The Journal of Open Source Software

MQT Core: The Backbone of the Munich Quantum Toolkit (MQT)

Lukas Burgholzer [1,2¶], **Yannick Stade** [1], **Tom Peham** [1], **and Robert Wille** [1,2,3]

1 Chair for Design Automation, Technical University of Munich, Germany **2** Munich Quantum Software Company GmbH, Garching near Munich, Germany **3** Software Competence Center Hagenberg GmbH, Hagenberg, Austria ¶ Corresponding author

DOI: 10.21105/joss.07478

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @1ucian0
- @edyounis
- @josh146

Submitted: 08 November 2024
Published: 07 April 2025

Summary

MQT Core is an open-source C++ and Python library for quantum computing that forms the backbone of the quantum software tools developed as part of the *Munich Quantum Toolkit (MQT, (Wille et al., 2024))* by the Chair for Design Automation at the Technical University of Munich as well as the Munich Quantum Software Company (MQSC). To this end, it consists of multiple components that are used throughout the MQT, including a fully fledged intermediate representation (IR) for quantum computations, a state-of-the-art decision diagram (DD) package for quantum computing, and a state-of-the-art ZX-diagram package for working with the ZX-calculus. Pre-built binaries are available via PyPI for all major operating systems and all modern Python versions. MQT Core is fully compatible with IBM's Qiskit 1.0 and above (Javadi-Abhari et al., 2024), as well as the OpenQASM format (Cross et al., 2022), enabling seamless integration with the broader quantum computing community.

Statement of Need

Quantum computing is rapidly transitioning from theoretical research to practice, with potential applications in fields such as finance, chemistry, machine learning, optimization, cryptography, and unstructured search. However, the development of scalable quantum applications requires automated, efficient, and accessible software tools that cater to the diverse needs of end users, engineers, and physicists across the entire quantum software stack.

The Munich Quantum Toolkit (MQT, (Wille et al., 2024)) addresses this need by leveraging decades of design automation expertise from the classical computing domain. Developed by the Chair for Design Automation at the Technical University of Munich, the MQT provides a comprehensive suite of tools designed to support various design tasks in quantum computing. These tasks include high-level application development, classical simulation, compilation, verification of quantum circuits, quantum error correction, and physical design.

MQT Core offers a flexible intermediate representation for quantum computations that forms the basis for working with quantum circuits throughout the MQT. The library provides interfaces to IBM's Qiskit (Javadi-Abhari et al., 2024) and the OpenQASM format (Cross et al., 2022) to make the developed tools accessible to the broader quantum computing community. Furthermore, MQT Core integrates state-of-the-art data structures for quantum computing, such as decision diagrams (Wille et al., 2023) and the ZX-calculus (Duncan et al., 2020; van de Wetering, 2020), that power the MQT's software packages for classical quantum circuit simulation (MQT DDSIM), compilation (MQT QMAP), verification (MQT QCEC), and more. As such, MQT Core has enabled more than 30 research papers over its first five years of development (Burgholzer et al., 2020; Burgholzer, Bauer, et al., 2021; Burgholzer, Raymond, et al., 2021; Burgholzer, Kueng, et al., 2021; Burgholzer, Ploier, et al., 2022a, 2022b; Burgholzer, Schneider, et al., 2022; Burgholzer & Wille, 2020a, 2020b, 2021; Grurl, Pichler, et al., 2023; Grurl et al., 2020, 2021; Grurl, Fuß, et al., 2023; Hillmich et al., 2021, 2020, 2022; Hillmich, Markov, et al., 2020; Hillmich, Kueng, et al., 2020; Peham et al., 2022b, 2022a, 2023b, 2023a; Peham, Brandl, et al., 2023; Sander et al., 2023; Schmid, Locher, et al., 2024; Schmid, Park, et al., 2024; Wille et al., 2020, 2021, 2022, 2023; Wille & Burgholzer, 2022).

To ensure performance, MQT Core is primarily implemented in C++. Since the quantum computing community predominantly uses Python, MQT Core provides Python bindings that allow seamless integration with existing Python-based quantum computing tools. In addition, pre-built Python wheels are available for all major platforms and Python versions, making it easy to install and use MQT Core in various environments without the need for manual compilation.

Related Work

MQT Core builds on a rich history of research in quantum computing, design automation, and data structures. The design of its IR is heavily inspired by IBM's Qiskit (Javadi-Abhari et al., 2024), with which it has stayed compatible since `qiskit-terra` version 0.16.1 (released at the end of 2020). MQT Core remains one of the few libraries providing drop-in replacements for large parts of Qiskit's core data structures in C++. Alternative IRs, that come as part of larger quantum computing frameworks, include Quantinuum's C++-based t|ket⟩ (Sivarajah et al., 2021), LBNL's Python-based `bqskit` (Younis et al., 2021), Xanadu's MLIR-based `catalyst` (Ittah et al., 2024), and NVIDIA's MLIR-based `CUDA-Q` (*CUDA-q*, 2024).

