Rigorous Compilation for Near-Term Quantum Computers

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Razvan-Paul-Sebastian Brandhofer

aus Diemrich, Rumänien

Hauptberichter: Prof. Ilia Polian
Mitberichter: Prof. Robert Wille

Tag der mündlichen Prüfung: 15. Februar 2024.

Institut für Technische Informatik der Universität Stuttgart

2024

Contents

Acknowledgments

I express my deepest gratitude to everyone who supported me in the past years. I thank Prof. Ilia Polian for his profound support and for the opportunity to do research under his professional supervision and guidance. Furthermore, I want to thank Prof. Robert Wille from the Technical University of Munich for being my second adviser and for reviewing this work. I also thank the Center for Integrated Quantum Science and Technology (IQST) for funding my research and Stefanie Barz as well as Simon Devitt for providing an insightful entry point to the field of quantum computing.
My special thanks go to Eric Schneider and Michael Kochte who cultivated my interest in academia and who continued to be my mentors after they completed their doctoral degrees. I am grateful for everyone I had the pleasure to meet during my academic journey and who made my academic and personal life more entertaining and productive. In that regard, I especially acknowledge Daniel Bhatti, Simone d'Aurelio, Niklas Gaiser, Shiang-Yu Huang, Jeldrik Huster, and Dimitra Kanta from the IQST as well as Claus Braun, Li-Wei Chen, Nourhan Elhamawy, Maël Gay, Biswash Ghimire, Yanjun Ji, Florian Klemme, Chang Liu, Natalia Lylina, Florian Neugebauer, Laura Rodríguez Gómez, Alexander Schöll, Denis Schwachhofer, Roshwin Sengupta, Devanshi Upadhyaya, Chih-Hao Wang, and Tarick Welling from the Institute für Technische Informatik.
It was a pleasure to experience and contribute to research at the Kyutech Institute of Technology and at IBM Quantum. I thank Prof. Xiaoqing Wen, Prof. Kohei Miyase, Stefan Holst, Akari, Elouan, Miwa, Nathan, Wang and Yufei as well as Alexandra, Christophe, Daniel, Florian, Friederike and Kevin for making these research visits a productive and entertaining endeavour.
I thank Alexander Ivrii and Philipp Göttlich for their swift and in-depth proofreading of this thesis. Conducting research would not have been possible for me without the continued administrative and technical support provided by Mirjam Breitling, Helmut Häfner and Lothar Hellmeier. Finally, I thank my parents, family, and friends, with a special mention to Lisa, for their continued support and encouragement.

Stuttgart, December 2023

Sebastian Brandhofer

Abstract

Quantum computing promises an exponential speedup for computational problems in material sciences, cryptography and drug design that are infeasible to resolve by traditional classical systems. As quantum computing technology matures, larger and more complex quantum states can be prepared on a quantum computer, enabling the resolution of larger problem instances, e.g. breaking larger cryptographic keys or modelling larger molecules accurately for the exploration of novel drugs. Near-term quantum computers, however, are characterized by large error rates, a relatively low number of qubits and a low connectivity between qubits.

These characteristics impose strict requirements on the structure of quantum computations that must be incorporated by compilation methods targeting near-term quantum computers in order to ensure compatibility and yield highly accurate results. Rigorous compilation methods have been explored for addressing these requirements as they exactly explore the solution space and thus yield a quantum computation that is optimal with respect to the incorporated requirements. However, previous rigorous compilation methods demonstrate limited applicability and typically focus on one aspect of the imposed requirements, i.e. reducing the duration or the number of swap gates in a quantum computation.

In this work, opportunities for improving near-term quantum computations through compilation are explored first. These compilation opportunities are included in rigorous compilation methods to investigate each aspect of the imposed requirements, i.e. the number of qubits, connectivity of qubits, duration and incurred errors. The developed rigorous compilation methods are then evaluated with respect to their ability to enable quantum computations that are otherwise not accessible with near-term quantum technology.

Experimental results demonstrate the ability of the developed rigorous compilation methods to extend the computational reach of near-term quantum computers by generating quantum computations with a reduced requirement on the number and connectivity of qubits as well as reducing the duration and incurred errors of performed quantum computations. Furthermore, the developed rigorous compilation methods extend their applicability to quantum circuit partitioning, qubit reuse and the translation between quantum computations generated for distinct quantum technologies.

Specifically, a developed rigorous compilation method exploiting the structure of a quantum computation to reuse qubits at runtime yielded a reduction in the required number of qubits of up to 5x and result error by up to 33%. The developed quantum circuit partitioning method optimally distributes a quantum computation to distinct separate partitions, reducing the required number of qubits by 40% and the cost of partitioning by 41% on average. Furthermore, a rigorous compilation method was developed for quantum computers based on neutral atoms that combines swap gate insertions and topology changes to reduce the impact of limited qubit connectivity on the quantum computation duration by up to 58% and on the result fidelity by up to 29%. Finally, the developed quantum circuit adaptation method enables to translate between distinct quantum technologies while considering heterogeneous computational primitives with distinct characteristics to reduce the idle time of qubits by up to 87% and the result fidelity by up to 40%.

Zusammenfassung

Quantenberechnungen versprechen eine exponentielle Laufzeitverbesserung für das Lösen von Problemen in der Materialforschung, Kryptografie und der Arzneimittelentwicklung, die durch klassische Rechnersysteme nicht mit akzeptablem Aufwand lösbar sind. Durch fortwährende Verbesserungen in Quantenrechner-Technologien, können größere und komplexere Quantenzustände auf einem Quantenrechner präpariert werden, wodurch größere Probleminstanzen gelöst werden können. Zum Beispiel können größere kryptografische Schlüssel gebrochen werden oder größere Moleküle exakt modelliert werden, um neue Arzneimittel zu entdecken. Gegenwärtige Quantenrechner sind jedoch durch hohe Fehlerraten, eine relativ niedrige Anzahl Quantenbits und eine geringe Konnektivität zwischen Quantenbits charakterisiert.

An die Struktur von Quantenberechnungen werden durch diese Charakteristiken strenge Anforderungen gestellt, die in Kompilierungsmethoden für gegenwärtige Quantenrechner eingebracht werden müssen, damit die Kompatibilität gewährleistet ist und eine hohe Genauigkeit der Resultate erzielt werden kann. Für diese Anforderungen wurden bereits rigorose Kompilierungsmethoden untersucht. Diese durchsuchen den Lösungsraum exakt und bestimmen dadurch Quantenberechnungen, die optimal bezüglich der gestellten Anforderungen sind. Existierende rigorose Kompilierungsmethoden haben jedoch eine bedingte Anwendbarkeit und beschränken sich typischerweise auf einzelne Aspekte der auferlegten Anforderungen, zum Beispiel die Reduzierung der Laufzeit oder die Anzahl von Swap-Gattern einer Quantenberechnung.

In dieser Arbeit werden zunächst Freiheitsgrade der Kompilierung für die Verbesserungen von gegenwärtigen Quantenberechnungen untersucht. Diese Freiheitsgrade werden in den rigorosen Kompilierungsmethoden verwendet, um alle Aspekte der auferlegten Anforderungen, das heißt die Anzahl sowie Konnektivität der Quantenbits, aber auch die

Laufzeit und die Anzahl anfallender Fehler, zu untersuchen. Die resultierenden rigorosen Kompilierungsmethoden werden dann in Bezug auf die Möglichkeit evaluiert, Quantenberechnungen durchzuführen, die andernfalls nicht mit der gegenwärtigen Quantenrechner-Technologie ausgeführt werden können.

Experimentelle Ergebnisse zeigten, dass die entwickelten rigorosen Kompilierungsmethoden die Berechnungsreichweite gegenwärtiger Quantenrechner erweitern können, indem sie Quantenberechnungen mit reduzierter Laufzeit und Fehlern generieren, die zudem reduzierte Anforderungen an die Anzahl und Konnektivität der Qubits hatten. Darüber hinaus wurde die Anwendbarkeit der entwickelten rigorosen Kompilierungsmethoden erweitert, um die Partitionierung, das Wiederverwenden von Qubits und die Übersetzung zwischen Quantenberechungen für unterschiedliche Quantenrechnertechnologien.

Im Besonderen hat die entwickelte rigorose Kompilierungsmethode, die die Struktur von Quantenberechnungen zum Wiederverwenden von Quantenbits ausnutzt, eine Verringerung der benötigten Quantenbits um Faktor fünf erbracht und die Fehler der Ergebnisse um bis zu 33% verringert. Die entwickelte Quantenschaltungspartitionierungsmethode verteilt die Ausführung einer Quantenberechnung optimal auf separate Partitionen, wodurch die Anzahl benötigter Quantenbits um bis zu 40% und die Kosten für die Partitionierung durchschnittlich um 41% reduziert wurden. Weiterhin wurde eine rigorose Kompilierungsmethode für Quantenrechner auf der Basis von neutralen Atomen entwickelt. Diese Kompilierungsmethode kombiniert die Nutzung von Swap-Gattern und topologischen Veränderungen, um die Auswirkungen beschränkter Quantenbit-Konnektivität auf die Ergebnisgüte um bis zu 29% und die Dauer der Quantenberechnung um bis zu 58% zu verringern. Abschließend wurde aufgezeigt, dass die entwickelten Quantenschaltungsadaptierungsmethode befähigt, zwischen verschiedenen Quantenrechnertechnologien zu übersetzen, während heterogene Berechnungsprimitive mit unterschiedlichen Charakteristiken berücksichtigt werden. Dabei wurde die Leerlaufzeit von Quantenbits um bis zu 87% verringert und die Ergebnisgüte um bis zu 40% verbessert.

List of Figures

List of Tables

Abbreviations and Notation

$\wedge$	Logical conjunction
$\vee$	Logical disjunction
$\leftrightarrow$	Logical equivalence
$\rightarrow$	Logical implication
$\neg$	Logical negation
BV	Bernstein-Vazirani
CC	Classical communication
CNF	Conjunctive normal form
EDA	Electronic design automation
ESP	Estimated success probability
η	Displacement factor of a qubit
GHZ state	Greenberger–Horne–Zeilinger state
$G = (V, E)$	Graph with vertices V and edges E
KAK	Cartan's decomposition
$\mathcal{M}$	Formal model
NISQ	Noisy Intermediate-Scale Quantum
P	Set of physical qubits
Q	Set of program qubits
QAOA	Quantum approximate optimization algorithm
QPD	Quasiprobability decomposition
QFT	Quantum Fourier transform
SAT	Boolean satisfiability
SMT	Satisfiability modulo theories
T	Maximal number of considered time steps
UCCSD	Unitary coupled-cluster singles and doubles
VQE	Variational quantum eigensolver

Chapter 1

Introduction

Quantum computing promises advantages in computation time and scale of resolvable problems [Man80, F+18, Sho94]. For unstructured problems, such as searching in unstructured data or determining certain graph properties, a speedup in computation time up to the fourth power has been shown [ABDK+21]. Structured problems, including applications in machine learning [Aar15], integer factorization [Sho94] or chemistry [AL99, WKAGR08, AGDLHG05], can even yield an exponential speedup in principle. These speedups allow the computation of problems at an otherwise infeasible scale, thus, e.g. causing the need for novel cryptography protocols [BL17].

Furthermore, an accurate simulation of large molecules imposes prohibitive memory requirements on traditional computer systems [RWS+17]. On quantum computers, however, molecules can be represented more efficiently, paving the way for a profound simulation of large molecule interactions thus enabling the design of more effective drugs [CRAG18], improvements to industry-scale energy-inefficient chemical processes [FotUN17, RWS+17], or novel materials [BBMKLC20].

However, only large-scale quantum computers with full quantum error correction are guaranteed to exhibit these advantages [Pre18, RWS+17, Sho94, GE21]. Full quantum error correction schemes can improve the accuracy of a quantum computation to an arbitrary level given a sufficiently large quantum computer and an error rate that is below a certain threshold [KLZ98]. The error rate threshold for full quantum error correction has been proven to be at the order of 1% [RH07, FSG09]. As existing near-term quantum computers begin to meet this error rate requirement [IBM23d, AAB+19, PDF+21],

quantum error correction experiments on these computers have started to demonstrate a limited reduction in error rate [Goo23, SER+23]. However, processing hard large-scale computational problems with full quantum error correction imposes large resource requirements that are even prohibitive for quantum computers in the immediate near-term [CTV17, GE21]. For instance, attacking the ubiquitous RSA cryptography system [RSA78] at a contemporary key length of 2048 bits [fSidIB20] with a state-of-the-art quantum error correction protocol requires quantum computers that are five orders of magnitude larger at an error rate that is one order of magnitude lower than currently available [GE21, AAB+19, IBM23d, PDF+21]. Reaching full quantum error correction at scales relevant for solving practical problems thus poses grand research and engineering challenges for the foreseeable future [BDG+22, FMMC12, GE21].

Near-Term Quantum Computing

As full quantum error correction is not available in the immediate future, near-term quantum computers are explored to yield novel scientific insights and economic value. Near-term quantum computers are not expected to support large-scale quantum computations such as required for attacking RSA at contemporary key lengths [BDG+22, Pre18, FMMC12]. Given the large state space addressable on smaller-scale quantum computations, however, it is an open research question whether a near-term quantum computer can yield a scientific or economic value [BDG+22, Pre18]. There are already demonstrations on near-term quantum computers where non-trivial computations have been performed which require substantial runtime on traditional computer systems [AAB+19, PCZ22, KEA+23, BC23, TSSF24]. Despite these demonstrations not yielding a practical value yet, they highlight the potential of near-term quantum computers for solving practical problems at a scale that is infeasible or unprofitable to solve on traditional computers. Before near-term quantum computers can yield a practical value, however, challenges associated with this relatively nascent technology must be addressed.

Near-term quantum computers are limited in the type and scale of quantum computations they can perform without yielding erroneous results. The computational steps, i.e. operations, performed while solving a problem on a near-term quantum computer fail

with a considerable error rate [Pre18, LB20, BDP21]. In addition, the information stored by a qubit is corrupted over time [Sch19]. For these reasons, they are also described as noisy intermediate-scale quantum (NISQ) computers that, by definition, have a limited number of error-affected 'computational elements', called quantum bits (qubits), in the range of several dozen to several hundred [Pre18]. A quantum computer operates on these qubits to store and determine information about the solution to a problem it is configured to solve [NC11]. Quantum computers are realized on varying technology platforms, such as superconducting electronic circuits [IBM23d, AAB+19, Rig23], ion traps [BKM16, LWF+17], quantum optics [Bar15, KMN+07], nitrogen-vacancy [DMD+13], semiconductor spin [PRE+22, ZCL+20, WBB+14] or neutral atoms [BSK+17]. Incorporating the heterogeneous computational primitives supported by these various quantum computing technologies poses challenges for compilation methods. For instance, the compilation method may need to translate a quantum computation between distinct technology platforms to make it compatible with a target quantum computer. Furthermore, distinct sets of computational primitives within one technology platform lead to a significantly larger solution space that a compiler method must traverse to yield an optimized solution.

