BOOLEAN NETWORK OPTIMIZATION BY STOCHASTIC REWIRING

K.O. CHEREVKO

In today's rapidly evolving technological world, the demand for smaller yet more powerful computers is escalating at an unprecedented pace, driving a significant need for research in circuit optimizations. This surge in demand underscores the critical role of logic synthesis, which lies at the heart of digital circuit design. Logic synthesis serves as a pivotal stage in converting a high-level functionality description into an efficient hardware implementation. It is key to meeting the ever-increasing demands for miniaturization and performance enhancement in modern electronics.

At its core, logic synthesis aims to optimize various aspects of circuit performance, including area, power consumption, speed, and delay, while ensuring the correct functionality of the circuit.

Among these optimization goals, area minimization stands out as one of the most straightforward and natural criteria. The size of a circuit directly impacts factors, such as manufacturing cost, chip area, and overall system complexity. By developing various solutions for area optimization, we address the industry's demand for producing smaller circuits, especially considering the increase in manufacturing costs of silicon chips [1].

The above-described factors prove the need for new area minimization algorithms. The contribution of this work is a novel algorithm for area minimization which is based on the notion of redundancy addition and removal. A similar approach has been used in the past, for example, [2] and [3]. However, the proposed method differs in several ways, in particular, it uses an AIG data structure for representing the circuit and relies on a simple randomization strategy to detect possible additions and removals while maintaining the correctness of the circuit. Another difference is that the proposed method targets structural change rather than immediate improvement. This is particularly important for hard-to-optimize circuits, which can be reduced only after a substantial restructuring. On top of that algorithm implementation involves various heuristics such as primary inputs prioritization and the following criterion for acceptable fanins.

Theorem 1. *Let f_n denote the function of the node n, f_i denote the function of the fanin, and f^{care} denote the care set of the node n. Let $f_1 = f_n$ represent the function before adding the fanin, and let $f_2 = f_n \wedge f_i$ represent the function*

after adding the fanin. Define $D = f_1 \oplus f_2$ as the Boolean difference after the change. Then, for the fanin to be inserted, the following property should hold:

$$D \wedge f^{care} = 0 \tag{1}$$

References

[1] Dr. Ian Cutress *TSMC Financial Year 2022*, https://morethanmoore.substack.com/p/tsmc-financial-year-2022.

[2] Lam, T.K. and Tang, W.C. and Wei, X. and Diao, Y. and Wu, D.Y.L. *Boolean Circuit Rewiring: Bridging Logical and Physical Designs* — 2016. — 304 p.

[3] Chih-Wei Chang and Marek-Sadowska, M. *Single-pass redundancy-addition-and-removal.* — 2001. — P. 606–609.

National University of Kyiv-Mohyla, Kyiv, Ukraine
Email address: christina.cherevko@gmail.com