The origin of the decision diagram package in MQT Core dates back to the seminal work on Quantum Multiple-Valued Decision Diagrams (QMDDs) (Niemann et al., 2016). It provides a state-of-the-art implementation of QMDDs that natively integrates with the MQT Core IR. Alternative types of quantum decision diagrams and related software packages include TDD's (Hong et al., 2022), Bit-Slicing Decision Diagrams (Tsai et al., 2021), and LIMDDs (Vinkhuijzen et al., 2023). In comparison to MQT Core, most of these libraries have remained academic prototypes and have not seen widespread adoption in the quantum computing community.

The ZX-diagram package in MQT Core is inspired by the `PyZX` library (Kissinger & Wetering, 2020) and the t|ket⟩ (Sivarajah et al., 2021) compiler. It provides an efficient C++ implementation of core ZX-calculus concepts and is tightly integrated with the MQT Core IR. Compared to other implementations, the ZX package in MQT Core is fine-tuned for verification use cases and provides dedicated support for handling qubit permutations as well as numerical inaccuracies that arise in practice.

Acknowledgements

We would like to thank all past and present contributors to the MQT Core project, including (in alphabetical order): Hartwig Bauer, Martin Fink, Ioan-Albert Florea, Elias Foramitti, Rebecca Ghidini, Thomas Grurl, Stefan Hillmich, Fabian Hingerl, Daniel Lummerstorfer, Joachim Marin, Katrin Muck, Christoph Pichler, Tobias Prienberger, Parham Rahimi, Janez Rotman, Damian Rovara, Roope Salmi, Aaron Sander, Ludwig Schmid, Sarah Schneider, Tianyi Wang, Theresa Wasserer, and Alwin Zulehner. Their contributions have been instrumental in making MQT Core a robust and feature-rich library.

The Munich Quantum Toolkit has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement

No. 101001318), the Bavarian State Ministry for Science and Arts through the Distinguished Professorship Program, as well as the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

References

Burgholzer, L., Bauer, H., & Wille, R. (2021). Hybrid Schrödinger-Feynman simulation of quantum circuits with decision diagrams. *Int'l Conf. On Quantum Computing and Engineering*. https://doi.org/10.1109/QCE52317.2021.00037

Burgholzer, L., Kueng, R., & Wille, R. (2021). Random stimuli generation for the verification of quantum circuits. *Asia and South Pacific Design Automation Conf.* https://doi.org/10.1145/3394885.3431590

Burgholzer, L., Ploier, A., & Wille, R. (2022a). Exploiting arbitrary paths for the simulation of quantum circuits with decision diagrams. *Design, Automation and Test in Europe*. https://doi.org/10.23919/DATE54114.2022.9774631

Burgholzer, L., Ploier, A., & Wille, R. (2022b). Simulation paths for quantum circuit simulation with decision diagrams: What to learn from tensor networks, and what not. *IEEE Trans. On CAD of Integrated Circuits and Systems*. https://doi.org/10.1109/TCAD.2022.3197969

Burgholzer, L., Raymond, R., Sengupta, I., & Wille, R. (2021). Efficient construction of functional representations for quantum algorithms. *Int'l Conf. Of Reversible Computation*. https://doi.org/10.1007/978-3-030-79837-6_14

Burgholzer, L., Raymond, R., & Wille, R. (2020). Verifying results of the IBM Qiskit quantum circuit compilation flow. *Int'l Conf. On Quantum Computing and Engineering*. https://doi.org/10.1109/QCE49297.2020.00051

Burgholzer, L., Schneider, S., & Wille, R. (2022). Limiting the search space in optimal quantum circuit mapping. *Asia and South Pacific Design Automation Conf.* https://doi.org/10.1109/ASP-DAC52403.2022.9712555

Burgholzer, L., & Wille, R. (2020a). Improved DD-based equivalence checking of quantum circuits. *Asia and South Pacific Design Automation Conf.* https://doi.org/10.1109/ASP-DAC47756.2020.9045153

Burgholzer, L., & Wille, R. (2020b). The power of simulation for equivalence checking in quantum computing. *Design Automation Conf.* https://doi.org/10.1109/DAC18072.2020.9218563