Challenge 1: Ubiquitous Errors

Due to the absence of full quantum error correction and the relatively nascent technology platforms used as a basis, errors are ubiquitous in near-term quantum computers [Pre18]. Errors are incurred by adverse physical processes that cause e.g. the inaccurate application of quantum operations [DMN13], or unwanted interactions with the environment such as electromagnetic fields [LWF+17] or radiation [VKO+20, Mar21]. When an error occurs, the state of a qubit is corrupted, i.e. the current qubit state deviates from the intended state, which ultimately leads to resolving a given problem incorrectly.

In near-term quantum computers, errors are primarily characterized for each type of applied operation and as a constant describing the decay of information on a qubit over time [KLR+08, MGE12, GJW+23, BPK23b, Sch19]. Thus, bounds on the type and number of operations in a quantum computation and on the total duration of quantum computations can be quantified for a given quantum computer [Pre18, BDP21]. For instance, the qubits

currently available in the `ibmq_ehningen` quantum computer provided by IBM Quantum decay exponentially at a time constant of roughly a hundred microseconds [IBM23b]. Furthermore, the error rate is in the one per cent zone with deviations of up to one order of magnitude possible depending on the exact type of operation [IBM23b]. At a one per cent error rate, one error with a potentially catastrophic impact is expected every hundred operations. Thus, quantum computations on this device would need to be completed in much less than a hundred microseconds and in much less than a hundred operations to yield a reasonable result quality [BDP21, LB20, Pre18].

Besides this relatively coarse quantification of errors, more nuanced characteristics of quantum computations can have a significant impact on the expected result error. One such characteristic is the qubit idle time [JJAB+21, DTDQ21, BKNB23], i.e. the total time between the first and last operation on a qubit during which no operation is applied. During qubit idle time additional errors can occur that have a significant impact on the computation [JJAB+21, DTDQ21, BKNB23]. Furthermore, near-term quantum computers are prone to errors that only occur in certain operational contexts such as simultaneous operations or a certain order of operations [RPL+19, SPR+20, KW23].

Challenge 2: Limited Qubit Connectivity

Inter-qubit operations are a prerequisite for solving problems with a practical runtime speedup on quantum computers [NC11]. However, relevant technology platforms used for near-term quantum computers only support inter-qubit operations on a restricted subset of qubit-qubit pairs at scale, i.e. their qubit connectivity is limited [IBM23d, AAB+19, BKM16, LWF+17, KMW21, WHWH20, Bar15, KMN+07, DMD+13, BSK+17]. On these near-term quantum computers, the qubit connectivity ranges from all-to-all connected qubits in small-scale ion trap quantum computers, to three-dimensional grids in neutral atoms [BSK+17], and two-dimensional grids or less on superconducting electronic circuits [IBM23d, AAB+19].

This restricted qubit connectivity is in contrast to the requirements set by problems targeted by near-term quantum computing such as portfolio optimization where inter-qubit operations can occur on all pairs of qubits [BBD+23]. Resolving such problems will induce inter-qubit operations that are not directly supported by the qubit connectivity of a

quantum computer. In these cases, auxiliary instructions such as swap operations, topology shifts or quantum state teleportation must be inserted into the quantum computation [ZPW18, WBZ19, HZW21, BBP21, TC20]. In the worst case of linearly connected qubits on the quantum computer, a number of auxiliary instructions must be inserted per time step that scales quadratically in the number of qubits [Bri17]. The insertion of such a large number of auxiliary instructions incurs further errors as the duration and the number of operations in the given quantum computation are increased. A restricted qubit connectivity thus leads to a potentially prohibitive increase in errors, limiting the scope of near-term quantum computers.

Challenge 3: Scarcity of Qubits

Next to the error behaviour and the qubit connectivity, the number of available qubits is a major performance metric of quantum computers. Qubits are a scarce resource in near-term quantum computers, with recent prototypes allowing a quantum computation to utilize up to 127 qubits [KEA+23]. In general, yielding a quantum speedup and thus tackling problems of practical relevance, requires quantum computers to at least surpass the simulation capabilities of classical computers. As the memory space requirement doubles for every simulated qubit, a simulation involving 50 qubits already imposes a prohibitive memory requirement of 18 petabytes on a classical computer. However, this bound only holds for general quantum computations on error-free all-to-all connected qubits. A much larger number of qubits can be simulated efficiently if a non-zero error rate affects the computation [ZSW20], the computation consists of certain structures or is performed by applying a restricted set of operations [MT06, ZSW20, PCZ22, BC23, TSSF24, Gid21]. For instance, recently a quantum computation on a near-term quantum computer with 127 error-affected qubits was simulated efficiently on a classical computer [BC23, TSSF24].
Apart from being a prerequisite for a practical runtime speedup, an increase in the number of qubits can also enable quantum computations with a reduced duration or number of operations. For quantum computations realizing arithmetic functions, ancilla (or: spare) qubits can be used to store intermediate states yielding more compact quantum computations that are less affected by limited qubit connectivity [SM13]. Another example is ancilla qubits used to store intermediate measurement results that are acted upon sub-

sequently to reduce the duration of the quantum computation [WKST19, LG22]. Since qubits are a scarce resource in near-term quantum computers, ancilla qubits may often not be available to reduce the size of a quantum computation.

Challenge 4: Heterogeneous Operations

Near-term quantum computers are realized on vastly different quantum computing technology platforms. While established quantum computer vendors are currently focusing on superconducting electronic circuits [AAB+19, IBM23d, Rig23], alternatives are abundant [BKM16, LWF+17, KMW21, WHWH20, Bar15, KMN+07, DMD+13, BSK+17]. These technology platforms support varying computational primitives that can realize varying sets of operations natively. Consequently, a given quantum computation typically needs to be represented by the set of operations natively supported by the target quantum computer. Furthermore, the set of operations supported by a quantum computer can change over time [IBM23b].

The dynamics of supported operations poses challenges for the classical control software of a near-term quantum computer. First, as most quantum computers currently available to researchers are based on superconducting electronic circuits [AAB+19, IBM23d, Rig23], quantum computations are often designed with the characteristics and supported operations of this technology platform in mind [BKNB23]. Thus, a target quantum computation may be defined by a set of operations not available to a given near-term quantum computer, especially if the quantum computer is not based on superconducting electronic circuits [BKNB23]. In this case, the quantum computation must be adapted to the target technology platform while considering the potential novel limitations of that platform. Second, the set of operations supported by a quantum computer may be degenerate, i.e. there are at least two operations in that set with the same functionality but potentially differing characteristics. For a degenerate set of supported operations, the classical control software incurs a larger compilation effort to carefully select the operations best suited for realizing a given quantum computation [BKNB23].

Rigorous Compilation of Quantum Computations

With large-scale quantum error correction imposing prohibitive resource requirements on current and near-term quantum computers, the field has turned to methods that can mitigate the impact of errors in near-term quantum computing [TC20, EBL18, TBG17, KTC+19, BDG+22]. Before a quantum computation can be executed, it must be compiled into a form that retains its intended functionality and also satisfies the requirements of the target quantum computer in terms of qubit connectivity, types of supported operations or the number of qubits [TC20, BDP21, ZPW18]. Several degrees of freedom, i.e. compilation opportunities, are at the disposal of the compilation step to determine such a quantum computation with characteristics that reduce the impact of errors on the target quantum computer. Compilation opportunities arise for a given quantum computation, for instance, when inserting auxiliary instructions due to limited qubit connectivity or when adapting to a different set of supported operations. These compilation opportunities can be used to address the challenges of near-term quantum computers before potentially employing additional error mitigation methods.

Compilation for near-term quantum computers has been demonstrated to yield significant improvements in the number of operations, duration and errors of quantum computations by suitably exploiting compilation opportunities [JJAB+21, TC20, ZPW18, BPK23b, BBP21, BPK23a, BKNB23]. Hereby, an increase in performance metrics of near-term quantum computers could be observed [JJAB+21, TC20, ZPW18]. One notable example is the doubled size of successfully executed random quantum computations, as defined by the quantum volume protocol [CBS+19, JJAB+21]. Results such as these highlight the importance of compilation for near-term quantum computations. In addition to the mentioned benefits, the compilation of quantum computations is a precursor for many quantum error mitigation methods that require a quantum computation as an input [KTC+19, TBG17].

In rigorous compilation of quantum computations, methods are employed that are based on a formal description of the available compilation opportunities and the given quantum computation. These rigorous methods can then reason about the complete solution space of the considered compilation opportunities to determine a quantum computation that is optimal with respect to a description of a preferred quantum computation. Furthermore,

the rigorous compilation of quantum computations enables an investigation into the supplied compilation opportunities themselves, i.e. it allows the compiler to decide whether one particular compilation opportunity is more beneficial for a quantum computation than another. This allows us to shape future iterations of near-term quantum computers to better support one compilation opportunity in favour of another. As a byproduct of the rigorousness, these methods can also give a definite answer when a quantum computation can not be compiled up to specified limits e.g. within a maximal computation duration.

Although rigorous methods incur a large runtime overhead in general [MB08], they are often still applicable as the size of near-term quantum computations is limited [Pre18, BDP21]. Furthermore, the rigorous compilation of quantum computations can often be performed offline before intending to execute a quantum computation [JBP22, BPK23b]. This allows to execute runtime-intensive rigorous methods for hours without impacting the throughput or latency of the quantum computation at the time it is executed on a quantum computer.

Contribution of this Thesis and Organization

The main contributions of this work are depicted in Figure 1.1. The work at hand develops rigorous compilation methods for extending the computational reach of near-term quantum computers, i.e. to enable quantum computations whose requirements were not met by the target quantum computer before the compilation was performed. To reach this goal, compilation opportunities tackling the major challenges of near-term quantum computers are first identified and then incorporated into rigorous compilation methods. The rigorous compilation methods determine a provably optimal compilation of a quantum computation based on a formal description of the available compilation opportunities.

Specifically, compilation opportunities employing qubit reuse, qubit connectivity changes induced by topology shifts, quantum circuit partitioning and quantum circuit adaptation have been identified to remedy near-term quantum computing challenges. The scarcity of qubits and limited qubit connectivity can be addressed by qubit reuse and quantum circuit partitioning for quantum computations with a certain structure. Furthermore, technology platforms for near-term quantum computing based on neutral atoms can perform qubit

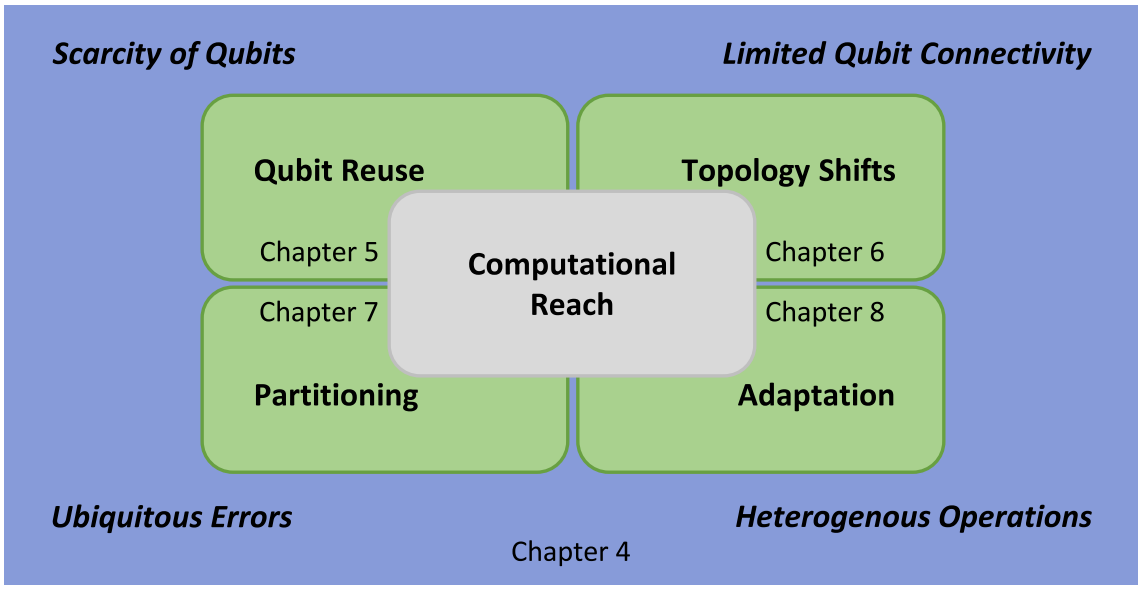


Figure 1.1: Main contribution of this work.

connectivity changes through topology shifts that can be used to address qubit connectivity challenges. As qubit reuse and the partitioning of quantum computations reduce the required number of qubits in a computation, they may also lower the overhead imposed by limited qubit connectivity. The heterogeneous operations arising in the nascent field of quantum computing are specifically addressed in quantum circuit adaptation. In addition, the introduced compilation methods can help to reduce the impact of errors in near-term quantum computers.

This work is structured into three parts. The first part (Chapters 2 and 3) gives the background on gate-based near-term quantum computers and introduces the rigorous compilation methods used in this work. The second part (Chapters 4 to 8) describes the main contribution of this work as follows:

Chapter 4 ("*Compilation Opportunities for Near-Term Quantum Computing Challenges*") — Here, compilation opportunities for addressing the challenges of near-term quantum computers are described and an overview of compilation methods is given.

Chapter 5 ("*Qubit Reuse*") — In this chapter, compilation opportunities based on qubit reuse are developed into a rigorous compilation method. As the availability of fast

high-quality qubit reset is mandatory for effective qubit reuse, the error behavior of qubit reset is first characterized on a near-term quantum computer.

Chapter 6 ("*Topology Changes Induced by Rydberg Shifts*") — Qubit connectivity changes induced by topology shifts are explored in this chapter for resolving the limited qubit connectivity. The typical approach of resolving qubit connectivity by only inserting certain quantum operations called swap gates is compared to employing topology shifts available with neutral atoms. This comparison yielded technology parameters for which topology shifts are strictly better performing than only inserting swap gates.

