Research Project Proposal: IP Protection through Logic Locking at Register-Transfer Level

Luca Collini, luca.collini@mail.polimi.it

1. Introduction to the problem

The integrated circuit (IC) supply chain is the process that realizes a physical chip starting from its design. The entire process is becoming more and more distributed across different parties [8]. In fact most digital design houses are becoming fabless due to increasing manufacturing costs. The introduction of third-party entities in the IC supply chain brings new security challenges [1]. The major concerns are related to reverse engineering for intellectual property (IP) theft or for introduction of malicious modifications in the design [8]. The estimated loss due to IP violations alone was of $4 billions in 2008 [7] while the total loss from IC counterfeiting was estimated to be about $169 billions in 2011 [4]. This clearly shows that hardware protection techniques such as logic locking are crucial for the IC industry [8].

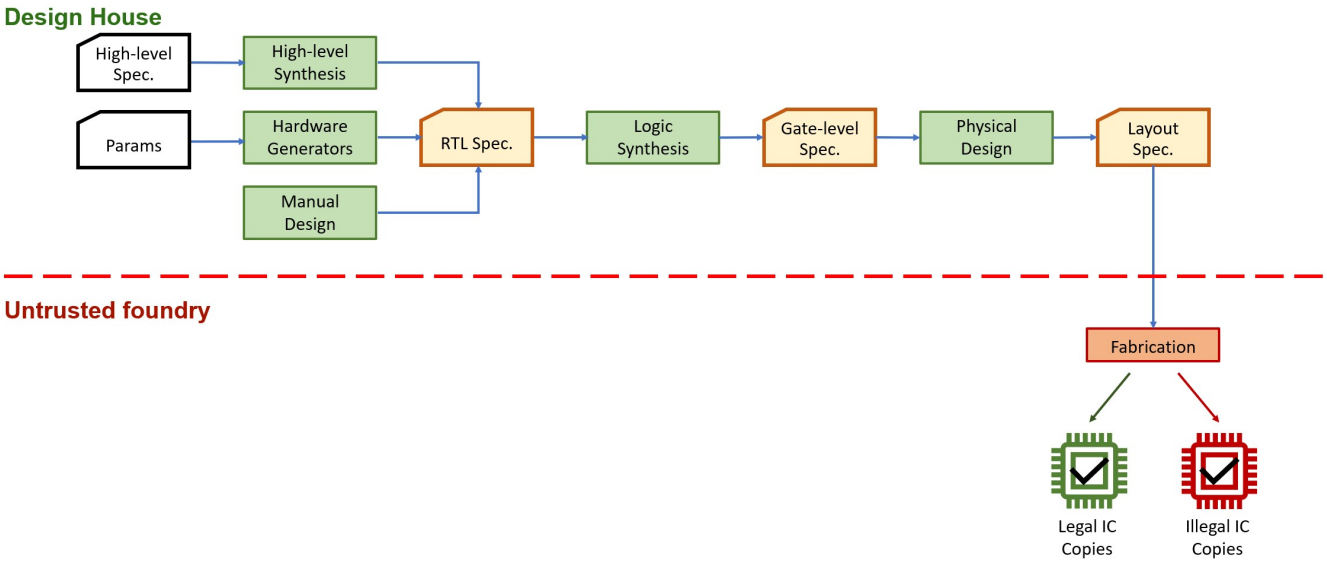


Figure 1: IC design flow

Logic Locking is a family of techniques for protecting the intellectual property (IP) of a design by adding logic controlled by an extra input, called *key*. In this way, the design is unusable until the correct key is provided to unlock it. While logic locking is conceptually simple, there are still open questions for its efficient application to chip designs:

- Which parts of the design should be locked?
- How to compare two locked designs from the security viewpoint?

Locking more parts of a design introduces more hardware overhead, but does not necessarily improve its security. Even though many metrics have been proposed to evaluate obfuscated designs, the only way to say which of two designs is better from the security viewpoint is to try to extract knowledge from the designs to eventually break them.

Logic locking brings area and timing overhead to the original design and in most cases it is unfeasible to lock the whole design, the goal of our research is to develop a complete logic locking framework that would allow us to understand where it is best to apply logic locking to obtain the best obfuscated design given a maximum area or timing overhead.

2. Main related works

Many logic locking techniques have been proposed. They can be classified as pre or post synthesis techniques depending on which design phase they need to be applied in. Each category has its pros and cons. Post-synthesis techniques are generally easier to apply because they do not need to modify the design tools. However some attacks exploit the fact that a post-synthesis insertion of key gates yields invalid design alternatives that do not adhere to well-established design patterns making it easy to deduce incorrect key bit values [9]. On the other hand pre-synthesis techniques, like TAO [6], apply obfuscation at the algorithmic level, but at the cost of more complex HLS tools and cannot be applied to components that are already described in hardware. Applying logic locking at the register-transfer level (RTL) may be a good compromise between HLS pre-synthesis techniques and post-synthesis techniques. A first approach at register-transfer level was proposed in [2] while a new promising preliminary approach has been presented in [5]. In this case, the description to lock is already at the hardware level (so it can be applied to pre-existing components) but semantic information is still in the design and not optimized by logic synthesis tools.

