a unique sequence of information, into the design. The subsequent

978-1-6654-9885-2/22/$31.00 ©2022 IEEE

verification of the signature can be used to claim the ownership of a chip design during litigation. On one hand, the signature must be unique (very extremely low probability of collision - i.e., of generating the same one by chance) but easy to be verified. High-level watermarking methods aim at inserting constraints into the HLS scheduling and binding steps to create a unique solution [9] or reusing the functional resources to implement a unique watermarking function [10]. Note that, in the latter case, the design also requires extra logic (similar to a scan-chain) to be able to functionally verify the signature.

B. Locking Locking

Logic locking is the process of inserting extra logic in the circuit controlled by a newly-inserted input (called *locking key*). The locking key is not given to the foundry but later installed into a tamper-proof memory to "activate" the functionality of the chip. While logic locking has been mostly applied at the gate level [11], there is a growing interest for raising the abstraction level to RTL or even more the previous stages. In these cases, this process is usually called behavioral locking and can be applied to the high-level specification (e.g., C/C++) [12], [13], during the HLS (even though it may require tool changes) [14], [15], or on the RTL descriptions [16]. In all cases, it is possible to protect more semantic information (e.g., proprietary constants, arithmetic operations, control flow) and use industrial EDA flows for chip design [17]. Like in the case of gate-level locking, the locking key must be kept secret.

C. FPGA Redaction

To further protect the hardware IP, designers can remove sensitive modules from the design and replace them with (e)FPGAs. In this case, the protection guarantees come from the concept of field-programmability of the devices. So, the foundry cannot know in advance which functions will be implemented on the devices. In these cases, the designers have to trade off security concerns [18] and EDA challenges [19], [20] to determine the best system-level architectures in terms of the number of FPGA devices and the corresponding parameters in case of custom instances.

IV. Open Challenges for High-Level IP Protection

While high-level approaches are promising, there are several open EDA and security challenges. First, the cost of protection methods is generally high, preventing many companies from adopting them. Second, the evolution of the corresponding attacks demands even more efficient methods. For example, SAT attacks are the standard *de-facto* for breaking and recovering the secret key by analyzing I/O relationships [7]. These approaches are becoming more and more sophisticated, including also machine-learning attacks [21]. Future methods need to be designed with clear security metrics and better integrated into synthesis tools.

V. Conclusions

This paper presented an overview of the high-level methods for hardware IP protection. In particular, it discussed the approaches that can be applied before logic synthesis, including watermarking, logic locking, and eFPGA redaction. While these approaches are effective to protect more semantic information, there are still several open challenges from both the security and EDA viewpoints.

Acknoledgements

The author would like to thank R. Karri (NYU), S. Garg (NYU), L. Collini (NYU), and B. Tan (University of Calgary) for the collaborations and useful discussions over the years.

References

[1] J. Hurtarte, E. Wolsheimer, and L. Tafoya, *Understanding Fabless IC Technology*. Elsevier, Aug. 2007.

[2] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *IEEE/ACM ICCAD*, 2013.

[3] W. Hu *et al.*, "An overview of hardware security and trust: Threats, countermeasures, and design tools," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 40, no. 6, 2021.

[4] U. Guin *et al.*, "Counterfeit Integrated Circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.

[5] R. S. Rajarathnam, Y. Lin, Y. Jin, and D. Z. Pan, "ReGDS: A Reverse Engineering Framework from GDSII to Gate-level Netlist," in *IEEE HOST*, 2020, pp. 154–163.

[6] J. Rajendran *et al.*, "Belling the CAD: Toward security-centric electronic system design," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1756–1769, Nov. 2015.

[7] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *IEEE HOST*, 2015, pp. 137–143.

[8] A. T. Abdel-Hamid *et al.*, "A survey on IP watermarking techniques," *Design Automation for Embedded Syst.*, vol. 9, no. 3, pp. 211–227, 2004.

[9] A. L. Oliveira, "Techniques for the creation of digital watermarks in sequential circuit designs," vol. 20, no. 9, pp. 1101–1117, Sep. 2001.

[10] C. Pilato *et al.*, "High-level synthesis of benevolent Trojans," in *ACM/IEEE DATE*, 2019, pp. 1124–1129.

[11] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP protection and supply chain security through logic obfuscation: A systematic overview," *ACM TODAES*, vol. 24, no. 6, pp. 1–36, 2019.

[12] M. R. Muttaki, R. Mohammadivojdan, M. Tehranipoor, and F. Farahmandi, "HLock: Locking IPs at the high-level language," in *ACM/EDAC/IEEE DAC*, 2021, pp. 79–84.

[13] N. Veeranna and B. C. Schafer, "Efficient behavioral intellectual properties source code obfuscation for high-level synthesis," in *IEEE LATS*, March 2017, pp. 1–6.

[14] C. Pilato, F. Regazzoni, R. Karri, and S. Garg, "TAO: Techniques for algorithm-level obfuscation during high-level synthesis," in *ACM/EDAC/IEEE DAC*, 2018, pp. 1–6.

[15] C. Pilato, L. Collini, L. Cassano, D. Sciuto, S. Garg, and R. Karri, "Optimizing the use of behavioral locking for high-level synthesis," *IEEE Transactions on CAD of Integrated Circuits and Systems*, 2022.

[16] C. Pilato, A. B. Chowdhury, D. Sciuto, S. Garg, and R. Karri, "ASSURE: RTL locking against an untrusted foundry," *IEEE Transactions on VLSI Systems*, vol. 29, no. 7, 2021.

[17] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, S. Kim, V. N. Kravets, Y.-L. Li, R. Varadarajan, and M. Woo, "DATC RDF-2021: Design flow and beyond," in *IEEE/ACM ICCAD*, 2021, pp. 1–6.

[18] J. Bhandari, A. K. Thalakkattu Moosa, B. Tan, C. Pilato, G. Gore, X. Tang, S. Temple, P.-E. Gaillardon, and R. Karri, "Exploring eFPGA-based redaction for IP protection," in *IEEE/ACM ICCAD*, 2021.

[19] C. Muscari Tomajoli, L. Collini, J. Bhandari, A. K. Thalakkattu Moosa, B. Tan, X. Tang, , P.-E. Gaillardon, R. Karri, and C. Pilato, "ALICE: An automatic design flow for eFPGA redaction," in *ACM/EDAC/IEEE DAC*, 2022.

[20] J. Chen, M. Zaman, Y. Makris, R. D. S. Blanton, S. Mitra, and B. C. Schafer, "DECOY: DEflection-Driven HLS-Based Computation Partitioning for Obfuscating Intellectual PropertY," in *ACM/EDAC/IEEE DAC*, 2020, pp. 1–6.

[21] D. Sisejkovic, L. Collini, B. Tan, C. Pilato, R. Karri, and R. Leupers, "Designing ML-resilient locking at register-transfer level," in *ACM/EDAC/IEEE DAC*, 2022.