Chapter 7 ("*Quantum Circuit Partitioning*") — In this chapter, quantum circuit partitioning is developed and demonstrated to significantly alleviate a lack of qubits in near-term quantum computers. However, quantum circuit partitioning generally incurs an exponential increase in quantum computing time, limiting the scope of this approach and diminishing potential quantum speedups. Therefore, a rigorous minimization of the runtime overhead is developed and demonstrated to significantly outperform previous work on quantum circuit partitioning.

Chapter 8 ("*Quantum Circuit Adaptation*") — The nascent field of quantum computing gives rise to vastly different technology platforms that compete for the first realization of a practically relevant quantum computer. The heterogeneity of technology platforms also leads to the realization of heterogeneous sets of operations on these platforms. It is therefore a typical task to adapt a quantum computation given for one set of operations supported by a technology platform to another set of operations. In this chapter we combine multiple compilation opportunities for the adaptation of quantum computations in one rigorous compilation method.

In the third and last part (Chapter 9) the main contributions of this work are summarized and an outlook to future work is given.

Part I

Background

Chapter 2

Quantum Computing

In a quantum computation, the preparation, transformation, communication and measurement of quantum states are used to solve challenging computational problems [Sho94, CRAG18, FotUN17, BBMKLC20]. Different approaches to quantum computing have been developed that can be broadly divided into analogue and digital computations as well as universal and non-universal approaches [DC08].

In non-universal approaches such as adiabatic quantum computing [DSC19] or boson sampling [AA11, ZWD+20] not all quantum computations can be performed to arbitrary precision even when the computation is not affected by errors [Pre18]. In contrast, universal quantum computers allow to perform any computation to arbitrary precision [DBE95] given a sufficiently large size and an error rate that is below a certain threshold [KLZ98, Kit03]. In digital quantum computers, a given computational problem is resolved by performing a set of discrete steps [NC11] whereas in analogue approaches the quantum computation is decomposed into a continuous transformation [DC08]. Adiabatic [DSC19] and continuous variable [BvL05] quantum computing are examples of analogue quantum computing, whereas measurement-based quantum computing [BBD+09, Nie06], and gate-based quantum computing [IBM23b, AAB+19] are examples of digital quantum computing. In this work, the focus lies on gate-based quantum computers where a quantum computation can be represented by a graph-based structure called quantum circuit.

2.1 Quantum State

The basic unit of information in quantum computing is a qubit, which describes the state of a two-level quantum system [NC11]. From the initial state of a qubit, a target quantum state is prepared by operators that define the occurring state transformation in each step of a quantum computation. After the target quantum state is prepared, the qubit is measured to yield a classical value that helps to determine the solution to a given problem. Higher-dimensional d-level quantum systems that can be described by qu$d$its or infinite dimensional quantum states are not considered in this work [NC11].

2.1.1 Representation

The state of a qubit can be represented by structures such as (state) vectors [NC11], decision diagrams [MT06, ZHW19], tensor networks [MS08], or density matrices [NC11]. Decision diagrams and tensor networks can represent quantum states with certain properties more efficiently than a state vector or a density matrix [MT06, ZHW19, MS08]. A density matrix can represent a collection of quantum states where each quantum state has a probability to occur [NC11], which is typically used to accurately represent the effect of noise on a quantum state.

State Vector

The state vector of a one-qubit quantum state can be defined as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.1}$$

where $|0\rangle$ and $|1\rangle$ form the computational basis and α, β are complex values [NC11]. The complex values α, β are probability amplitudes that correspond to the measurement probabilities of the basis states, i.e. their squared magnitudes sum up to one $|\alpha|^2+|\beta|^2 = 1$ with $|\alpha|^2$ and $|\beta|^2$ corresponding to the measurement probability of the measurement value associated with $|0\rangle$ and $|1\rangle$.
The Dirac-notation [NC11] indicates n-dimensional vectors that correspond to a 'ket' $|\cdot\rangle$ or a 'bra' $\langle\cdot|$ of a quantum state such that $\langle\cdot|$ is the conjugate transpose of $|\cdot\rangle$ as $\langle\cdot| = |\cdot\rangle^\dagger =$

$(|\cdot\rangle^*)^T$. The following vectors can be used to represent the computational basis:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ with } \langle 0| \equiv \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \langle 1| \equiv \begin{pmatrix} 0 & 1 \end{pmatrix}. \tag{2.2}$$

An arbitrary quantum state can be represented as a linear combination of the elements in the computational basis as seen in Equation (2.1).

Density matrix

In contrast to state vectors, a density matrix can not only represent a pure quantum state but also a mixture of pure states, i.e. a mixed quantum state, where each pure state may occur with a certain probability. The density matrix is associated with the density operator:

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|, \tag{2.3}$$

where the probability of a quantum state $|\psi_i\rangle$ to occur is p_i. Let us assume that a mixed quantum state ρ' is prepared by flipping a fair coin and preparing either the $|0\rangle$ state or $|1\rangle$ state depending on the result of the coin flip. The outcome of this mixed quantum state preparation can be represented by:

$$\rho' = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 0 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.4}$$

The mixed state formalism is often used to represent the effect of errors that occur during quantum computation. Then, the probability of a pure state to occur in a mixed state is given by error mechanisms instead of a fair coin flip [DMN13]. Notice that the density matrix in general contains twice as many values as a state vector for the same state size.

2.1.2 Measurement

In order to measure a, possibly mixed, quantum state a set of measurement operators M must be chosen that sum up to the identity matrix I [NC11]. For measurements in the computational basis, the following measurement operators $P_i \in M$ constitute a valid

choice for M:

$$P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.5}$$

The probability of a result i when measuring a mixed state ρ subject to the measurement operators P_i is given by:

$$p(i) = \text{tr}(P_i^\dagger P_i \rho), \tag{2.6}$$

where $\text{tr}(\cdot)$ is the trace of a matrix, i.e. the sum of the elements on the main diagonal [NC11]. After measurement, the mixed state ρ collapses to

$$\sum_i \frac{P_i \rho P_i^\dagger}{p(i)}. \tag{2.7}$$

2.1.3 Transformations

Quantum state transformations can be divided into unitary operators, i.e. quantum gates, and non-unitary operators [NC11].

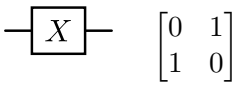
It is sufficient to consider unitary operators for universal quantum computing, i.e. the ability of a quantum computer to solve any problem a classical computer can solve, possibly at a speedup [NC11, DBE95]. A set of unitary operators allowing for universal quantum computation is called a universal gate set. A (quantum) instruction may describe a unitary operator, a non-unitary operator, measurements or other transformations that do not affect a quantum state such as topology changes.

Unitary Operator

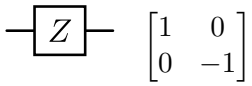
An operator U is unitary if $UU^\dagger = U^\dagger U = I$, i.e. operator U is reversible and preserves the inner product. Figure 2.1 depicts a selection of quantum gates often used in relevant works. A visualization of each quantum gate and its matrix representation is included in Figure 2.1. The R_z quantum gate and the R_x quantum gate together with the CNOT gate, CZ gate, or Toffoli gate are examples of universal gate sets. The action of a quantum gate g on a quantum state $|\psi\rangle$ can be represented by a matrix multiplication. For instance,

applying a Hadamard gate H to the computational zero state $|0\rangle$ yields:

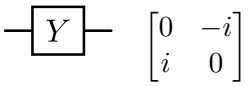
$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix} = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \quad (2.8)$$



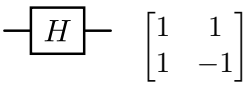
(a) Pauli-X gate



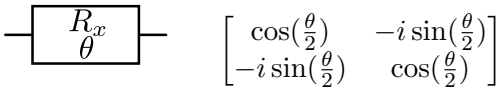
(b) Pauli-Z Gate



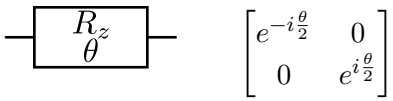
(c) Pauli-Y gate



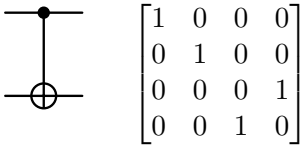
(d) Hadamard gate



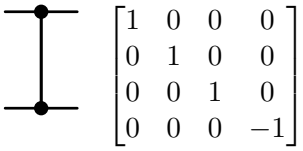
(e) $R_x(\theta)$ gate



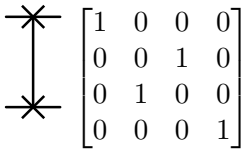
(f) $R_z(\theta)$ gate



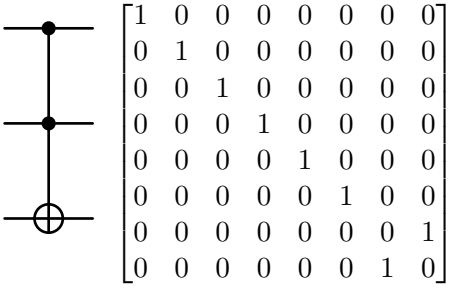
(g) Controlled NOT (CNOT or CX)



(h) Controlled Z (CZ)



(i) Swap



(j) Three-qubit Toffoli gate

Figure 2.1: Quantum gates with their typical visualization and their matrix representation.

Non-Unitary Operator

Analogous to measurements, a non-unitary operator is represented by a sum of operators rather than a single unitary matrix [NC11]. An example of a non-unitary operator well known from classical computers is the reset operation that sets a storage element, e.g. a register, to the zero value. In the quantum case, the reset operation sets an arbitrary

quantum state $|\psi\rangle$ to a computational basis state, i.e. typically the zero state $|0\rangle$. The action of the reset operation on a single-qubit quantum state can be represented by

$$O_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad O_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{2.9}$$

and is applied to the quantum state ρ' in Equation (2.4) by

$$\begin{aligned} \sum_i O_i \rho' O_i^\dagger &= O_0 \rho' O_0^\dagger + O_1 \rho' O_1^\dagger \\ &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = |0\rangle \langle 0| . \end{aligned}$$

Thus, the classical mixed $|0\rangle$ and $|1\rangle$ state in Equation (2.4) was transformed to the computational $|0\rangle$ state.

2.2 Near-Term Gate-Based Quantum Computer

Depending on an external signal that can e.g. be in the form of microwave pulses, an n-qubit quantum computer can retain, manipulate and measure an n-qubit state given by:

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle , \tag{2.10}$$

where α_x are complex probability amplitudes whose squared magnitudes sum up to one and $|x\rangle$ is a computational basis state (generalization of Equation (2.1) to n-qubits).
Measuring the quantum state $|\psi\rangle$ in the standard basis yields a measurement outcome k corresponding to the basis state $|k\rangle$ with probability $|\alpha_k|^2$. In addition, the quantum state $|\psi\rangle$ collapses to the basis state $|k\rangle$.
The realization of quantum computers is pursued on varying technology platforms, such as superconducting electronic circuits [IBM23d, AAB+19, Rig23], ion traps [BKM16,

LWF+17], quantum optics [Bar15, KMN+07], nitrogen-vacancy [DMD+13], semiconductor spin [PRE+22, ZCL+20, WBB+14] or neutral atoms [BSK+17]. These technology platforms offer various computational primitives that exhibit vastly different characteristics and can realize different basis gate sets and other native instructions such as topology changes (see Chapter 6) or the non-unitary reset operation.

2.2.1 Topology

A topology describes the available qubits and direct qubit-qubit interactions supported by a quantum computer. Typically, the topology is represented by a graph $G = (V, E)$ where V is the set of vertices representing the qubits of the quantum computer and E is the set of edges where an edge $e = (v, u) \in E$ represents an available interaction between the qubit represented by vertex v and the qubit represented by vertex u. The topology graph is also often called coupling map or connectivity graph and does not have to be a directed graph. The connectivity of a quantum computer can be indicated by various metrics. In this work, the connectivity of a quantum computer is defined by the average qubit connectivity, i.e. the average number of adjacent qubits, which corresponds to the average vertex degree in the topology graph. A pair of qubits are connected (to each other) if there exists a path in the topology graph connecting the vertices corresponding to the qubit pair.

2.2.2 Processing a Quantum Computation

Using a quantum computer to solve a computational problem involves several steps that are outlined in Figure 2.2. First, a quantum algorithm needs to be defined by potentially abstract steps that are performed to solve a given computational problem. Then, the quantum algorithm is transformed by a compiler procedure into a quantum circuit that can be executed directly by the target quantum computer. In a subsequent step, the quantum computer executes the determined quantum circuit, i.e. it performs the instructions contained in that quantum circuit and measures the hereby prepared state. This step is typically repeated by the quantum computer to yield sufficiently accurate information about the prepared quantum state that corresponds to the solution of a targeted computational problem. Variational quantum algorithms such as the variational quantum eigensolver (VQE) [PMS+14] and the quantum approximate optimization algorithm

(QAOA) [FGG14] form a computational cycle by introducing a dependency on the measurement results of the current cycle to the generation of the next quantum circuit to be executed. This computational cycle is performed until the measurement results converge or a computational budget is exceeded [PMS+14, FGG14].

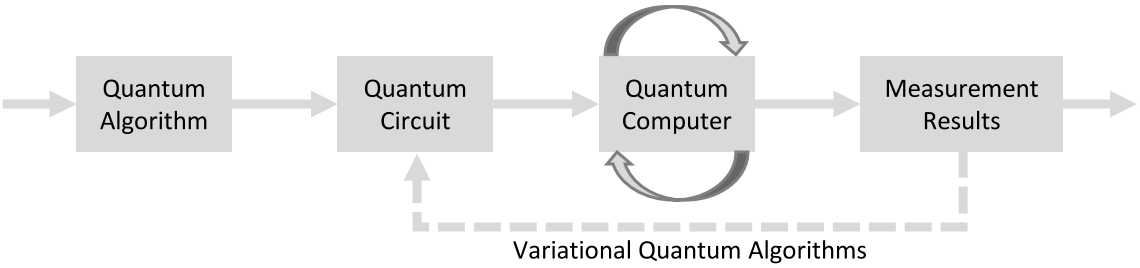


Figure 2.2: Steps for executing a quantum circuit on a quantum computer.

2.2.3 Calibration

Before any quantum circuit can be executed on a quantum computer, the qubits and the interactions of the qubits must be characterized during which the operational parameters of the instructions supported by the quantum computer are determined [DiV00, MAdCB20]. The operational parameters define how exactly the external signal, e.g. microwave pulses for superconducting qubits, for a quantum computer must be generated to perform the intended instruction. In practice and for current scaled-up quantum computers, these parameters are determined empirically by executing a set of quantum circuits that constitute calibration protocols [MAdCB20]. For quantum computers based on superconducting electronic circuits, the calibration protocol determines the frequencies, amplitudes and duration of microwave pulses that realize the basis gate set [MAB+18]. These operational parameters drift over time [KEBH+23, IBM23a] and can significantly deviate from one qubit to another. For instance, the gate duration of a CNOT gate can change by 3.2x from 181 ns to 587 ns during the same calibration window depending on the chosen CNOT pair. Therefore, frequent calibrations provide essential information for compilation methods developed for near-term quantum computers.

