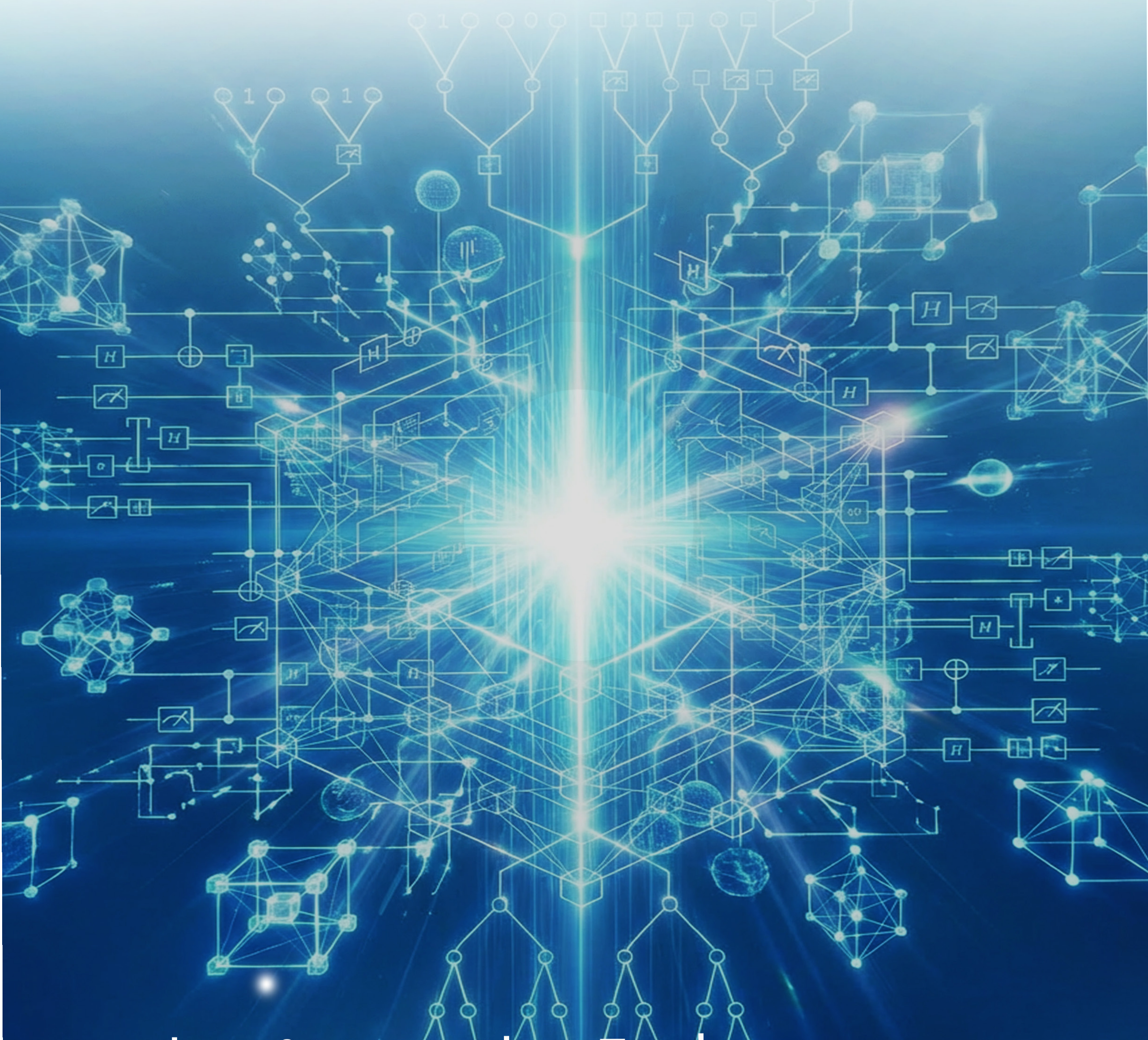
MUNICH QUANTUM TOOLKIT



Design Automation Tools and Software for Quantum Computing

MUNICH QUANTUM TOOLKIT

The Munich Quantum Toolkit (MQT) is developed by the Chair for Design Automation at the Technical University of Munich and supported by the Munich Quantum Software Company (MQSC). Among others, it is part of the Munich Quantum Software Stack (MQSS) ecosystem, which is being developed as part of the Munich Quantum Valley (MQV) initiative.