Burgholzer, L., & Wille, R. (2021). Advanced equivalence checking for quantum circuits. *IEEE Trans. On CAD of Integrated Circuits and Systems*. https://doi.org/10.1109/TCAD.2020.3032630

Cross, A., Javadi-Abhari, A., Alexander, T., De Beaudrap, N., Bishop, L. S., Heidel, S., Ryan, C. A., Sivarajah, P., Smolin, J., Gambetta, J. M., & others. (2022). OpenQASM 3: A broader and deeper quantum assembly language. *ACM Transactions on Quantum Computing*, *3*(3), 1–50. https://doi.org/10.1145/3505636

CUDA-q. (2024). https://doi.org/10.5281/zenodo.8092233

Duncan, R., Kissinger, A., Perdrix, S., & van de Wetering, J. (2020). Graph-theoretic simplification of quantum circuits with the ZX-calculus. *Quantum*, *4*, 279. https://doi.org/10.22331/q-2020-06-04-279

Grurl, T., Fuß, J., & Wille, R. (2020). Considering decoherence errors in the simulation of quantum circuits using decision diagrams. *Iccad*. https://doi.org/10.1145/3400302.3415622

Grurl, T., Fuß, J., & Wille, R. (2023). Noise-aware quantum circuit simulation with decision diagrams. *IEEE Trans. On CAD of Integrated Circuits and Systems*, *42*(3), 860–873. https://doi.org/10.1109/TCAD.2022.3182628

Grurl, T., Kueng, R., Fuß, J., & Wille, R. (2021). Stochastic quantum circuit simulation using decision diagrams. *Design, Automation and Test in Europe (DATE)*. https://doi.org/10.23919/DATE51398.2021.9474135

Grurl, T., Pichler, C., Fuß, J., & Wille, R. (2023). Automatic implementation and evaluation of error-correcting codes for quantum computing: An open-source framework for quantum error correction. *VLSI Design*, 301–306. https://doi.org/10.1109/VLSID57277.2023.00068

Hillmich, S., Kueng, R., Markov, I. L., & Wille, R. (2020). As accurate as needed, as efficient as possible: Approximations in DD-based quantum circuit simulation. *Design, Automation and Test in Europe*. https://doi.org/10.23919/DATE51398.2021.9474034

Hillmich, S., Markov, I. L., & Wille, R. (2020). Just like the real thing: Fast weak simulation of quantum computation. *Design Automation Conf.* https://doi.org/10.1109/DAC18072.2020.9218555

Hillmich, S., Zulehner, A., Kueng, R., Markov, I. L., & Wille, R. (2022). Approximating decision diagrams for quantum circuit simulation. *ACM Transactions on Quantum Computing*, *3*(4), 1–21. https://doi.org/10.1145/3530776

Hillmich, S., Zulehner, A., & Wille, R. (2021). Exploiting quantum teleportation in quantum circuit mapping. *Asia and South Pacific Design Automation Conf.*, 792–797. https://doi.org/10.1145/3394885.3431604

Hillmich, S., Zulehner, A., & Wille, R. (2020). Concurrency in DD-based quantum circuit simulation. *Asia and South Pacific Design Automation Conf.* https://doi.org/10.1109/ASP-DAC47756.2020.9045711

Hong, X., Zhou, X., Li, S., Feng, Y., & Ying, M. (2022). A tensor network based decision diagram for representation of quantum circuits. *ACM Transactions on Design Automation of Electronic Systems*. https://doi.org/10.1145/3514355

Ittah, D., Asadi, A., Lopez, E. O., Mironov, S., Banning, S., Moyard, R., Peng, M. J., & Izaac, J. (2024). Catalyst: A python JIT compiler for auto-differentiable hybrid quantum programs. *Journal of Open Source Software*, *9*(99), 6720. https://doi.org/10.21105/joss.06720

Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C. J., Lishman, J., Gacon, J., Martiel, S., Nation, P. D., Bishop, L. S., Cross, A. W., Johnson, B. R., & Gambetta, J. M. (2024). *Quantum computing with Qiskit*. https://doi.org/10.48550/arXiv.2405.08810

Kissinger, A., & Wetering, J. van de. (2020). PyZX: Large scale automated diagrammatic reasoning. *Int'l Conf. Quantum Physics and Logic*. https://doi.org/10.4204/EPTCS.318.14

Niemann, P., Wille, R., Miller, D. M., Thornton, M. A., & Drechsler, R. (2016). QMDDs: Efficient quantum function representation and manipulation. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*. https://doi.org/10.1109/TCAD.2015.2459034

