Chair for Design Automation
TUM School of Computation, Information and Technology
Technical University of Munich

Design Automation Tools and Software for Quantum Computing

Lukas Burgholzer (Supervisor: Robert Wille)

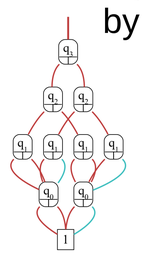
connect

Abstract

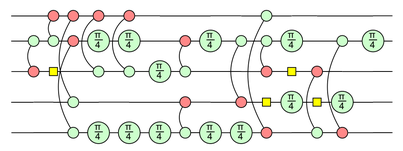
Quantum computing promises to solve problems beyond the reach of today's machines, but it requires efficient and reliable software tools to realize its potential. This poster gives an overview of various contributions towards design automation methods and software for quantum computing that leverage existing knowledge and expertise in classical circuit and system design. It focuses on three major tasks: simulation, compilation, and verification of quantum circuits. The proposed solutions demonstrate significant improvements in efficiency, scalability, and reliability for all tasks and constitute the backbone of the *Munich Quantum Toolkit (MQT),* a collection of open-source tools for quantum computing. The respective solutions advance the state of the art in quantum computing and illustrate the benefits of design automation methods for this emerging field.

Data-Structures & Core Methods – MQT Core

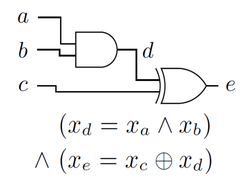
In our work, we developed efficient data-structures and core methods that are the **backbone of any reliable design approach or tool**. They allow **tackling the computational challenges** posed by various design tasks in quantum computing.



Decision Diagrams



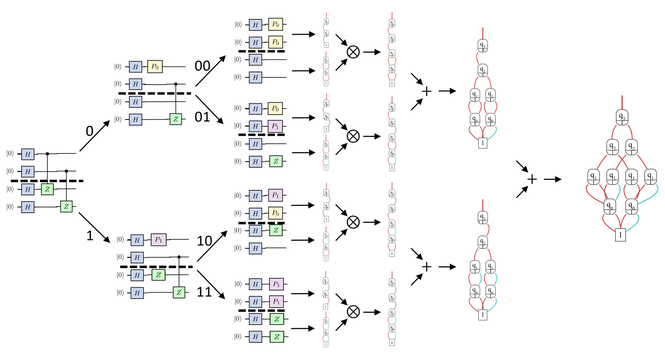
ZX Calculus



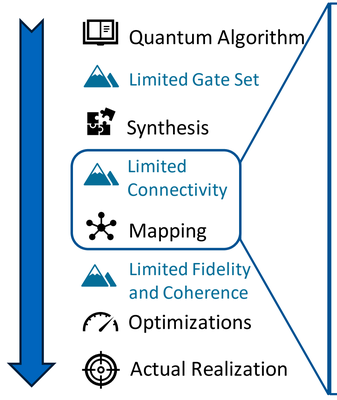
SAT Encodings

Simulation – MQT DDSIM

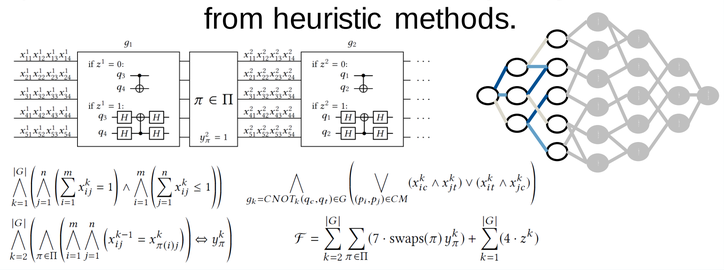
Simulations of quantum circuits on classical computers allows **exploring and testing** applications, **studying** the **complete** quantum **state**, and **measuring the advantages** of quantum computers over classical ones.



