

Quantum computers are coming! But how to use their potential?

Prof. Dr. Robert Wille

Design automation and software tools for quantum computing

We live in a world dominated by electronic systems: personal computers, smartphones, cars, industrial machines, and virtually every powered device. Those devices eventually became possible because, 80 years ago, electrical engineers built the first electronic circuits. Over time, they evolved the technology, made it scalable, powerful, and ubiquitous.

Current systems are composed of billions of components and are tremendously difficult to design and realise. In fact, it requires computer scientists and design automation expertise to enable the utilisation and penetration of electrical systems in almost all aspects of our daily life. We have programming languages, compilers, synthesis tools, verification methods, testing methods, debugging tools, and so on. Only with those methods can we tackle the tremendous complexity of designing the next smartphone or the next AI solution.

A new age of computing

However, we are currently at the dawn of a new computing age. Quantum computers—for many years just an academic dream—are becoming a reality. These machines have the potential to solve certain tasks in minutes for which conventional (super-)computers would take millennia. Numerous quantum computing applications with a near-term perspective (e.g., in finance, chemistry, machine learning, optimisation) and a long-term perspective (i.e., in cryptography, database search) are currently investigated. Numerous research groups, established companies and new startups are building increasingly powerful quantum computers—and making them accessible for everyone.

The quantum systems they are developing work in a completely different fashion than conventional devices. Rather than just dealing with 0 and 1, quantum computers work with quantum bits or qubits—which can assume any value between 0 and 1. Quantum-mechanical effects such as superposition or entanglement provide

powerful capabilities, but they also lead to new challenges. Due to the radically different computational primitives, seemingly simple tasks in the design of corresponding computers and algorithms get substantially harder for quantum computing compared to conventional circuits and systems. This affects how we currently conduct design automation for quantum computing—or, more accurately put, how we do not.

In fact, our established programming languages, compilers and design tools, as well as test and verification schemes do not work for quantum computers anymore. In many aspects, considering the design for quantum computers, we are back at square one. And this may lead to a situation where we have quantum computers but no proper (automatic) methods that aid us in using their potential!

The DA QC project

The ERC Consolidator Grant project DA QC (Design Automation for Quantum Computing) aims to develop design automation methods and software tools for quantum computing that will help avoid such a situation. It receives its motivation from the success witnessed in the design and realisation of conventional computing systems, where design automation and software tools play a pivotal role that eventually made most of today's electronic systems possible. At the same time, the project is aware that simply adapting conventional solutions won't be enough for quantum computing. The need to work with quantum-mechanical entities rather than electrical entities imposes several new challenges that need to be addressed.

Main challenges

The development of design automation and software tools for quantum computing is in its infancy. To be successful, the following challenges need to be addressed:

Computational complexity

Many design tasks in quantum computing come with enormous complexity. Even presumably simple tasks such as a logic simulation (which has linear complexity in the conventional domain) suddenly yield exponential complexity in quantum computing. Design automation can help here. In the project, we develop proper data structures and methods that aim to exploit redundancy in the description and make the problem more manageable.

Terminology and formalisations

In the quantum domain, terminologies and formalisations are often unclear and ambiguous to software or design automation experts. This frequently led to situations where 'wrong problems' are addressed, or inappropriate models are used. The project aims at consolidating on that and providing a proper basis that allows to fully exploit the potential of design automation for quantum computing.

Interdisciplinarity

Quantum computers are developed and built by physicists. Their control requires electrical engineers. Their potential is evaluated by theorists. And, eventually, computer scientists are needed to program and realise corresponding applications. A closer interaction between these communities is key to address the challenges just discussed. But bridging the gap between them requires lots of resources and time to exchange with experts and stakeholders of the field. Different 'languages' as well as 'cultures' in the respective fields additionally impede interactions. In the project, we are working in close cooperation with stakeholders to bring these communities together for their mutual benefit!

Conventional vs Quantum computing

Consider the tasks of simulating a quantum algorithm on a conventional machine—an important task in the early development of quantum applications, e.g., to test and evaluate an idea. Conceptually, this task is simple: a matrix describes the quantum operation, and a vector describes the initial quantum state. Matrix-vector multiplication then allows the resulting output state to be derived. However, the matrices and vectors we are talking about are huge—in fact, they are exponential with respect to the number of used qubits. In the project, we develop dedicated data structures such as decision diagrams. They exploit redundancies in matrices and vectors in a fashion which often allows a substantial reduction in the space required. First studies showed that, for some selected instances, established methods (relying on matrices and vectors) required 32 Gigabytes of memory for certain simulations, while the proposed decision diagrams allowed to simulate that very same instance with just 50 Megabytes. Eventually, this could make the difference between the need for a supercomputer or a simple notebook in order to run a simulation.

Another example is compilation, i.e., the task of realising a given quantum functionality for the different architectures available thus far and will be available in the near future. In the conventional domain, compilers are indispensable. But for quantum computing, existing solutions are in their infancy. Often, the first

". . . considering the design for quantum computers, we are back at square one. And this may lead to a situation where we have quantum computers but no proper (automatic) methods that aid us in using their potential!"