Check it out!

More at:
https://mqt.readthedocs.io

GitHub Munich Quantum Toolkit
https://github.com/munich-quantum-toolkit

A new era of computing

Quantum computers are becoming a reality and numerous quantum computing applications with a near-term perspective (e.g., for finance, chemistry, machine learning, and optimisation) and with a long-term perspective (e.g., for cryptography or unstructured search) are currently being investigated. Unlike conventional machines that process only 0s and 1s, quantum computers use qubits that can exist in superpositions and be entangled with each other. These features enable powerful new capabilities but also pose fundamental challenges for programming, design, and verification—challenges that make even basic tasks far more complex than in classical computing.

The need for quantum computing software

In fact, designing and realising potential applications for these devices in a scalable fashion requires automated, efficient, and user-friendly software tools that cater to the needs of end users, engineers, and physicists at every level of the entire quantum software stack. Many of the problems to be tackled in that regard are similar to design problems from the classical realm for which sophisticated design automation tools have been developed in the previous decades. But these established programming languages, compilers, and verification tools are not directly applicable to quantum computers. Without new methods, there is a risk of having powerful hardware with no effective way to exploit it.

The Munich Quantum Toolkit (MQT)

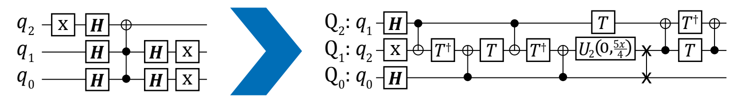
The Munich Quantum Toolkit (MQT) is a collection of software tools for quantum computing that explicitly addresses these needs. Our overarching objective is to provide solutions for design tasks across the entire quantum software stack. This entails high-level support for end users in realising their applications, efficient methods for the classical simulation, compilation, and verification of quantum circuits, tools for quantum error correction, support for physical design, and more. These methods are supported by corresponding data structures (such as decision diagrams or the ZX-calculus) and core methods (such as SAT encodings/solvers). All of the developed tools are available as open-source implementations and are hosted on github.com/munich-quantum-toolkit.

Application [1] [2]

- Quantum program benchmark suite
- Resource estimation
- Workflows for deriving quantum solutions to classical problems

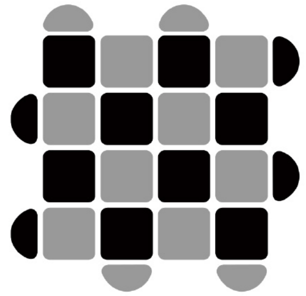
Compilation [5] [6] [7] [8] [9] [10]

- Compiler infrastructure (MLIR / QIR)
- Technology-specific compiler passes for
  - Superconducting platforms
  - Neutral Atom platforms
  - Trapped Ion platforms
- Compiler and scheduling optimisation
- Multi-level (qudit) compilation



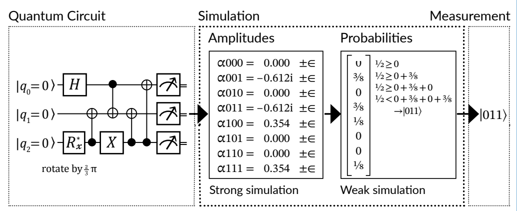
Error correction [13]

- Decoding algorithms for colour codes and QLDPC codes
- Fault-tolerant gadget optimisation
- Lattice surgery compilation and numerical simulations



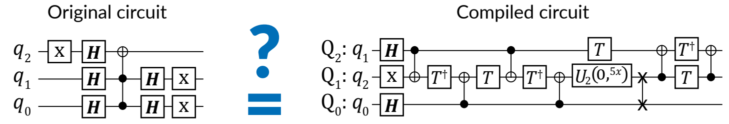
Simulation [3] [4]

- Efficient classical circuit simulation based on decision diagrams, supporting unitary, hybrid, and sampling approaches
- Simulation of open quantum systems and noisy circuits using tensor network methods