Compilation – MQT QMAP

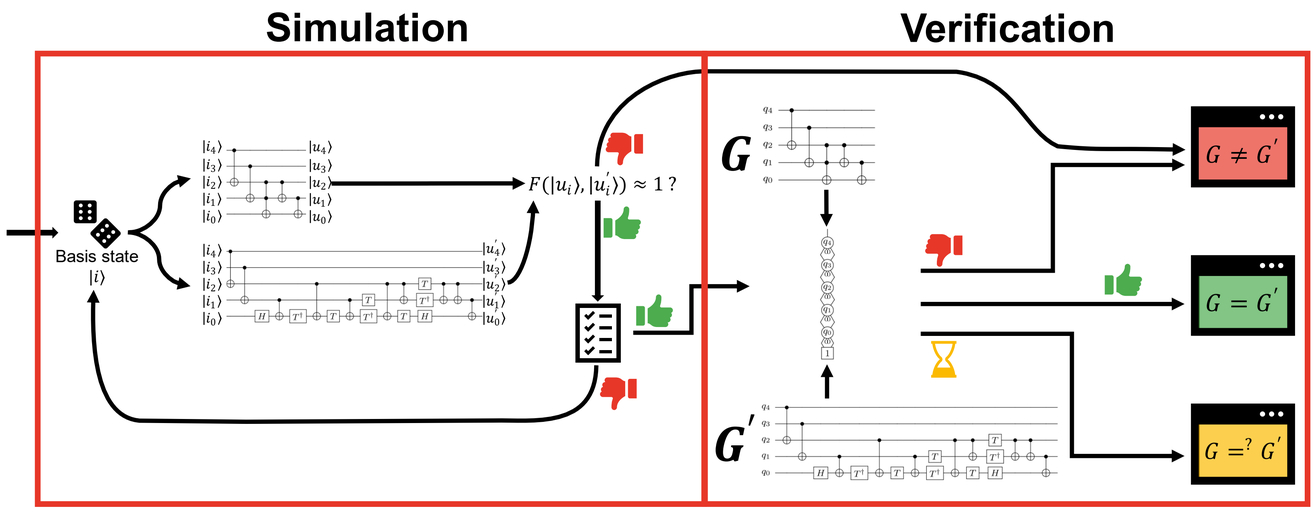


Developing **optimal solutions** is crucial for establishing **lower bounds** on the achievable performance and to **evaluate the quality** of results from heuristic methods.



Verification – MQT QCEC

Verification is an essential step to **increase trust in the whole compilation stack** for quantum computing and **ensure a reliable and error-free design flow.**



Complete verification flow that supports:

- ❑ Verifying results from compilation flows
- ❑ Specialized stimuli generation schemes
- ❑ Verification of dynamic quantum circuits
- ❑ Verification of parametrized quantum circuits

Open-Source Implementations

- ❑ **Everything is open source**
- ❑ **C++** for performance
- ❑ **Python** for accessibility
- ❑ **Push-button solutions**
- ❑ Native **Qiskit** integration
- ❑ Actively **maintained** and **documented**

MQT Core | MQT DDSIM | MQT QMAP | MQT QCEC

Publications & Accomplishments

- ❑ Publications **since 2019:**
  - ❑ **8 journal articles**
    (2x TCAD, 2x TQC, 2x Softw. Imp., 1x JETCAS, 1x Array)
  - ❑ **31 papers** in international conferences, including
    - ❑ 7 ASP-DAC (1x Candidate BPA)
    - ❑ 6 DAC
    - ❑ 2 DATE
    - ❑ 1 ICCAD
- ❑ Mentor at the 10th and 11th NYUAD Hackathon (2022+2023)
- ❑ JKU Young Researchers' Award from JKU Linz (2022)
- ❑ IBM Qiskit Advocate (2021)
- ❑ Invitation to 2 IBM Qiskit Camps (2019, 2020)
- ❑ Top placements at various quantum computing challenges

Lukas Burgholzer| lukas.burgholzer@tum.de | Chair for Design Automation @ Technical University of Munich