Peham, T., Brandl, N., Kueng, R., Wille, R., & Burgholzer, L. (2023). Depth-optimal synthesis of Clifford circuits with SAT solvers. *Int'l Conf. On Quantum Computing and Engineering*. https://doi.org/10.1109/QCE57702.2023.00095

Peham, T., Burgholzer, L., & Wille, R. (2022a). Equivalence checking of quantum circuits with the ZX-Calculus. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. https://doi.org/10.1109/JETCAS.2022.3202204

Peham, T., Burgholzer, L., & Wille, R. (2022b). Equivalence checking paradigms in quantum circuit design: A case study. *Design Automation Conf.* https://doi.org/10.1145/3489517.3530480

Peham, T., Burgholzer, L., & Wille, R. (2023a). Equivalence checking of parameterized quantum circuits: Verifying the compilation of variational quantum algorithms. *Asia and South Pacific Design Automation Conf.* https://doi.org/10.1145/3566097.3567932

Peham, T., Burgholzer, L., & Wille, R. (2023b). On optimal subarchitectures for quantum circuit mapping. *ACM Transactions on Quantum Computing*. https://doi.org/10.1145/3593594

Sander, A., Burgholzer, L., & Wille, R. (2023). Towards Hamiltonian simulation with decision diagrams. *Int'l Conf. On Quantum Computing and Engineering*. https://doi.org/10.1109/QCE57702.2023.00039

Schmid, L., Locher, D., Rispler, M., Blatt, S., Zeiher, J., Müller, M., & Wille, R. (2024). Computational capabilities and compiler development for neutral atom quantum processors - connecting tool developers and hardware experts. *Quantum Science and Technology*. https://doi.org/10.1088/2058-9565/ad33ac

Schmid, L., Park, S., & Wille, R. (2024). Hybrid circuit mapping: Leveraging the full spectrum of computational capabilities of neutral atom quantum computers. *Design Automation Conf.* https://doi.org/10.48550/arXiv.2311.14164

Sivarajah, S., Dilkes, S., Cowtan, A., Simmons, W., Edgington, A., & Duncan, R. (2021). T$|$ket$\rangle$: A retargetable compiler for NISQ devices. *Quantum Science and Technology*. https://doi.org/10.1088/2058-9565/ab8e92

Tsai, Y.-H., Jiang, J.-H. R., & Jhang, C.-S. (2021). Bit-slicing the Hilbert space: Scaling up accurate quantum circuit simulation. *Design Automation Conference*. https://doi.org/10.1109/DAC18074.2021.9586191

van de Wetering, J. (2020). *ZX-calculus for the working quantum computer scientist*. https://doi.org/10.48550/arXiv.2012.13966

Vinkhuijzen, L., Coopmans, T., Elkouss, D., Dunjko, V., & Laarman, A. (2023). LIMDD: A decision diagram for simulation of quantum computing including stabilizer states. *Quantum*. https://doi.org/10.22331/q-2023-09-11-1108

Wille, R., Berent, L., Forster, T., Kunasaikaran, J., Mato, K., Peham, T., Quetschlich, N., Rovara, D., Sander, A., Schmid, L., Schoenberger, D., Stade, Y., & Burgholzer, L. (2024). The MQT Handbook: A summary of design automation tools and software for quantum computing. *IEEE International Conference on Quantum Software*. https://doi.org/10.1109/QSW62656.2024.00013

Wille, R., & Burgholzer, L. (2022). Verification of quantum circuits. In A. Chattopadhyay (Ed.), *Handbook of Computer Architecture* (pp. 1–28). Springer Nature Singapore. https://doi.org/10.1007/978-981-15-6401-7_43-1

Wille, R., Burgholzer, L., & Artner, M. (2021). Visualizing decision diagrams for quantum computing. *Design, Automation and Test in Europe*. https://doi.org/10.23919/DATE51398.2021.9474236

Wille, R., Hillmich, S., & Burgholzer, L. (2020). Efficient and correct compilation of quantum circuits. *IEEE International Symposium on Circuits and Systems*. https://doi.org/10.1109/ISCAS45731.2020.9180791

Wille, R., Hillmich, S., & Burgholzer, L. (2023). Decision diagrams for quantum computing. In *Design Automation of Quantum Computers*. Springer International Publishing. https://doi.org/10.1007/978-3-031-15699-1_1

Wille, R., Hillmich, S., & Lukas, B. (2022). Tools for quantum computing based on decision diagrams. *ACM Transactions on Quantum Computing*. https://doi.org/10.1145/3491246

Younis, E., Iancu, C., Lavrijsen, W., Davis, M., & Smith, E. (2021). *Berkeley quantum synthesis toolkit (BQSKit) v1*. Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States). https://doi.org/10.11578/DC.20210603.2