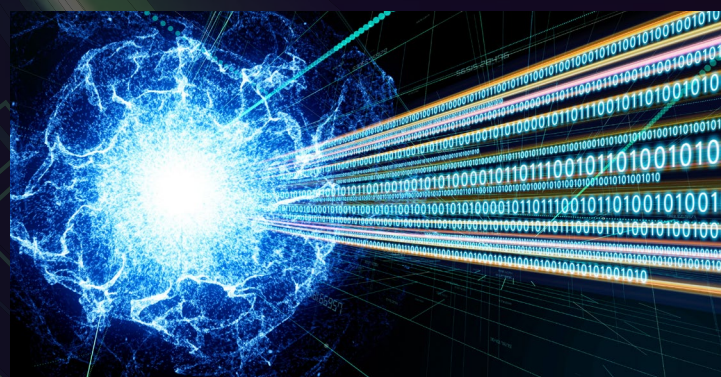
solutions considered the 'wrong models' which do not properly match constraints from actual physical realisations. Also, complexity is a challenge since many subtasks are shown to be computationally hard. Further efforts are needed to eventually get a suite of compilers that can handle this complexity and, additionally, is suited to realise various quantum computing applications for the corresponding machines.

Finally, let's consider verification, or more precisely, the task of checking whether the quantum circuit obtained, e.g., from a compilation process, indeed realises the originally intended quantum functionality. As in simulation, proper data structures such as the decision diagrams mentioned above can help here—after all, we need to compare functional descriptions of quantum computations. But, the project aims for more than that. We will investigate how characteristics of quantum computing can be exploited to compare two quantum circuits without ever having to compute their actual function and, therefore, never requiring an exponential amount of memory. If

done in the right way, this could enable us to verify the results of entire compilation flows in feasible time—a very intriguing prospect of utilising features that are not available in the conventional domain. Promising preliminary results are already available and have been incorporated into several open-source tools available to all researchers and engineers in the field.

Just the beginning

Corresponding case studies and evaluations confirmed the promise of these initial tools, but they are just the beginning. In the following years, we will develop a complete software stack that aids designers working in quantum computing and utilises design automation wherever possible. The long-term goal is to build a bridge between the design automation community and the quantum computing community and provide the foundation for design automation methods for quantum computing that conventional design automation realised for conventional circuits and systems in the past.



Adobe Stock © metamorworks

"In the following years, we will develop a complete software stack that aids designers working in quantum computing and utilises design automation wherever possible."

Software solutions

Software solutions for quantum computing that realise some of the ideas from the project have been made publicly available as open-source projects.

QFR: a library for quantum functionality representation

QFR provides the means for constructing the functionality of a given quantum circuit using decision diagrams in the form of a Python package. It is the backbone of the project's quantum software tools. QFR is available at <https://github.com/iic-jku/qfr>.

DD Package – A Package for Decision Diagrams

DD realises the idea of decision diagrams for quantum computing. In principle, quantum states and operations are represented on a conventional computer with an exponential number of complex amplitudes. Decision diagrams aim for a more compact representation that exploits redundancies to gain an efficient representation. In many cases, this enables us to represent quantum states and quantum operations much more compactly than, e.g., exponentially large state vectors and matrices, respectively.

DD is available at https://github.com/iic-jku/dd_package.

DDSIM is a quantum circuit simulator

DDSIM is based on decision diagrams. Efficient simulators are essential for the validation of future quantum computers and for designing quantum algorithms. Yet, quantum circuit simulations frequently suffer from the exponentially large representation of quantum states and operations. Many simulations can be performed substantially faster using decision diagrams rather than other state-of-the-art solutions.

DDSIM is available at <https://github.com/iic-jku/ddsim>.

QMAP: a tool for quantum circuit mapping

QMAP uses several methods for mapping quantum circuits to different architectures. After a quantum circuit (i.e., a quantum algorithm) is synthesised, it must be mapped to a physical quantum computer. This constitutes a non-trivial task since physical architectures require certain constraints to be satisfied. The tool uses informed-search heuristics as well as reasoning engines such as SMT solvers to tackle this problem.

QMAP is available at <https://github.com/iic-jku/qmap>.

QCEC: a tool for quantum circuit equivalence checking

QCEC realises several verification methods for quantum circuits. They can be used, for example, to check whether the originally intended functionality is indeed preserved throughout all levels of abstraction when compiling a quantum circuit for execution on an actual device. The methods are capable of efficiently verifying the results of compilation flows.

QCEC is available at <https://github.com/iic-jku/qcec>.

DDVis: a web-based tool for visualising decision diagrams

DDVis assists researchers and engineers without a background in using decision diagrams. It allows them to explore their behaviour when used in design tasks such as simulation, synthesis, and verification.

DDVis is available at https://iic.jku.at/eda/research/quantum_dd/tool/.

PROJECT NAME DESIGN AUTOMATION FOR QUANTUM COMPUTING

PROJECT SUMMARY

Quantum computing is becoming a reality, but automated methods and software tools for this technology are just beginning. This project aims to develop automatic and efficient methods, e.g., for simulation, compilation, or verification of quantum circuits. DA QC will exploit design automation expertise proven powerful for conventional circuits and systems yet hardly utilised in quantum computing.

PROJECT LEAD

Prof. Dr Robert Wille is a computer scientist by training but covers a broad spectrum of topics, with a particular focus on the development of automatic methods for the design, simulation, verification, and test of complex systems in hard- and software. He considers conventional technologies (from formal specifications to realisation) and future technologies (including quantum computing, microfluidic biochips, optical circuits, memristors, reversible circuits, and adiabatic circuits).

PROJECT PARTNERS

The project is executed at the Johannes Kepler University Linz (Austria), the Technical University Munich (Germany), as well as the Software Competence Center Hagenberg GmbH (Austria).

CONTACT DETAILS

Prof. Dr. Robert Wille
Johannes Kepler University Linz
Technical University Munich
Software Competence Center Hagenberg GmbH

+49 176 2344 0964

mail@rwille.de

www.rwille.de

@brtwwl



FUNDING

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No.101001318.