2.3 Quantum Circuit

All quantum computations on a gate-based quantum computer can be represented by a quantum circuit, i.e. quantum circuits are a universal description of such quantum computations [NC11]. A quantum circuit is a graph-based structure that models qubits, instructions applied to these qubits and the temporal order of these instructions. To distinguish from the physical qubits in a quantum computer, qubits in a quantum circuit are also called abstract, virtual, logical or program qubits. Furthermore, ancilla (program) qubits may be included in a quantum circuit. Ancilla qubits are not necessarily required to perform a given quantum computation but are used for purposes such as error detection, error correction, or more efficient realizations of instructions [FMMC12, OOCG20].

2.3.1 Representations

Figure 2.3 shows an example quantum circuit that is used to demonstrate the definitions in this section. In the graph of Figure 2.3, time flows from left to right, e.g. the CZ quantum gate (see Figure 2.1) depends on a CNOT quantum gate and one R_z quantum gate. A quantum circuit is typically represented by a graph where individual instructions correspond to vertices and the temporal order (or: dependency) of instructions is encoded in edges between these vertices. Instructions can depend on, i.e. be performed conditional to, a quantum state or classical value that may be the result of a previous measurement. Depending on the analysis or compilation task, a more detailed graph structure may be required (see e.g. Figure 7.7). For compilation purposes also other graph structures may be used for representing a quantum circuit [Qis23, vdW20].

2.3.2 Metrics

The structure of a quantum circuit can be quantified using several metrics. For one, the width of a quantum circuit amounts to the number of qubits in a quantum computer. The runtime of a quantum circuit on a quantum computer is often represented by the quantum circuit depth that corresponds to the length of the critical path in the graph of the quantum circuit. Another popular metric is the estimated success probability [NPS+20], which is also called estimated (circuit) fidelity [QBW23a]. This metric is defined by the product

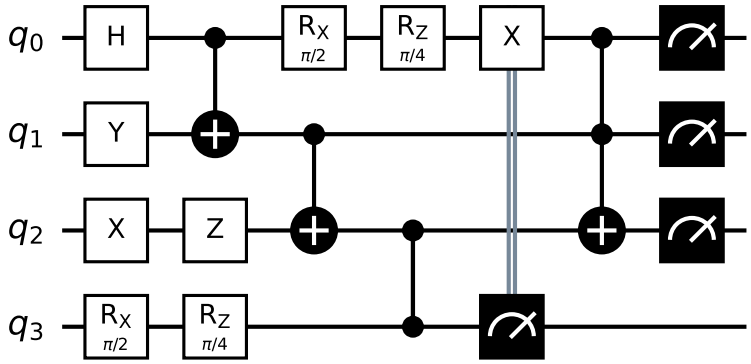


Figure 2.3: A 4-qubit quantum circuit with quantum gates, conditional operators and measurements.

of the quantum gates fidelities in the quantum circuit. In Figure 2.3, the depth of the depicted quantum circuit is seven and the width equals four.

2.3.3 Success Criteria

The successful execution of a quantum circuit can be determined by computing the fidelity, i.e. the overlap of the quantum state prepared by the quantum circuit with an error-free simulation [NC11], or by quantifying the frequency of one correct measurement result [BV97, BDP21] or a group of correct measurement results [JJAB$^+$21]. The fidelity of a mixed state ρ and a pure state $|\psi\rangle$ can be computed as:

$$F(\rho, |\psi\rangle) = \sqrt{\langle\psi| \rho |\psi\rangle} \tag{2.11}$$

However, when performing a quantum circuit on a quantum computer, the exact values of the probability amplitudes of the prepared state $|\psi\rangle$ are not known. Instead as in the previous Figure 2.3, measurements that yield limited information about a prepared state $|\psi\rangle$ must be conducted. Another widely-used [Qis23] measure of fidelity known as the Hellinger fidelity [Hel09] can help to determine the overlap of two sets of measurement results as

$$\sum_{i=1}^{k} \sqrt{p_i q_i}, \tag{2.12}$$

where p_i and q_i are the probabilities of measuring i in their corresponding set of measurement results. If a quantum circuit can not be simulated in a reasonable time or within

memory requirements, determining the success criteria or validating the execution of a quantum circuit is challenging and typically requires expertise in the problem domain [KEA+23].

2.4 Electronic Design Automation for Quantum Computing

Electronic design automation (EDA) is applied to a wide range of topics in quantum computing that range from designing physical realizations of quantum computers to designing quantum computations, i.e. quantum circuits, executed on a quantum computer [Sto21, RS20]. In EDA for quantum computers, the simulation and analysis of physical systems facilitating qubits and qubit state transformations, i.e. quantum gates, is typically combined with an automated optimized physical layout of components inducing these physical systems [MMD+21, AMOL22, MMT+21]. EDA for quantum circuits typically includes the analysis of quantum circuit characteristics such as the susceptibility to errors [BDP21, FMMC12], the simulation of quantum circuits using classical compute systems [NC11, MS08, ZHW19], the validation of a quantum circuit [BW20b, BW20a], and the compilation of quantum circuits [TC20, ZPW18].

Quantum circuit compilation transforms the description of a quantum algorithm to quantum circuits that are compatible with a given quantum computer. The description of a quantum algorithm can include abstract instructions such as "entangle a pair of qubits", a series of unitary matrices, a quantum circuit or a combination thereof.

A compatible quantum circuit may only contain operations that are in the basis gate set and are supported by the qubit connectivity of the target quantum computer. However, quantum circuit compilation often also applies secondary criteria to yield a preferred compatible quantum circuit with a lower expected error and a lower overhead in terms of qubits or time. Quantum circuit compilation attempts to generate a preferred quantum circuit by optimizing for specific quantum circuit characteristics such as quantum circuit depth in addition to ensuring compatibility [SSCP18, ZPW18, WBZ19, LDX19, CDD+19].

Figure 2.4 shows common quantum circuit compilation tasks for quantum computers based on superconducting electronic circuits. Quantum circuit compilation can be coarsely

divided into an offline part that can be performed well before a quantum circuit is executed by a quantum computer and an at-runtime (or: online) step that is informed by recent operational parameters of the target quantum computer [JBP22, QBW23b, DCKFF23]. Dividing the quantum circuit compilation in such a way can improve the throughput of quantum circuit executions significantly while still allowing the use of the most recent calibration data in the at-runtime step and allowing for a larger runtime on offline compilation tasks [TC20, WLD14, LDX19].

Figure 2.4: Typical quantum circuit compilation tasks for a quantum computer based on superconducting electronic circuits.

First, a synthesis task is performed in quantum circuit compilation. During synthesis, a possibly abstract description of a quantum algorithm or quantum computation is replaced by quantum gates that are included in the basis gate set of the target quantum computer. The synthesis step also determines the order of quantum gate executions and the program qubits they affect.

In many heuristic compilation algorithms, a preprocessing step is then conducted that determines a suitable initial qubit assignment [LDX19, ZPW18]. In this compilation task, each program qubit in the target quantum circuit is assigned to a physical qubit on the quantum computer for the first time step of the quantum circuit computation. The qubit assignment is adapted in subsequent time steps of the quantum computation if a quantum gate occurs on a pair of qubits that is not supported by the qubit connectivity of the target quantum computer. A subsequent change in qubit assignment may also change the time step in which a quantum gate is executed and the physical qubits it affects.

For instance Figure 2.5 shows the topology graph of a six-qubit quantum computer and a six-qubit quantum circuit containing six two-qubit gates. Consider the two-qubit quantum gates CNOT(1, 3) and CNOT(2, 4) between the qubit pair $1, 3$ and the qubit pair $2, 4$. They can not be performed directly by the quantum computer since its topology does not support these interactions. However, the quantum circuit in Figure 2.5 can be mapped by exchanging the qubit state of 1 with 4 before and after performing the CNOT gate on the

indicated qubit pairs. The states of a pair of qubits can be exchanged by inserting a swap gate on the specific qubit pair.

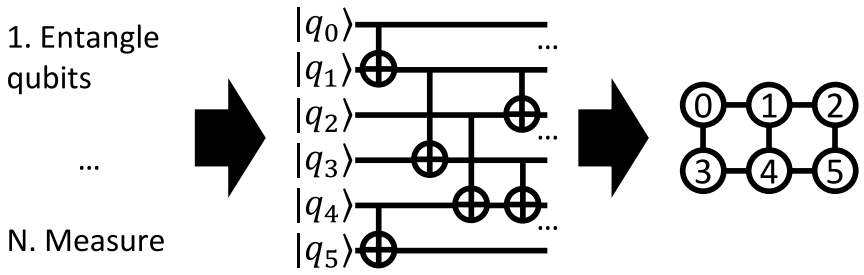


Figure 2.5: Synthesis of a quantum algorithm (left) to a six-qubit quantum circuit (middle) that is mapped to the topology (right) of a six-qubit quantum computer.

The first task in the at-runtime part of quantum circuit compilation is often an assignment of program qubits to physical qubits that are expected to incur low errors according to the most recent operational parameters [TQ19, NPS+20, NT23]. Due to the regularity of qubit connectivity in near-term quantum computers, the reassignment of qubits can often be performed without requiring an additional quantum circuit mapping step.

In the second last step of quantum circuit compilation, the schedule of a quantum circuit is fixed while considering the most recent quantum gate duration as defined by the operational parameters. Sustained periods of idling qubits can lead to additional qubit idle errors that can be reduced by determining a schedule with less qubit idle time or by inserting dynamical decoupling sequences into the hitherto compiled quantum circuit [BKNB23, NTS22b, NTS22a]. Scheduling may also reduce other types of errors such as crosstalk errors [MMMJA20]. The last step of quantum circuit compilation replaces the quantum gates in the hitherto compiled quantum circuit with the corresponding signals that are input to the quantum computer. For quantum computers based on superconducting qubits, this signal is composed of microwave pulses that manipulate the qubit state.

Note that there is currently no agreement on suitable abstraction layers for quantum circuit compilation and the depicted compilation tasks may be executed at different points of the compilation flow with different objectives. For instance, in a recent work circuit mapping is considered in conjunction with synthesis after a preliminary synthesis and initial qubit assignment has been conducted [LYW+23]. Furthermore, the initial qubit assign-

ment task is resolved in conjunction with the circuit mapping task in rigorous methods. Thus, the compilation tasks depicted in Figure 2.4 can only serve as an example.
Furthermore, additional steps can be employed during quantum circuit compilation to reduce the qubit requirement of a quantum circuit (e.g. see Chapters 5 and 7) or to translate a quantum circuit from one target quantum computer to another (e.g. see Chapter 8) and can thus serve as a replacement or as a subsequent step for the synthesis task. Other compilation tasks may exploit instructions that are only available on specific quantum computing technology platforms (e.g. see Chapter 6). Except for the method described in Chapter 8, the methods of this work expect the input quantum circuits to have been synthesized to a target basis gate set using e.g. any of the methods in [BKNB23, DST$^+$20, Tuc05].

Chapter 3

Formal Modelling

In formal modelling, systems and their interactions are rigorously specified using mathematical constructs [VL91]. This rigorous specification is then used as a basis to prove, analyze or simulate specific properties of the modelled systems [VL91]. Formal modelling has been successfully applied to problems in software engineering, biology, social sciences, security, scheduling, planning, quantum circuit compilation [CDM02, BKM07, Mil07, Lan81, Oli93, LPMMS16, MHL+13, TC20, BKNB23, BPK23b, BPK23a, BBP21, WBZ19]. In this section, first the preliminaries for formal modelling based on satisfiability modulo theories are described before highlighting the connection to rigorous quantum circuit compilation based on formal modelling and qualitatively comparing rigorous compilation to heuristic approaches.

3.1 Satisfiability Modulo Theories

Satisfiability modulo theories (SMT) generalize the Boolean satisfiability decision problem with theories such as linear integer arithmetic, fixed-size bit-vectors, or arrays [MB08]. An SMT solver is a software tool that decides the satisfiability of Boolean formulas extended with these theories, i.e. the solver yields a satisfiable assignment to the variables of the extended formula [MB08]. This SMT decision problem can be generalized to determining a satisfying assignment that is optimal with respect to a given objective function [BPF15]. As Boolean satisfiability is already NP-complete [Coo71], computing satisfiability modulo other theories may be NP-hard and may not be decidable depending on the involved theories [Ric68].

3.1.1 Theories

A theory $\mathcal{T}$ is a set of sentences that define the set of permissible expressions, i.e. the set of permissible variables and operators as well as their combination [End01]. A formula in theory $\mathcal{T}$ is a permissible expression that may involve one or more variables as well as one or more operators applied to subsets of these variables. A satisfying variable assignment to a formula is also called a model of that formula.

Boolean

A Boolean theory contains Boolean variables that can either be true (often associated with 1) or false (often associated with 0) and a set of logical operators [End01]. The set of logical operators that constitute a formula typically includes the unary negation operator ($\neg$) and the binary disjunction ($\vee$), conjunction ($\wedge$), implication ($\implies$) and equivalence ($\Leftrightarrow$) operators. However, modern solvers often only accept a canonical form of Boolean formulas called the conjunctive normal form (CNF) that can be generated efficiently from a Boolean formula [Tse83] and consists of a conjunction of clauses, i.e. disjunctions of variables and their negation [SNC09, MB08].

Linear Integer Arithmetic

A linear integer arithmetic theory contains variables that can be assigned integer values and linear arithmetic operators as well as inequalities [BN20, Bjø11]. The set of linear arithmetic operators is typically the addition ($+$) (including multiplication with constants) and subtraction ($-$) with (in)equalities describing whether an expression involving integer variables equals ($=$), unequals ($\neq$), is greater than ($>$), is at least as large as ($\geq$), is smaller than ($<$) or is at most as large as ($\leq$) as another expression of integer variables.

3.1.2 Solving

Deciding the satisfiability of a satisfiability modulo theories formula requires reconciling the theories involved in a formula by combining the individual theory solvers [BN20, Bjø11, MB08]. This is not always possible while retaining decidability, i.e. the

combination of decidable theories may not be decidable [BHvM09]. In case the combination is decidable, a formula involving multiple theories can be decided by first splitting the formula into distinct parts involving only one theory. These distinct parts are addressed by the corresponding theory solver while allowing communication between the individual solvers in the form of shared variables or equalities that can be enumerated a priori [NO79, Opp80] or updated iteratively depending on the current theory model [dMB08]. For Boolean theories several solvers have been proposed that are typically based on the DPLL algorithm [DLL62, BFP23, GZ17, MB08]. For a survey of solvers for linear integer arithmetic see [Mon16, BSW15, JDM13].