Different metrics have been proposed but all of them are empiric and experimental, making hard to apply them in automatic optimization methods. Finding a security property to compare two designs and tell which one is more secure is an open problem in hardware obfuscation. In contrast with most other areas of computer security, hardware obfuscation is missing security properties clearly defined by mathematical terms [8]. This would allow the use of optimization techniques to find the best obfuscated design with a fixed maximum overhead of area. ASSURE [5] and CDFG [2] show that logic locking at register-transfer level deserves further investigations. An approach at RT level would be close to code obfuscation and it may allow to inherit useful knowledge from that field. In fact, RTL descriptions are specified with hardware description languages such as Verilog and VHDL. Applying obfuscation on a Verilog description would not be much different than obfuscating a Java code. In this sense we aim to start from the concept of *Opaque Predicates* for software obfuscation and bring it to logic locking. Opaque predicates are the main building blocks for software obfuscation, based on predicates whose outcome depend on a value known by the programmer but difficult to calculate for the attacker [3]. In case of logic locking this value would be the locking key.

3. Research plan

The goal of this research is to develop a complete logic locking framework at register-transfer level that will allow us to say which parts of the design are better to obfuscate in order to obtain the most secure design given a constraint on area or timing overheads.

The nature of the research is hybrid since it is *theoretical* in the study of the techniques and metrics while it is of *application* and *implementation* nature in the testing phase and *experimental* in the evaluation phase. We plan to carry out this research with a two-phase development plan. The goal of the first development phase is to build a prototype implementing the RTL obfuscation techniques and develop the evaluation chain to measure cost (area and timing overhead) and security. After the first development phase, we will make a first evaluation of the techniques and metrics. The second phase aims at refining the techniques and the metrics identified in the first phase thanks to the evaluation feedback.

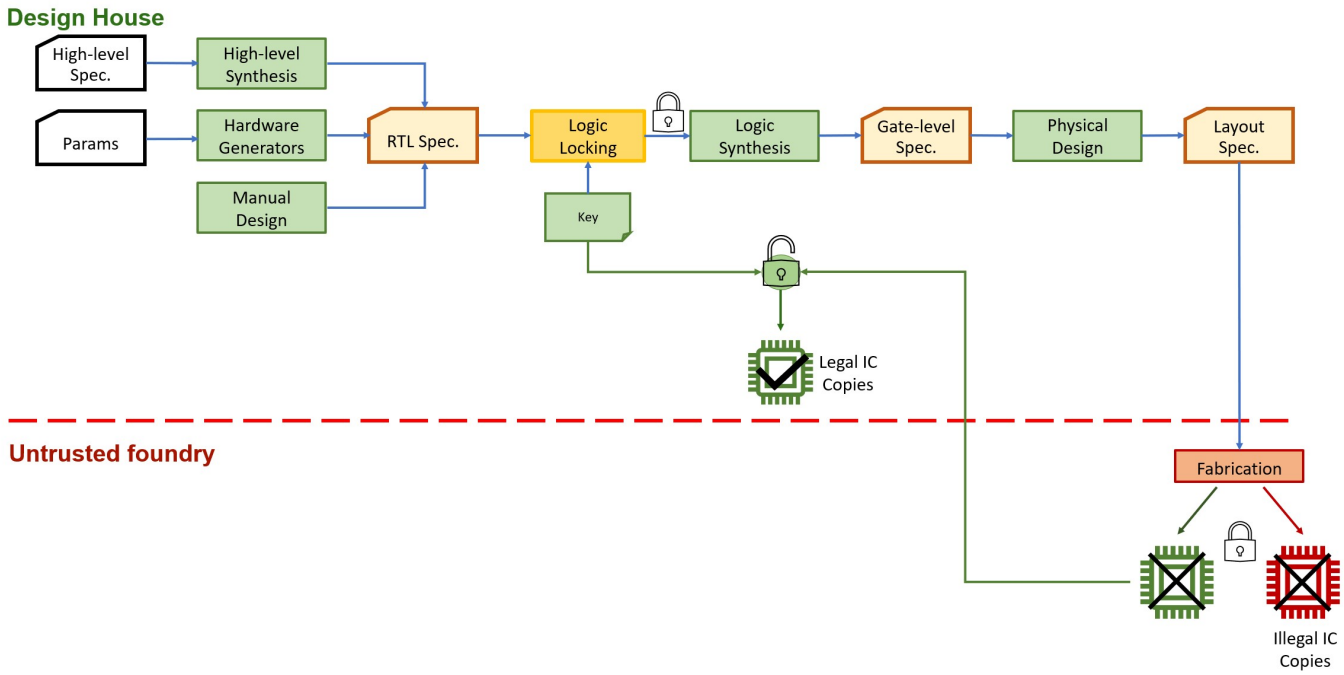
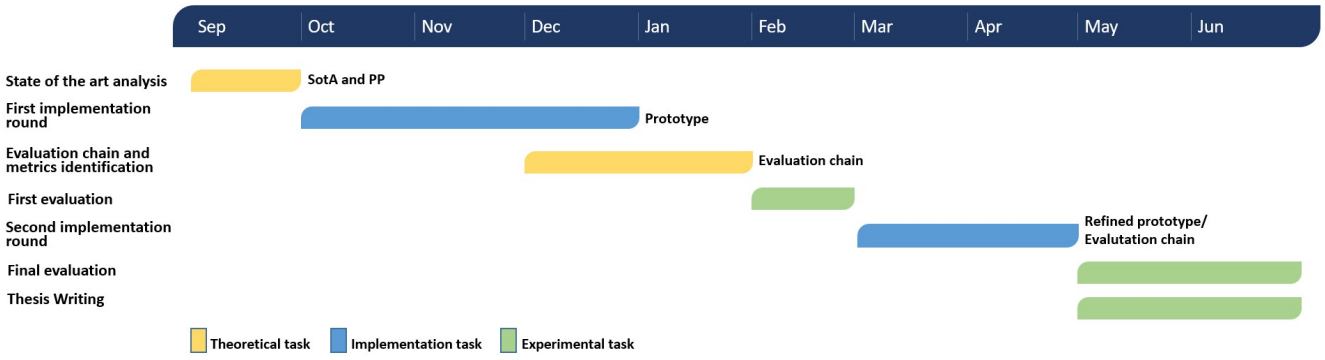


