DATE: 2023/12/1

Department of Electrical and Electronic Information Engineering		ID	M203256
Name	Nobuhiko Bando		

Supervisor	Shuichi Ichikawa

Abstract

Title	Evaluation of logic locking technique implemented in high-level synthesis

In recent years, as the manufacturing cost of large-scale integrated circuits has risen, many companies become fabless and outsourced the manufacturing of integrated circuits. Outsourcing of manufacturing has raised concerns about the theft of intellectual property (IP) through reverse engineering. To protect IP from such security threats, various methods have been proposed to obfuscate the netlist of logic circuits. Logic locking is an IP protection techniques, in which a circuit operates correctly only when the correct key is applied to the circuit. Many of logic locking techniques used hardware description languages (HDLs) and were often embedded at the netlist level. In recent years, a new technology called high-level synthesis has been provided to automatically generate the HDL descriptions from the corresponding operating-level descriptions in high-level languages such as C. High-level synthesis makes it possible to embed logic locking easily and quickly, making it usable in more practical situations.

Takeda (2020) simulated and implemented a kind of logic locking called SFLL-HD into CHStone, a benchmark suite for high-level synthesis, in reference to the method proposed by Yasin et al. (2019), which combines high-level synthesis and SFLL-HD. Output corruption was evaluated by comparing the output of the wrong key values with the output of the correct key values, i.e., by the percentage of the output corruption. Logic locking was embedded in three of the CHStone applications, and in some cases output corruption rate was approximately 4.2 times greater than the expected values.

The deviation of the Takeda's experimental results from the expected values was considered to arise from the inappropriate target of logic-locking. Therefore, I performed simulations of the different part of the logic-locking target from Takeda's method and compared them with Takeda's results. Takeda locked "some functions," while this study locked the "entire circuit." In addition, the number of simulated CHStone applications was increased from three to eight. The amount of resources required to implement each application was investigated and compared with Takeda's results.

Simulation results show that the proposed method yields closer results to the expected values than Takeda's method. For all Hamming distances except where the measured value is zero, the error rate between the average of the simulation results and the expected values in this study was smaller than that between the average of Takeda's simulation results and the expected values. Implementation results show that resource overhead varies depending on the size of the application. Small applications show an overhead of more than 10%, while medium and larger applications show an overhead of 3% or less. The above results show that locking the entire circuit is more effective than locking a part of the function for practical implementation of the locking technique.