3.1.3 Optimization

A number of computational problems require optimization over the set of all satisfying assignments—determining just any satisfying assignment to a formula may not be sufficient [BPK23b, BBP21, BPK23a, BKNB23]. This optimization may be associated with variables or parts of a given formula. A variable-level optimization associates a cost or weight that is accumulated when a specified variable assignment is present, e.g. when a certain Boolean variable is assigned the truth value. Alternatively, a cost can be incurred when a part of the formula evaluates to a specific value. Again taking the example of Boolean theories in CNF, a clause that evaluates to false may incur a cost, which yields the MaxSAT problem of satisfying as many clauses as possible or minimizing the cost of clauses that are not satisfied [NB22]. Furthermore, the optimization over satisfying assignments can also be generalized by employing objective functions that take a model as an input and return the cost of such a model depending on the assigned variable values. Modern SMT solvers offer a range of optimization procedures that guarantee optimality over a combination of theories and a variety of objective function definitions [BPF15].

3.2 Application to Rigorous Compilation of Quantum Circuits

The structure of a quantum circuit can be modelled using Boolean and integer variables together with the logical and linear arithmetic operators introduced in the theories of this chapter. While a model of the circuit structure typically does not include a model

of the circuit functionality, modelling the circuit structure enables rigorous compilation when bundled with a set of compilation opportunities and a definition of a preferred quantum circuit, e.g. given as an objective function. For instance, the quantum circuit in the previous Figure 2.3 has 16 operations that can be modelled by 16 integer variables t_i denoting the time at which a quantum gate i is performed and by another 21 integer variables $x_{i,j}$ that describe the location, i.e. the qubits $j \in J_i$ on which a quantum gate i acting on $|J_i|$ qubits occurs. More integer variables are needed for the specification location than for the specification of time as there are two- and three-qubit gates where more than one qubit per gate must be defined as the location. Ensuring that a quantum gate g is computed after the computation of depending quantum gates $h \in H$ is completed can be formulated by the following set of equations:

$$t_g \geq t_h \quad \forall h \in H \tag{3.1}$$

Furthermore, a location may only be used once by any quantum gate in a time step:

$$(t_g = t_{g'}) \implies \bigwedge_{j \in J_g, j' \in J_{g'}} (x_{g,j} \neq x_{g',j'}) \tag{3.2}$$

This gives a complete, valid and fixed description of the quantum circuit structure that can be used to execute the quantum circuit on a quantum computer with all-to-all connectivity.

Figure 3.1 shows a conceptual flow chart on how this model of quantum circuit structure can be extended by equations that allow the rigorous optimization of an aspect of this quantum circuit. A description of a given quantum circuit and a target quantum computer is input to a procedure that generates an SMT formula. This SMT formula is extended and modified to represent a compilation opportunity of interest such as qubit reuse (Chapter 5), quantum circuit partitioning (Chapter 7), quantum circuit adaptation (Chapter 8) or the topology changes induced by shifts in Rydberg quantum computers (Chapter 6). Without an objective function to select a set of satisfying variable assignments, the extended formula may yield a quantum circuit that is less preferable than the original quantum circuit, i.e. circuit characteristics such as quantum circuit depth do not improve. Defining an objective function, however, allows to determine a preferable quan-

tum circuit that may include the modelled compilation opportunity if it improves original quantum circuit characteristics. The last step depicted in Figure 3.1 transforms the determined model to a quantum circuit with a valid structure. This is typically done by inspection, i.e. in linear time [TC20, BKNB23, BPK23a, BPK23b, BBP21].

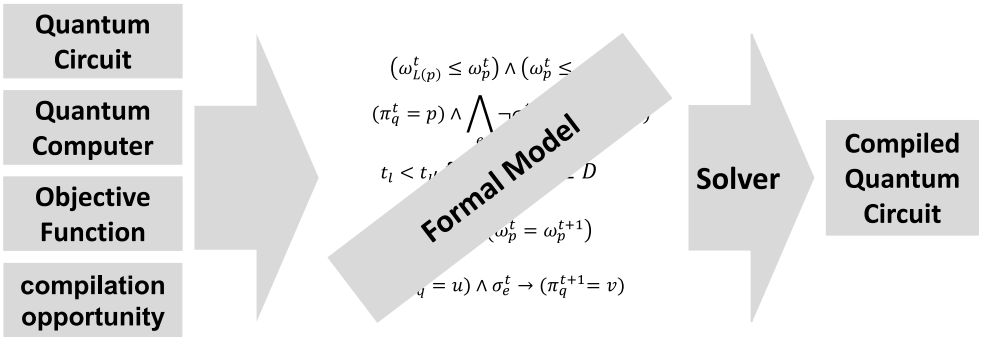


Figure 3.1: Rigorous compilation using formal modelling.

In contrast to rigorous quantum circuit compilation, heuristic approaches are not able to consistently find the optimal transformation of a quantum circuit given an objective function and a description of compilation opportunities. This inconsistency makes it challenging to establish a baseline for the best obtainable quantum circuit and to compare compilation opportunities directly with each other. When heuristic approaches are used to compare a set of compilation opportunities or circuit transformation directly, any yielded difference may be rooted in the heuristic approach itself instead of a difference in the employed compilation opportunities. However, optimality comes at the price of scalability — heuristic approaches are able to scale to large quantum circuits with hundreds or even thousands of qubits [LDX19, ZPW18], while rigorous approaches based on formal modelling are only applicable to small-scale quantum circuits with (currently) up to 50 qubits [LKT+23]. Since modern solvers addressing Boolean theories can solve formulas consisting of millions or even billions of clauses [Nad23, NB22], it seems plausible that further research into rigorous quantum circuit compilation may yield techniques that allow addressing larger-scale quantum circuits.

Part II

Rigorous Compilation for Near-Term Quantum Computers

Chapter 4

Compilation Opportunities for Near-Term Quantum Computing Challenges

In this chapter, the challenges of near-term quantum computers are discussed in detail while compilation opportunities for addressing these challenges are compared. Specifically, compilation opportunities for the scarcity of qubits, the limited qubit connectivity, ubiquitous errors and heterogeneous operations are described and an overview of existing approaches is given.

4.1 Scarcity of Qubits

The scarcity of qubits in near-term quantum computers is one of the main obstacles to fully error-corrected quantum computers [FMMC12, RWS+17, BDG+22]. For near-term quantum computers where full quantum error correction is not available, the number of available physical qubits also has a major impact on the size of problem instances that can be resolved. As near-term quantum computers exhibit a large variance in qubit quality, i.e. operations on subsets of qubits can be performed at a significantly higher fidelity than on other qubit subsets, solving a problem on a lower number of physical qubits may yield an overall better result. A paramount objective is therefore the reduction of the requirement a quantum computation poses on the number of physical qubits in the quantum computer. In the following sections, it is assumed that the domain expertise of a target problem has been fully exhausted to maximally reduce the qubit requirement. For instance, the graph can be pre-partitioned for instances of the maximum cut problem to yield a significant

reduction in qubit number [ZDTT23, LAG22]. In addition, a quantum computation can be compressed to require a lower number of qubits if the state space of the quantum computation is (partially) known [RMH+14, DLSP22, WCL+23]. Further techniques that compress a quantum circuit by using higher-level quantum systems (qu*d*its) instead of qubits are not considered [GBD+19, PF21, Fan07, MHW23].

4.1.1 Qubit Reuse

The state of a qubit is only required between the initialization and measurement [PWD16, NC11]. After measurement, the state of that physical qubit can be reset to a known basis state and subsequently reused by a different program qubit of the quantum computation. This compilation opportunity can thus reduce the number of required qubits for a quantum computation whenever a computation exhibits patterns that can be exploited by qubit reuse. As high-fidelity reset operations are a requirement for full quantum error correction in the long term [FMMC12], near-term quantum computers support this operation at an increasingly lower error. Thus, qubit reuse has been increasingly investigated on near-term quantum computations to reduce the qubit requirement [HJC+22, DCKFF23, PWD16, BPK23b].

However, qubit reuse can also have an adverse impact on a quantum computation. As one physical qubit is occupied with the operations of multiple program qubits, the duration of a quantum computation can increase, leading to further incurred errors due to decoherence. The incurred errors may initially be less relevant if a quantum computation that was infeasible due to the limited number of available qubits becomes feasible due to qubit reuse. The reduction in the number of qubits and increase in quantum computation duration yielded by this compilation opportunity is explored in Chapter 5.

4.1.2 Quantum Circuit Partitioning

In quantum circuit partitioning, a given quantum computation with a potentially infeasible qubit requirement could be divided into a number of smaller partitions that each contain a subset of operations from a given quantum computation [TTS+21, ALRS+20, BPK23a]. The number of required qubits can thus be reduced by selecting partitions with operations that act on disjoint subsets of qubits. These partitions can then be computed independently

from each other, i.e. there are no n-qubit quantum gates between partitions, and the results of these partitions are used to recover the result of the original computation.
While quantum circuit partitioning can be applied to arbitrary quantum computations, the runtime overhead increases exponentially in the effort required for partitioning. Hence, quantum circuit partitioning becomes prohibitive for larger quantum computations depending on the partitioning effort, i.e. the distribution of two-qubit gates between partitions. Therefore, it is in general crucial to minimize the effort for partitioning to reduce the impact on the runtime and thus to allow partitioning larger quantum computations. Quantum circuit partitioning for reducing the qubit requirement of computations is explored in Chapter 7.

4.1.3 Synthesis

When synthesizing unitary matrices or an abstractly defined functionality into quantum gates, there is often a range of options regarding the qubit usage in the synthesized quantum circuit. For instance, a single quantum full adder can be implemented by four qubits at depth 14 or by five qubits at depth four [OOCG20]. A given quantum circuit where the synthesis into quantum gates has already been determined can be inspected for opportunities to reapply synthesis with lower qubit cost in order to yield a qubit reduction. While synthesis with varying overheads in qubit-number exists for a wide range of functions such as the full adder [OOCG20], this approach can not yield qubit reductions in general quantum computations. Furthermore, these approaches rely on the ability to detect parts of the quantum circuit for which a synthesis with lower qubit requirements exists. This typically goes beyond direct template matching [IMM+22] as a large number of synthesis results may exist for a certain function.

4.2 Limited Qubit Connectivity

There is a stark contrast between the qubit connectivity given by the topology of near-term quantum computers and the qubit connectivity required by quantum computations one may want to perform. While this deficiency of near-term quantum computers can be addressed in principle [SSCP18], the overhead incurred by corresponding methods limits

the computational reach of near-term quantum computers [HJG+20]. Figure 4.1 depicts a collection of topology graphs representing the qubit connectivity of a range of quantum computers. The lowest qubit connectivity of a connected set of qubits is exhibited by a line graph as seen in Figure 4.1a where each qubit is at most connected to two other qubits. The H-graph in Figure 4.1b has the same average qubit connectivity as a line graph of the same size but yields a larger maximum qubit connectivity where one qubit is connected to three other qubits directly. Both, the H-graph and the line graph qubit connectivities have been realized on previous generations of IBM quantum computers [IBM23b]. In a square lattice (depicted in Figure 4.1d), a qubit is at most connected to four other qubits whereas the triangular lattice (depicted in Figure 4.1e) allows for a qubit to be connected to up to six other qubits. The square lattice has been realized on a quantum computer by Google during their quantum supremacy experiment [AAB+19] while the triangular lattice can be realized by quantum computers based on neutral atoms [EWL+21]. The largest quantum computers currently available for quantum computations provide a 'heavy-hex' qubit connectivity as seen in Figure 4.1f that is on average 5% higher-connected than the line graph [IBM21]. Quantum computers based on ion traps are in principle capable of performing operations between each pair of qubits as shown in Figure 4.1c at a small scale. With the notable exception of small-scale ion trap computers, all depicted topologies are planar, i.e. the edges denoting the qubit connectivity can be laid out on a plane without crossings.

In contrast, quantum computations such as those required for performing portfolio optimization require an interaction between each pair of qubits [BBD+23]. The quantum version of the Fourier transformation incurs an exponential speedup but also requires an interaction between each pair of qubits [COP94]. Grover's algorithm can yield a quadratic speedup in principle but requires a control-phase operator on a part of the algorithm called the diffusion operator [Gro97].

When the qubit connectivity supported by a quantum computer does not cover the required qubit connectivity, an additional effort is incurred that typically increases the size of the quantum computation in terms of depth, number of gates or required qubits. Thus, a mismatch in qubit connectivity between the quantum computation and the quantum computer causes an additional effort that limits the size of feasible quantum computations and

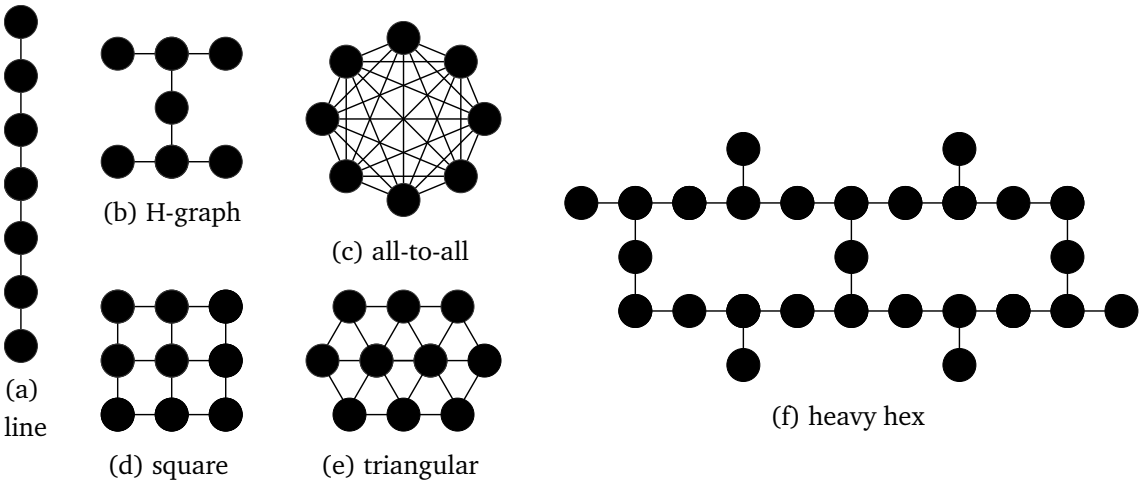


Figure 4.1: Selection of topology graphs in near-term quantum computers.

reduces the potential runtime advantage of quantum computations [HJG+20]. Several options to accommodate the limited qubit connectivity in near-term quantum computers are described in the following sections where it is assumed that a synthesis step already determined a suitable quantum circuit [VMB22, MvdGD23].