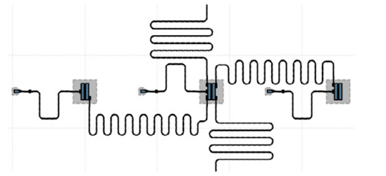
Verification [11] [12]

- Equivalence checking of quantum circuits
- Verification of compilation results
- Debugging of quantum programs



Hardware

- Application specific physical design for superconducting platform



Data structures and core methods [14] [15] [16]

- Efficient data structures
- Dedicated core methods (optimal and heuristic)
- Based on C++ and Python

Decision Diagrams

Tensor Networks

ZX-Calculus

SAT/SMT Solvers

Machine Learning

Heuristics

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

Leave a ★

Munich Quantum Toolkit

1 MQT ProblemSolver — Application

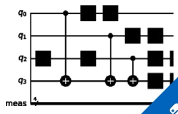
A tool for solving problems using quantum computing.

github.com/munich-quantum-toolkit/problemsolver

PyPI

2 MQT Bench — Application

A quantum circuit benchmark suite.

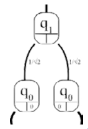


github.com/munich-quantum-toolkit/bench

PyPI

3 MQT DDSIM — Simulation

A tool for classical quantum circuit simulation-based on decision diagrams.



github.com/munich-quantum-toolkit/ddsim

PyPI

4 MQT YAQS — Simulation

A tool for simulating open quantum systems, noisy quantum circuits and realistic quantum hardware.

Yet Another YAQS Quantum Simulator

github.com/munich-quantum-toolkit/yaqs

PyPI

5 MQT Predictor — Compilation

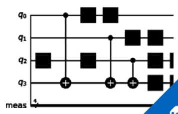
A tool for determining good quantum circuit compilation options.

github.com/munich-quantum-toolkit/predictor

PyPI

6 MQT SyReC — Compilation

A tool for the synthesis of reversible circuits/quantum computing oracles.



github.com/munich-quantum-toolkit/syrec

PyPI

7 MQT QMAP — Compilation

A tool for quantum circuit mapping and Clifford circuit optimisation/synthesis.

github.com/munich-quantum-toolkit/qmap

PyPI

8 MQT NAViz — Compilation

An application to visualise compilation output for neutral atom quantum computers.

github.com/munich-quantum-toolkit/naviz

PyPI

9 MQT IonShuttler — Compilation

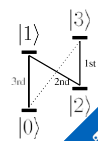
A tool for generating shuttling schedules for QCCD architectures.

github.com/munich-quantum-toolkit/ionshuttler

PyPI

10 MQT Qudits — Compilation

A tool for compiling to high-dimensional quantum systems.



github.com/munich-quantum-toolkit/qudits

PyPI

11 MQT Debugger — Verification

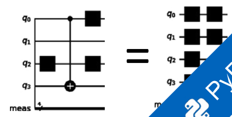
A tool for debugging quantum circuits which can be integrated into your IDE.

github.com/munich-quantum-toolkit/debugger

PyPI

12 MQT QCEC — Verification

A tool for quantum circuit equivalence checking.



github.com/munich-quantum-toolkit/qcec

PyPI

13 MQT QECC — QEC

A tool for quantum error correcting codes.

github.com/munich-quantum-toolkit/qecc

PyPI

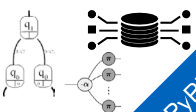
14 MQT DDVis — Data Structures

A web-application visualising decision diagrams for quantum computing.

github.com/munich-quantum-toolkit/ddvis

15 MQT Core — Data Structures

The backbone of the MQT intermediate representation (IR) decision diagram and ZX package.



github.com/munich-quantum-toolkit/core

PyPI

16 MQT QuSAT — Core Methods

A tool for encoding quantum computing using satisfiability testing (SAT) techniques.

$$F \wedge (x_1 \wedge \neg x_2)$$
$$F \wedge (x_3 \wedge x_2)$$

github.com/munich-quantum-toolkit/qusat

PyPI

Technische Universität München

European Research Council — Established by the European Commission

Munich Quantum Valley

Grant agreement ID: 101001318