RevKit: A Toolkit for Reversible Circuit Design

Mathias Soeken  Stefan Frehse  Robert Wille  Rolf Drechsler
Institute of Computer Science, University of Bremen, Bremen, Germany
revkit@informatik.uni-bremen.de – http://www.revkit.org

It is a widely supported prediction that conventional computer hardware technologies are going to reach their limits in the near future. Thus, to further satisfy the needs for more computational power, alternatives are required that go beyond the scope of conventional technologies. Reversible logic marks a promising new direction where all operations are performed in an invertible manner [1]. That is, in contrast to conventional logic, only bijective operations are allowed implying a reversible computation (i.e. the inputs can be obtained from the outputs and vice versa). This reversibility builds the basis for emerging technologies that may replace, or at least enhance, conventional computer chips, e.g. in the domain of quantum computation, low-power design, optical computing, DNA computing, or nanotechnologies.

Motivated by these applications, nowadays reversible logic is seen as a promising research area. As a consequence, in the last years, computer scientists started to develop new methods for the design of reversible circuits. Among others, these include approaches for synthesis, optimization, simulation, verification, and test. However, most of the resulting methods are not publicly available. This often makes the development of new methods more difficult since e.g. previous approaches are not available for comparison. Furthermore, approaches have to be re-implemented from scratch in order to modify or improve them. The lack of tools for reversible hardware design impedes the entrance for beginners into the topic.

In this work, we introduce RevKit, an open source toolkit for reversible circuit design. The motivation behind RevKit is to make recent developments in the domain of reversible circuit design accessible to other researchers. Besides basic functionality (such as parsers and export functions), RevKit already provides elaborated methods for synthesis, optimization, and verification. It is built on top of a modular and extendable framework which enables the addition of new methods and tools. Furthermore, an intuitive graphical user interface easily allows to create and execute customized design flows and, thus, exploits the full potential of the various design tools.

Fig. 1 shows a possible application scenario of RevKit. Using drag-and-drop, a customized design flow composed of some of the approaches available in RevKit has been defined aiming at the realization of a given function (function.pla) as a circuit. Therefore, two different approaches are applied, called BDD synthesis and Transformation Based. Both methods are enriched by appropriate post-synthesis or pre-synthesis methods (for more details on the respective methods, we refer to [2]). Finally, the two resulting circuits are checked for equivalence to verify the correctness of the results.

Using RevKit various customized flows can easily be specified and executed. The original specification as well as the resulting circuit can be visualized (as an example see Fig. 2). Different abstractions and hierarchies are thereby supported. Therewith, complex tasks for reversible circuit design can be handled using the toolkit. RevKit is available online at http://www.revkit.org.

REFERENCES


Fig. 1. Application scenario

Fig. 2. Visualization of a circuit