4.2.1 Swap Gate Insertions

The mismatch in qubit connectivity can be resolved by inserting swap gates into the quantum computation (see Section 2.4). During this compilation task, the states of directly connected qubits are successively exchanged until the qubit connectivity requirement in a specific time step of a quantum computation is satisfied. This process is repeated for each time step until the quantum computation only consists of qubit interactions that are satisfied by the qubit connectivity in a given quantum computer.

Swap gate insertion has been investigated by algorithms that rely on heuristics to scale for large quantum computations [LDX19, Qis23, ZPW18, TQ19, NPS+20, CDD+19] or that rigorously determine a swap gate insertion with minimum cost [TC20, SSCP18, WBZ19]. This approach of satisfying limited qubit connectivity has been explored in Chapter 6 and Chapter 5, where optimal swap gate insertion is considered alongside alternative mechanisms.

4.2.2 Bridging

The limited qubit connectivity of near-term quantum computers can also be accommodated by 'bridging' all two-qubit quantum gates that violate the qubit connectivity supported by the topology [NMD21]. When bridging a two-qubit quantum gate, the corresponding quantum gate is replaced by a set of two-qubit gates that satisfy the qubit connectivity and extend the effect of the two-qubit quantum gate to the desired qubit(s) without affecting the state of other qubits. Figure 4.2 shows a bridged CNOT gate on a set of qubits connected as a line graph (see Figure 4.1a). For each qubit over which the given CNOT is extended, two further CNOT quantum gates are inserted. While bridging a single two-qubit quantum gate is typically less costly than inserting swap gates, the cost of bridging multiple two-qubit quantum gates independently may offset the cost of inserting the necessary swap gates for all of these two-qubit gates [NMD21]. Thus, depending on the exact structure of a quantum computation, bridging or swap gate insertions may incur a smaller cost.

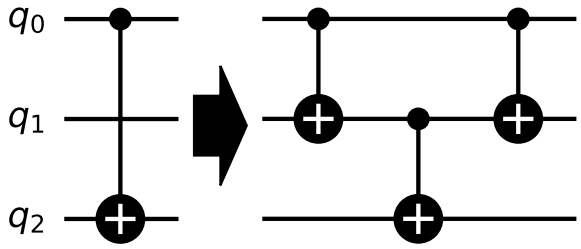


Figure 4.2: Bridging a CNOT over qubit q_1.

4.2.3 Quantum Teleportation

However, this requires the consumption of an entangled quantum state, performing two measurements and communicating the two measurement results [NC11]. This protocol can be employed to satisfy the qubit connectivity of a quantum computation [HZW21, HZW21, DSB+22] by first generating entanglement between specific pairs of qubits before the given quantum computation is executed. Then, when a two-qubit quantum gate violates the qubit connectivity of the quantum computer topology, the state of one qubit interacting in that two-qubit quantum gate is teleported to a neighbour of the other gate qubit before computing the two-qubit quantum gate.

While this approach can satisfy a required qubit connectivity in constant time [HZW21], it poses additional requirements on a target quantum computer. For one, the target quantum computer must contain ancilla qubits that can not participate in the quantum computation and need to stay entangled until they are consumed for teleportation. Second, the given quantum computer must support a fast, high-fidelity measurement of qubits as well as fast classical communication to apply conditional operations within the computer. While nowadays quantum computers begin to support these operations with high fidelity, it will be challenging to efficiently generate and sustain the required entanglement in the presence of limited qubit connectivity and relatively large error rates [BTW+23].

4.2.4 Changes in Topology

Instead of modifying the given quantum computation, the qubit connectivity of the target quantum computer can be modified e.g. in technology platforms based on neutral atoms [EBK+16, BBP21]. In these technology platforms, the atoms representing the qubit states can be physically moved in space in order to change the topology of a quantum computer and thus induce qubit connectivities that were not prevalent initially. The potential change in topology is typically restricted by the experimental setup of the target quantum computer, e.g. topology changes are investigated that can be induced by qubit movements along one axis in Chapter 6.
Topology changes can be beneficial depending on the exact technology parameters that determine the speed and fidelity of such a topology change with respect to alternatives such as swap gates [BBP21]. However, using topology changes for satisfying qubit connectivity requirements is still in its infancy with large-scale neutral-atoms quantum computers in development [BBP21, TBLC22].

4.2.5 Quantum Circuit Partitioning

The requirements on qubit connectivity could also be reduced by employing quantum circuit partitioning. By dividing a given quantum circuit into smaller independent partitions that can be computed individually, the requirements on qubit connectivity can be relaxed. While quantum circuit partitioning is explored in Chapter 7 for only minimizing the num-

ber of qubits required by a quantum computation, an additional (potentially secondary) objective could be used to lessen the requirement on qubit connectivity.

4.2.6 Synthesis

For certain classes of computations such as phase-polynomials [VMB22, MvdGD23], compilation methods can consider the qubit connectivity of a given quantum computer directly when synthesizing the quantum computation. By considering qubit connectivity in the synthesis step of a compilation, an insertion of auxiliary instructions such as swap gates can be avoided [VMB22, MvdGD23]. Furthermore, the synthesis of a unitary operator could be extended to be flexible with respect to induced qubit assignment changes [LYW+23]. Typically, a fixed qubit assignment is assumed during the synthesis of a unitary operator that may yield a larger cost for synthesis and lead to a worse qubit assignment [LYW+23].

4.3 Ubiquitous Errors

Near-term quantum computers are characterized by a larger error rate that effectively limits the size of quantum computations that can be faithfully computed. Figure 4.3 shows the CNOT error rate of the 27-qubit quantum computer `ibmq_ehningen` over a span of two and a half years as determined by calibration protocols performed by the quantum computing operator. The average CNOT error rate is depicted as a solid black line and the best and worst CNOT error rates in the device are depicted by the light blue region in the graph. Periods during which the `ibmq_ehningen` quantum computer was not available for computations are marked by red-striped rectangles.

As depicted in Figure 4.3, the average CNOT error rate ranges from 0.6% to 5.1% while the error rate varies by 0.2% to 100%, i.e. indicating a complete CNOT failure, during the recorded time. In addition to this significant change in CNOT error rate over time, the error rates of the qubit pairs at the same time step varied by a factor of 1.93x to 57.39x of the observed time span. The large variance in CNOT error rate in the same time span highlights the importance of carefully selecting a subset of qubits for a quantum computation (also see Chapter 5) [TQ19, NT23].

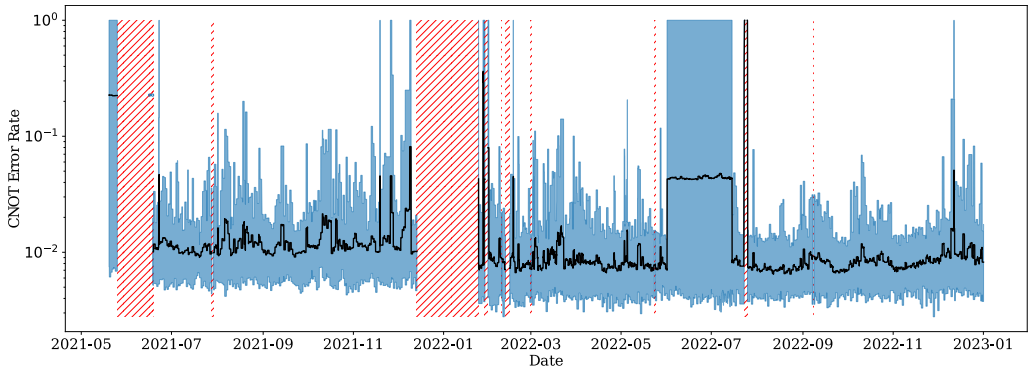


Figure 4.3: Two-qubit quantum gate error characteristics on the near-term quantum computer `ibmq_ehningen`.

Given these error characteristics, the question arises what result quality one can expect to extract from quantum computations at a given error rate without employing full quantum error correction. In Figure 4.4, this question is investigated for small-scale quantum computations that can still be simulated exactly subject to a uniform discrete error model where after each quantum gate in the computation, a single qubit Pauli gate (see Section 2.1.3) may be inserted randomly. The tolerated Pauli error rate is determined for a probability of at least 66% to satisfy the corresponding success criterion such as state fidelity, measuring the correct value, or quantum volume (see Section 2.3.3). This analysis was performed using exhaustive error simulation [BDP21]. The depicted quantum computations determine solutions for arithmetic functions [WGT+08], Grover's algorithm [Gro97], quantum Fourier transform (QFT) [COP94], the hidden linear function problem (HLF) [BGK18], the Bernstein-Vazirani (BV) algorithm [BV97]. In addition, two types of quantum 'ansatz' circuits are investigated (UCCSD [B+18] and RYRZ [KMT+17]) and the quantum volume (QV) protocol is evaluated subject to errors [CBS+19]. The tolerated error rate of all evaluated quantum computations scales inversely with their size. While there is a different offset associated with each family of quantum computations, all quantum computations require much less than one error to occur per execution to achieve a success probability of 66%. Specifically, HLF required the lowest error rate and QV allowed for the highest error rate at a success probability of 66%.

As seen in Figure 4.4, the characteristics of a quantum computation and the type of errors have a significant impact on result quality. Errors due to decoherence, for instance,

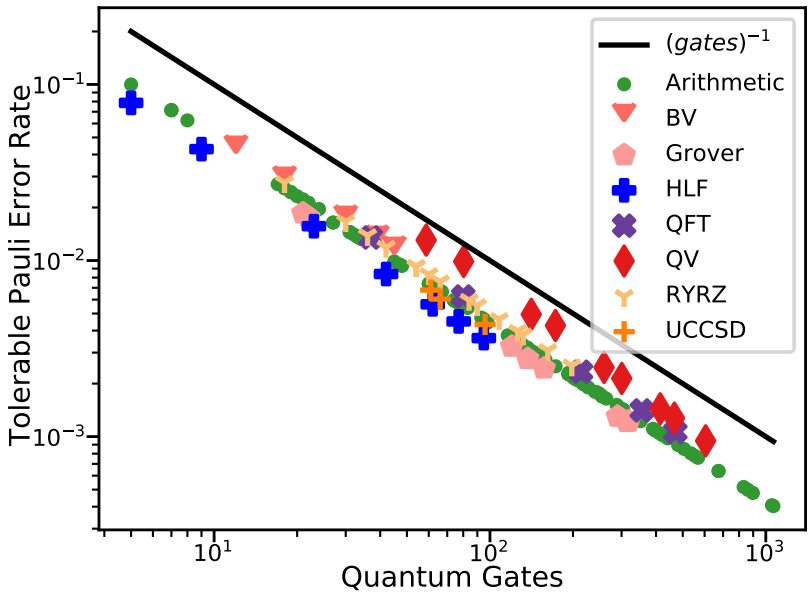


Figure 4.4: Tolerable Pauli error rate of small-scale quantum circuits for a success probability of 66%.

scale with the duration of the quantum computation, whereas other errors, arising from the control of quantum operations, occur with a certain chance when a quantum gate is applied. Further types of errors such as context-dependent errors or errors due to qubit idle time occur in more nuanced situations. Rigorous compilation can consider these error sources by generating quantum circuits that avoid incurring these errors.

4.3.1 Number of Quantum Gates and Quantum Circuit Depth

For independently and identically distributed (iid) errors such as the ones described previously in this section, the number of operations has a major impact on the result quality. It is therefore a common optimization objective to reduce the number of applied operations in a quantum computation [TC20, WBZ19]. In addition to the number of quantum gates, the runtime of a quantum circuit on a quantum computer is a relevant figure of merit as it indicates the incurred error due to decoherence. Typically, the depth of a quantum circuit (see Section 2.3.2) is used as a metric for the runtime of a quantum circuit instead of accurately modelling the duration of each individual gate [TC20]. The duration of a quantum gate is typically defined by the duration that an external signal must be applied

to the quantum computer in order to realize that quantum gate. In superconducting electronic circuits, this quantity corresponds to the duration of the microwave pulse for that quantum gate [IBM23b]. If the runtime of the quantum gates in the quantum circuit is roughly uniform, the depth of a quantum circuit is a sufficiently accurate metric for the minimization of quantum circuit runtime.
An objective based on the quantum circuit depth or the number of quantum gates can inform the steps performed for accommodating the limited qubit connectivity of quantum computers (see Chapters 5 and 6), the synthesis step in the compilation of quantum computations or translating a quantum computation from one technology platform to another (see Chapter 8). Furthermore, quantum circuit partitioning can also reduce the depth of a quantum circuit [PS23, Qis23].

4.3.2 Qubit Idle Time

A qubit is idle when no operation is acting on it. During the qubit idle time, the qubit incurs errors that are not directly captured by metrics such as the quantum circuit depth or the number of quantum gates in the quantum circuit [DTDQ21, NTS22b, NTS22a, JJAB+21]. Errors due to qubit idle time can be reduced by mitigating the effect of qubit idle time through e.g. dynamical decoupling sequences or by reducing the idle time itself. The insertion of dynamical decoupling sequences into the quantum circuit has been demonstrated to decrease the impact of qubit idle time [DTDQ21]. However, the insertion of these sequences increases the number of quantum gates in the quantum circuit, thus potentially diminishing the potential reduction in errors [NTS22b, NTS22a]. Depending on the quantum circuit, reducing the idle time itself may be preferable and has been investigated specifically in Chapter 8.

4.3.3 Context-Dependent Errors

Quantum computations may incur further errors subject to operational contexts such as the state of error-affected qubits or their neighbouring qubits [KW23], and previous, subsequent, or concurrent operations [RPL+19]. Quantum circuit compilation can address errors that depend on the temporal order of operations by rescheduling, i.e. determining a

different schedule in which operations are performed [MMMJA20]. Alternatively, dynamical decoupling sequences can be inserted into the quantum circuit to remove error-affected operations from their temporal context. Rescheduling can always be performed on operations that do not act on the same set of qubits. The order of operations on the same set of qubits can only be changed if these operations commute with each other [NC11]. State-dependent errors are challenging to address during quantum circuit compilation as the state of qubits at a specific time step is typically not known and computationally hard to determine for arbitrary computations. While context-dependent errors have not been investigated in this work, they can be addressed by rigorous compilation in principle [MMMJA20].