Figure 2: IC design flow with RTL locking

The following Gantt Diagram shows the different tasks on a time scale. The green, experimental evaluation tasks mark the end of development phases.



The outputs of the research will be evaluated according to some existing metrics, such as area and timing overheads, entropy and differential entropy [1], together with novel metrics that will be developed during the research. Area and timing overheads tell us the cost of the obfuscation in terms of power consumption and performance. Entropy is a metric related to the number of distinct outputs of the circuit. A circuit with entropy equal to one most resembles a random function. Differential entropy is measured with a miter circuit and represents the proportion of output bits values that differ between the obfuscated and the original design.

References

[1] Amir, S., Shakya, B., Xu, X., Jin, Y., Bhunia, S., Tehranipoor, M. M., and Forte, D. Development and evaluation of hardware obfuscation benchmarks. *Journal of Hardware and Systems Security 2* (2018), 142–161.

[2] Chakraborty, R., and Bhunia, S. Rtl hardware ip protection using key-based control and data flow obfuscation. pp. 405–410.

[3] Collberg, C., Thomborson, C., and Low, D. A taxonomy of obfuscating transformations. *http://www.cs.auckland.ac.nz/staff-cgi-bin/mjd/csTRcgi.pl?serial* (01 1997).

[4] Omdia. Top 5 most counterfeited parts represent a $169 billion potential challenge for global semiconductor market. Available at: https://www.electronicproducts.com/top-5-most-counterfeited-parts-represent-a-169-billion-potential-challenge-for-global-semiconductor-market/ (Last accessed: November 1, 2020), 2012.

[5] Pilato, C., Chowdhury, A. B., Sciuto, D., Garg, S., and Karri, R. Assure: Rtl locking against an untrusted foundry, 2020.

[6] Pilato, C., Regazzoni, F., Karri, R., and Garg, S. TAO: Techniques for algorithm-level obfuscation during high-level synthesis. In *Proceedings of the 55th Annual Design Automation Conference* (2018), DAC '18.

[7] Semi. Innovation is at risk as semiconductor equipment and materials industry loses up to $4 billion annually due to ip infringement. Available at: http://dev7.semi.org/en/white-paper-ip-infringement-causes-4-billion-loss-industry-annually (Last accessed: November 1, 2020), 2008.

[8] Shamsi, K., Li, M., Plaks, K., Fazzari, S., Pan, D. Z., and Jin, Y. Ip protection and supply chain security through logic obfuscation: A systematic overview. *ACM Trans. Des. Autom. Electron. Syst. 24*, 6 (Sept. 2019).

[9] Tan, B., Karri, R., Limaye, N., Sengupta, A., Sinanoglu, O., Rahman, M. M., Bhunia, S., Duvalsaint, D., D., R., Blanton, Rezaei, A., Shen, Y., Zhou, H., Li, L., Orailoglu, A., Han, Z., Benedetti, A., Brignone, L., Yasin, M., Rajendran, J., Zuzak, M., Srivastava, A., Guin, U., Karfa, C., Basu, K., Menon, V. V., French, M., Song, P., Stellari, F., Nam, G.-J., Gadfort, P., Althoff, A., Tostenrude, J., Fazzari, S., Breckenfeld, E., and Plaks, K. Benchmarking at the frontier of hardware security: Lessons from logic locking, 2020.