
Summary: The generation of reversible circuits from high-level code is an important problem in several application domains, including low-power electronics and quantum computing. Existing tools compile and optimize reversible circuits for various metrics, such as the overall circuit size or the total amount of space required to implement a given function reversibly. However, little effort has been spent on verifying the correctness of the results, an issue of particular importance in quantum computing. There, compilation allows not only mapping to hardware, but also the estimation of resources required to implement a given quantum algorithm, a process that is crucial for identifying which algorithms will outperform their classical counterparts. We present a reversible circuit compiler called REVerC, which has been formally verified in $F^*$ and compiles circuits that operate correctly with respect to the input program. Our compiler compiles the ReVs language [21] to combinational reversible circuits with as few ancillary bits as possible, and provably cleans temporary values.

For the entire collection see [Zbl 1369.68033].

MSC:

68N20  Theory of compilers and interpreters
68Q06  Networks and circuits as models of computation; circuit complexity
68Q09  Other nonclassical models of computation
68Q12  Quantum algorithms and complexity in the theory of computing
68Q60  Specification and verification (program logics, model checking, etc.)

Full Text: DOI arXiv