4.4 Heterogeneous Operations

Quantum computing can be realized on different technology platforms with distinct limitations and sets of basis quantum gates [Pre18]. The IBM quantum computers based on superconducting electronic circuits, for instance, support a single two-qubit CNOT quantum gate, while for semiconductor spin qubits, the entangling gates are typically CPHASE or CROT [PRE+22]. When a quantum circuit is generated for a given quantum computation, a gate set may have been chosen that is different to the technology platform intended for computation. In these cases, the generated quantum circuit must be adapted to the basis gate set of the technology platform intended for computation. Furthermore, the characteristics of the technology platform must be considered to yield a quantum circuit that can be computed with high fidelity. Here, we introduce several commonly used substitution techniques that can be used to adapt a quantum circuit from a source basis gate set to a target basis gate set.

4.4.1 Basis Translation

Using this substitution technique, quantum gates from a source basis gate set can be replaced by quantum gates from a target basis gate set according to an a priori-defined equivalence library. Such an equivalence library represents various ways of replacing one specific quantum gate with a set of quantum gates that exhibits the same functionality,

i.e. the same state transformation. For instance, a CNOT quantum gate can be replaced by a set of single-qubit gates together with a two-qubit quantum gate such as CZ or R_{ZX} [NC11]. If these gates occur in a given quantum computation, they can each be replaced by the CNOT quantum gate and single-qubit gates as defined by a substitution rule of the a priori-defined equivalence library.

4.4.2 Template Optimization

This substitution technique is typically used to determine a quantum circuit with reduced error or duration [ETE21]. During template optimization, three individual steps are performed in general. First, templates that define an equivalence relation between two distinct basis gate sets are defined and input to the substitution technique. A template is generally a quantum circuit defined by two sets of quantum gates that evaluate to the identity operation. Templates defined for adapting a quantum circuit consist of two parts that are functionally inverses but are based on different basis gates. Some example templates are shown in Chapter 8, specifically in Figures 8.2b to 8.2d. In a second step, the template occurrences in the given quantum circuit are determined, i.e. all maximal matches of the templates in the given quantum circuit are computed [ASWD13, MMD03, RDH14].
During the last step (i.e. template substitution), the matched part of the template is replaced by the inverse of the unmatched part if the unmatched part of the template has a lower cost. The cost of a part of a template can be evaluated with various metrics, such as error rate, or gate duration. As opposed to direct basis translation, template optimization offers the flexibility of converting between different sets of basis gates and subsequently yielding certain circuit patterns more effectively.

4.4.3 Synthesis of Unitary Matrices

In this substitution technique, a given unitary operator describing a set of quantum gates is synthesized into a sequence of single and two-qubit gates. For applications where arbitrary quantum gates occur, such as quantum volume circuits [CBS$^+$19], unitary synthesis is particularly useful to convert the unitary matrices of the arbitrary quantum gates to the set of supported basis gates in order to yield a compatible quantum computation. Several methods have been proposed to reduce the number of gates produced by unitary

synthesis, such as cosine-sine matrix decomposition (CSD) [MVBS04], quantum Shannon decomposition (QSD) [SBM06], and KAK decomposition [NKS06].

4.4.4 Suitability for Adapting a Quantum Circuit

The various quantum circuit substitution techniques introduced in this section are typically included independently in a quantum circuit compiler and work well on their own. However, the resulting adapted quantum circuit will have an increased error or duration if each technique is applied independently of the others. For instance, direct basis translation and unitary decomposition are typically only performed with one type of two-qubit basis gate which lacks flexibility when a combination of multiple two-qubit quantum gates is available to a given quantum computer and would improve the adapted quantum circuit even further. Adapting a quantum computation with template optimization, allows to incorporate multiple two-qubit basis quantum gates but limits the adaptation to local solutions as determined by the successive application of templates [IMM+22]. As multiple templates targeting different basis gates are not evaluated in conjunction in the matching phase of template optimization a globally optimal solution can not be reached. A clear method for combining these quantum circuit substitution approaches in an optimized way is the subject of Chapter 8.

Chapter 5

Qubit Reuse

Reusing the qubits of a quantum computer after the computations have concluded offers a compilation opportunity for rigorous compilation methods. For this compilation opportunity, reset operations are required in conjunction with measurements during the computation of a quantum circuit, i.e. mid-circuit measurements. The combination of these two operations is also the basis for many quantum error correction protocols [EWG$^+$18, FMMC12]. As full quantum error correction is the long-term goal of quantum computing, these operations are increasingly supported at relatively high fidelity and low duration by quantum computer operators such as IBM, Google and Honeywell [MPB$^+$16, Goo23, PDF$^+$21]. In principle, employing the reuse of qubits can yield the following benefits to the compilation of quantum computations:

- Reducing the number of qubits required for a quantum computation: a computation requiring n qubits may be compiled into a, potentially longer, computation that requires $n - i$ qubits.

- Accommodating the limited connectivity of near-term quantum computers: a reused qubit may require the insertion of a smaller number of auxiliary instructions (see Chapter 1) to satisfy connectivity requirements.

- Reducing errors: first, less auxiliary instruction insertions lead to less error [TQ19, WBZ19, IBM23b, BDP21]. Second, it was demonstrated in [TQ19, NT23, JBP22] that the error per qubit in near-term quantum computers varies significantly, i.e. a

part of the available qubits incur less error. Thus, more low-error qubits can be used by reducing the number of required qubits.

However, employing qubit reuse may also increase the error in quantum computations. For one, qubit reuse may increase the duration of the compiled quantum computation incurring a larger error due to decoherence [Sch19]. Second, the reset operations and mid-circuit measurements essential for qubit reuse exhibit a high variability in error (see Section 5.3). It is therefore crucial to carefully employ qubit reuse during compilation to improve the result quality yielded by near-term quantum computers. The compilation method developed in this chapter yielded a Hellinger fidelity improvement by up to 4.3x on the near-term quantum computer `ibm_hanoi`. Furthermore, the developed compilation method demonstrated the ability to replace swap gate insertions at a minimal increase in quantum circuit depth.
This chapter addresses the compilation of quantum computation by employing qubit reuse:

- The error of reset operations on a 27-qubit quantum computer is characterized while quantifying state-dependent and other context-dependent errors.
- A rigorous compilation method is developed that optimally considers swap gate insertions and qubit reuse to generate a quantum computation.
- The runtime-intensive rigorous compilation method is combined with a quick assignment of qubits [JBP22, NT23] that considers current reset errors at runtime.
- The obtained reset error characterization is also used to determine the number of reset operations per qubit that minimizes the error.
- The developed method is evaluated by executing compiled quantum computations on IBM quantum computers. This allows to investigate relevant characteristics of quantum computations such as the improvement in fidelity.

The remainder of this chapter is organized as follows. Section 5.1 gives the background to qubit reuse and describes an example compilation on a given 7-qubit Bernstein-Vazirani quantum circuit (see Figure 5.1) in detail. In Section 5.2 the related work on compilation

using qubit reuse is outlined. The reset error characterization experiments devised for obtaining the reset error rate are described and evaluated on an IBM quantum computer in Section 5.3. Then, the rigorous compilation method employing qubit reuse and the developed qubit assignment for incorporating reset error data are introduced in Section 5.4. The introduced rigorous compilation method is evaluated in Section 5.5 and the chapter is summarized in Section 5.6. Parts of this chapter have been published as [BPK23b].

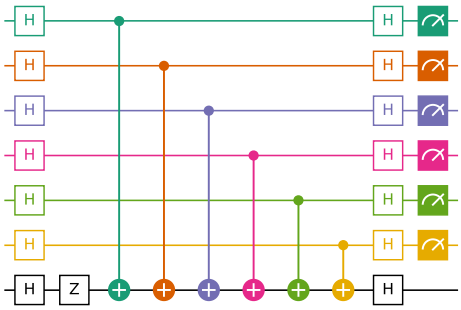


Figure 5.1: A 7-qubit Bernstein-Vazirani quantum circuit.

5.1 Qubit Reset and Reuse

As described in [PWD16], a qubit state must only be kept between initialization and measurement. This allows for a compilation opportunity where a qubit is reused after the interactions specified by a computation are concluded and thus a measurement is performed. Before a qubit can be reused, the qubit must be set to a known state. Typically the known state is the $|0\rangle$ state. The reset operation sets a qubit state to the $|0\rangle$ state on a qubit with an arbitrary, unknown state. After a previously used qubit is reset, it is able to store the state and perform the computations of a different qubit in the quantum circuit. This compilation opportunity requires fast, high-fidelity reset and mid-circuit measurement operations.

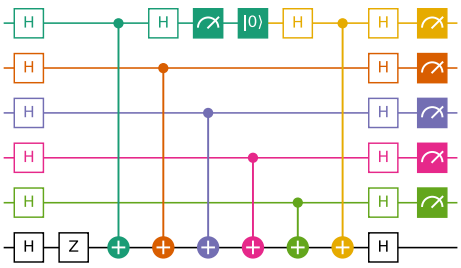
In Figure 5.2, a 7-qubit Bernstein-Vazirani (BV) quantum circuit is compiled using qubit reuse. As seen in Figures 5.2a to 5.2c, different quantum computations can be obtained when compiling with qubit reset. The original quantum computation (Figure 5.1), has the shortest duration if the quantum computer designated for execution is not limited in qubit connectivity. As near-term quantum computers exhibit a limited qubit connectivity at

scale, swap gates that increase the duration of a quantum computation would need to be inserted. For this example of the 7-qubit Bernstein-Vazirani quantum circuit in Figure 5.1, three swap gates would need to be inserted on the heavy-hex qubit connectivity of IBM quantum computers.

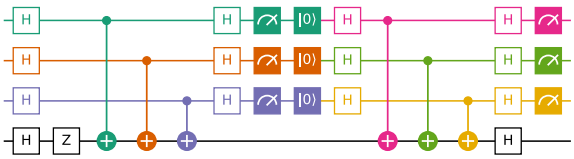
By reusing one qubit of the 7-qubit Bernstein-Vazirani quantum circuit in Figure 5.1, the quantum computation in Figure 5.2a is yielded. The first (cyan) qubit is reused by the computations on the last (yellow) qubit after executing one mid-circuit measurement and one reset operation. By reusing one qubit, the number of required swap gate insertions is reduced by one (assuming heavy-hex qubit connectivity) while not increasing the depth of the quantum circuit.

In the compilation of Figure 5.2b three qubits are reused while the number of required swap gates is now zero on the assumed heavy-hex qubit connectivity. For this compilation, however, the duration, as given by the quantum circuit depth, is increased by 40% compared to the original quantum computation Figure 5.1.

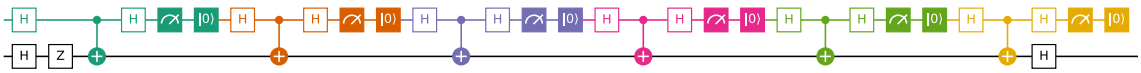
Lastly, the compilation of Figure 5.2d), reuses the maximum number of qubits at no swap gates even at linear connectivity. For this compilation, however, the duration of the quantum computation is also severely increased.



(a) Compiling with one reused qubit.



(b) Minimizing swap gates with qubit reuse.



(c) Minimizing the qubit requirement with maximal qubit reuse.

Figure 5.2: Qubit reuse compilation for the 7-qubit Bernstein-Vazirani circuit with different optimization objectives.

In this example, the different compilation opportunities using qubit reuse were demonstrated. Yielding a high-quality result on a near-term quantum computer requires a rigor-

ous compilation method with qubit reuse to consider the duration of a quantum computation, the number of swap gate insertions and the number of qubits.

5.2 Related Work

Since near-term quantum computers increasingly start to support the operations necessary for mid-circuit measurements and qubit resets [MPB+16, EWG+18, CTI+21, PDF+21, Goo23], recent approaches investigate quantum circuit optimization through exploiting qubit reuse [HJC+22, SKK23, DCKFF23, PWD16, BPK23b]. The related work can be coarsely distinguished into heuristics that can process large quantum circuits [HJC+22, SKK23, PWD16] and rigorous compilation methods where the focus lies on determining optimal quantum computations [DCKFF23].

In this chapter, a rigorous compilation method for qubit reuse is developed that simultaneously considers the duration and the incurred effort for accommodating limited qubit connectivity of the target quantum computation. The developed rigorous compilation method thus is able to produce a quantum circuit that can be directly executed on the target quantum computer. In a recent work on rigorous compilation for qubit reuse [DCKFF23], the effort for accommodating limited qubit connectivity is not considered. For this recent work, qubit reuse is not primarily used to replace the insertion of auxiliary instructions but only to reduce the qubit requirement. The insertion of auxiliary instructions must be determined independently in a secondary step [Qis23, LDX19, TC20, ZPW18, WBZ19].

Characterizing errors in quantum computers has been extensively investigated in [KLR+08, MGE12, GMT+12, KW23, BDWP21, EWP+19, GJW+23, RRG+22, MDS22, MPB+16, EWG+18]. Specifically, the error of individually applying reset operations on single qubits of IBM quantum computers has been analyzed in [MPB+16, EWG+18]. The reset operation on small quantum computers with up to 5-qubit has been characterized in [MDS22] where the security of reset operations was investigated. In contrast to the work in [MDS22], the errors of reset operations in a 27-qubit quantum computer are characterized and reported while detailing context-dependent errors [RPL+19].

5.3 Reset Characterization Experiments

In this chapter, three different experiments are devised to characterize the variability of reset errors and the impact of the operational context on the reset error. The evaluated operational context consists of the number of repeated reset operations, the state of the reset qubit and the simultaneous reset of qubits. An overview of the three devised reset error characterization experiments is given in Figure 5.3.

On the left of Figure 5.3, a characterization experiment (`Reset Rand.`) is depicted where qubits are individually initialized randomly and reset subsequently. Specifically, W random single-qubit gates $U_1, ..., U_W$ are drawn for each sample of the `Reset Rand.` characterization experiments. On each qubit $q \in P$, a random single-qubit gates U_i and the reset operation are applied $r \in \{1, ..., R\}$ times individually before concluding the computation on a qubit with a measurement operator. The characterization experiment `Reset Rand. (Simul.)` depicted in the middle of Figure 5.3, applies the steps of the previous `Reset Rand.` experiment simultaneously on all qubits of the target quantum computer. In the last characterization experiment (`Reset X Simul.`), instead of applying random gates, single-qubit Pauli-X quantum gates are applied on all qubits simultaneously. As in the previous characterization experiments, the reset operation is applied r times before concluding the computation on a qubit with a measurement. The reset error is quantified as the frequency of non-zero measurement outcomes at the end of each characterization experiment. Similarly, the reset fidelity is the frequency of zero measurement outcomes.

The different reset error characterization experiments vary in the number of required quantum circuit executions on a target quantum computer with $|P|$ qubits, W applied random gates and the maximum number of reset operation repetitions R. The largest effort in quantum circuit executions is given by the `Reset Rand.` experiment, yielding $|P| \cdot W \cdot R$ experiments. The `Reset Rand. Simul.` experiment incurs less effort at $W \cdot R$ executions while the `Reset X Simul.` experiment incurs the least effort at R executions.

In this set of characterization experiments, we expect to observe errors that depend on the initial state of a reset qubit. Evidence for these state-dependent reset errors is expected to emerge through a difference in quantified error from the `Reset Rand. Simul.` experiment to the `Reset X Simul.` experiment. In addition, the random quantum gate applications

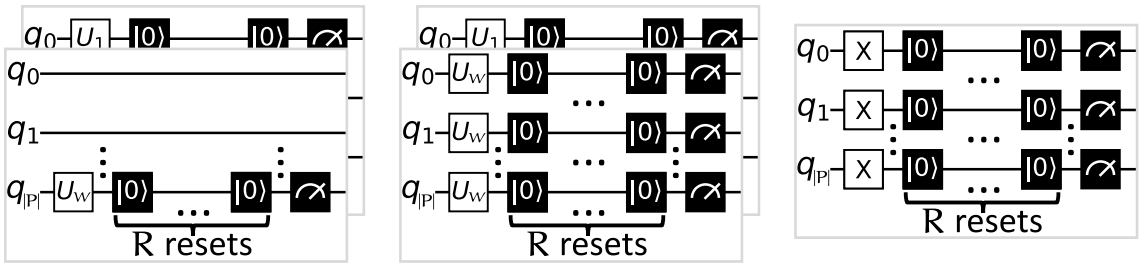


Figure 5.3: The three reset error characterization experiments described in this chapter, ordered by the number of required quantum circuits (left: `Reset Rand.`, middle: `Reset Rand. Simul.`, right: `Reset X Simul.`).

are expected to yield a different reset error rate depending on the initialized state if state-dependent errors occur. Without state-dependent errors, each random state initialization should yield the same reset error. Further context-dependent errors occurring when reset operations are applied simultaneously are investigated by comparing the experiments that prepare and reset a qubit individually (`Reset Rand.`) to the experiments that act on all qubits simultaneously (`Reset Rand. Simul.` and `Reset X Simul.`).

The `Reset X Simul.` characterization experiment requires fewer quantum circuit executions and is thus more efficient than the `Reset Rand.` experiment and the `Reset Rand. Simul.` experiment. However, this lower effort may come at the cost of a lower characterization accuracy. Therefore, it will be discussed whether the `Reset X Simul.` is sufficient to quantify the qubit reset error.

The characterization experiments devised in this chapter are performed on the 27-qubit quantum computer `ibmq_ehningen` to quantify the individual qubit reset error, in the remainder of this chapter. The different characterization experiments are interspersed to account for shifting error rates on the quantum computer during the evaluation that may occur e.g. due to calibration. As a consequence the `Reset Rand.` experiment is performed with $r \in R$ reset repetition and random single-qubit gate $U_1, ..., U_W$, followed by the `Reset Rand. Simul.` experiment and `Reset X Simul.` experiment. The same experimental parameters, i.e. random gate U_i and reset repetition r, are employed where applicable before continuing with the next set of parameters. The conducted experiments performed $W = 50$ random state initializations and up to $R = 5$ reset repetitions.

5.3.1 Reset Error Quantification on a Near-Term Quantum Computer

The fidelity of resetting a particular qubit of the `ibmq_ehningen` near-term quantum computer is depicted in Figure 5.4. Here, the fidelity after applying one reset operation is reported as quantified by the reset error characterization experiments introduced in the previous section. The fidelity of the reset operation is depicted on the y-axis of Figure 5.4 while the x-axis indicates the respective qubit of the `ibmq_ehningen` near-term quantum computer. The average reset fidelity of a qubit is marked by a red line for each reset error characterization experiment.

The evaluation shows significant outliers (depicted as grey circles) on most qubits. For instance, the reset fidelity on qubit 0 ranges from almost unit fidelity to less than 70%, depending on the employed reset error characterization protocol. Apart from the depicted outliers, the reset fidelity of the qubits shows little variance except for qubits 8, 12, 15, and 19. The best average reset fidelity of 98.98% was observed on qubit 24 while the best worst-case behaviour was observed on qubit 20 where a minimal reset fidelity of 96.6% was achieved over all reset characterization experiment runs. A completely correct reset with a fidelity of 100% was only occasionally achieved on qubit 21 during our experiments. The worst average reset fidelity of 88.57% was observed on qubit 8 while the worst reset was registered on qubit 0 with a fidelity of 67.55%. An outlier rate of at least 8.6% was shown for more than half of the qubits while qubit 0 was observed to have the largest rate of outliers (18%).

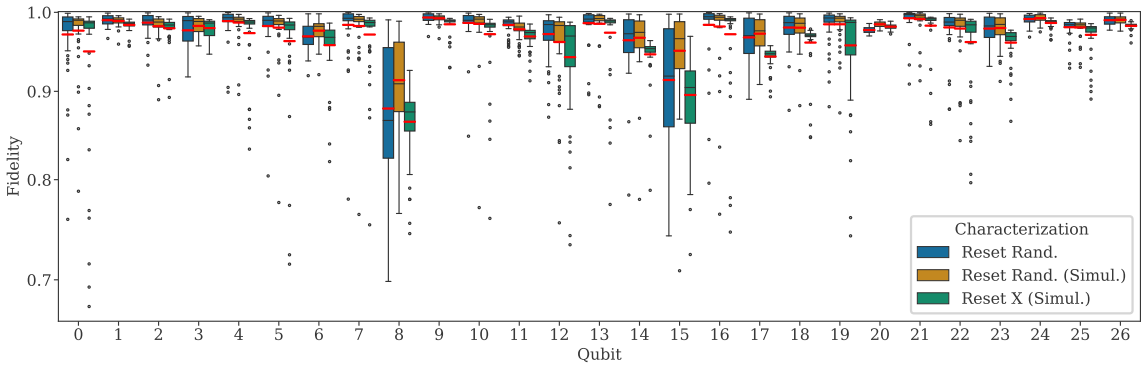


Figure 5.4: Fidelity of the reset operation for the 27-qubit quantum computer `ibmq_ehningen` as reported by the introduced reset error characterization experiments. A red line marks the respective mean value.

The `Reset Rand.` experiment reports the highest best-case fidelity in general while the `Reset Rand. Simul.` experiment and `Reset X Simul.` experiment report lower best-case fidelities. The `Reset Rand.` experiment reports a larger reset fidelity on most qubits than the other reset characterization experiments with the `Reset X Simul.` experiment reporting the worst reset fidelity. The average reset fidelity, however, is reported to be slightly larger by the `Reset Rand. Simul.` characterization experiment at 97.7% (+-3.2%) compared to the `Reset Rand.` characterization experiment at 97.5% (+-3.8%). We therefore conclude that for compiling with qubit reuse, errors due to concurrent reset operations are negligible.

5.3.2 Fidelity of Repeated Reset Operations

In Figure 5.5, the fidelity of repeated reset operations is shown on the y-axis (outliers omitted) as reported by the `Reset X Simul.` experiment for each qubit (x-axis) of the `ibmq_ehningen` quantum computer. The reset operation is applied r times with $r \in \{1, ..., 5\}$ on each qubit. As there is a large spread in reset fidelity for the different qubits and reset repetitions (depicted as Reset Rep. in Figure 5.5), a subset of qubits are registered on a second y-axis. Specifically, qubits 8, 12, 15, 19, and 22 are registered on the second y-axis that ranges from 78% to 99.9% and the remaining qubits are registered on the first y-axis that ranges from 93.3% to 100%. The worst reset fidelity is observed for more than 80% of the qubits when the reset operation is not repeated (blue bar). The fidelity is improved dramatically when the reset operation is repeated once (orange bar), where the reset infidelity is halved for some qubits such as qubit 17. A further increase in reset repetition did not improve the average reset fidelity of qubit 17. However, the largest average reset fidelity of qubit 23 was observed at 3 reset operation applications with a diminishing fidelity observed at further reset repetitions. As the reset fidelity varies significantly per qubit and reset repetition, we conclude that there is an optimal number of reset repetitions for each qubit that yields the largest average reset fidelity. Therefore, the application of qubit reuse to the compilation for near-term quantum computers should be combined with an assignment step where the qubits that currently exhibit the largest fidelity are selected for computation with the best repetition of the reset operation.

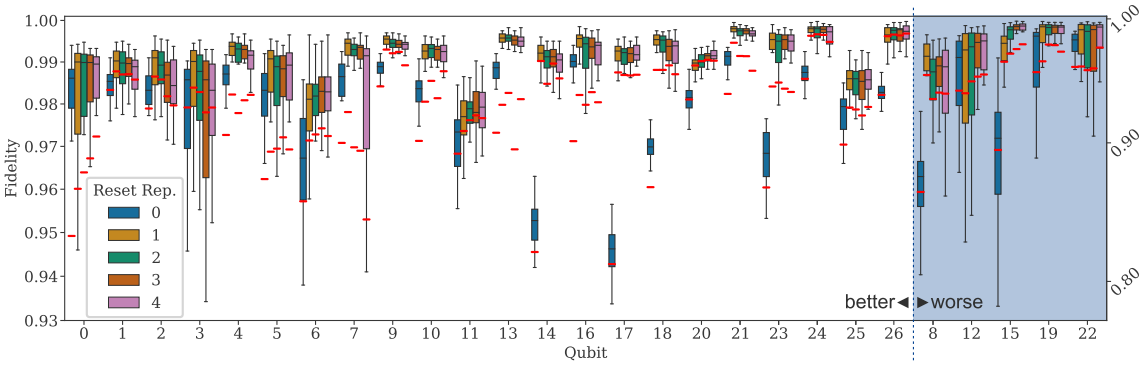


Figure 5.5: Fidelity of simultaneously applying the reset operation on the qubits of the 27-qubit quantum computer `ibmq_ehningen`, after preparing each qubit in the $|1\rangle$-state (`Reset X Simul.`, outliers omitted). The reset operation is repeated zero to four times. The reset fidelity on qubits 8, 12, 15, 19, and 22 is plotted on a second y-axis. A red line marks the respective mean value.

5.3.3 State-Dependent Errors

The impact of the initial state of a reset qubit on the reset fidelity is investigated in Figure 5.6. In this section, the `Reset Rand. Simul.` characterization experiment has been used to evaluate the reset fidelity on the qubits of the `ibmq_ehningen` quantum computer. The y-axis shows the reset fidelity of all qubits for a random state initialization that has an overlap with the $|0\rangle$ −state as indicated on the x-axis.

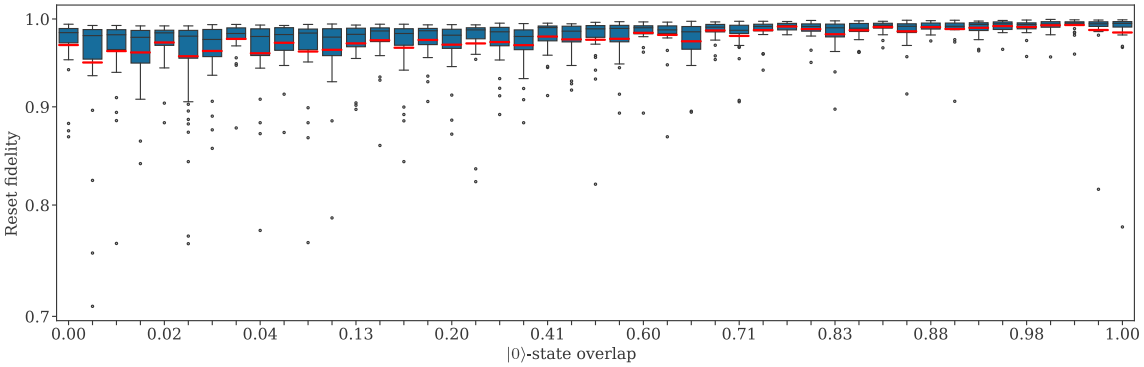


Figure 5.6: The fidelity of resetting a single-qubit state by applying one reset operation for various overlaps with the $|0\rangle$-state on the quantum computer `ibmq_ehningen` (`Reset Rand. Simul.`). A red line marks the respective mean value.

In Figure 5.6, the reset fidelity ranges from roughly 70% for an initial state with less than 2% overlap with the $|0\rangle$-state to almost 100% with initial states that have an overlap of

more than 98% with the $|0\rangle$-state. The reset fidelity increases gradually at a significantly decreased variance for an increasing overlap with the $|0\rangle$-state. Furthermore, the number and magnitude of reset fidelity outliers decrease with a larger overlap. When one reset operation is applied (as shown in Figure 5.6), the correlation between fidelity and overlap with the $|0\rangle$-state is largest at Pearson correlation coefficient of 0.33. That correlation coefficient halves to 0.15 when two reset operations are applied and is further reduced to 0.13 for the maximum number of considered reset repetitions (four repetitions). Thus, we conclude that the reset operation in near-term quantum computers is subject to state-dependent errors that can be significantly reduced by repeating the reset operation. As state-dependent reset errors can have a significant impact on reset fidelity as shown in Figure 5.6, a compilation based on qubit reuse must consider a repeated application of the reset operation to reduce the incurred error.

5.4 Rigorous Compilation with Qubit Reuse

In Figure 5.7, the steps of the rigorous compilation method based on qubit reuse are described. In general, the developed method is separated into an offline part (coloured red) and an at-runtime part (coloured blue) [JBP22]. The offline part is executed ahead of the quantum computation execution time, allowing for heavy-weight computations on a classical computer without delaying the quantum computation at the intended runtime phase. In this chapter, a satisfiability modulo theories (SMT) model $\mathcal{M}$ is generated by the developed rigorous compilation method. This SMT model is determined from the input quantum computation, a swap insertion model (e.g. [TC20]), an objective function defining a preferred quantum computation, and the qubit reuse compilation opportunity that employs reset operations and mid-circuit measurements. An SMT solver [MB08] is then used to compute a satisfying assignment to the determined SMT model $\mathcal{M}$. A satisfying assignment to model $\mathcal{M}$ completely defines a quantum computation that correctly applies qubit reuse and accommodates the limited qubit connectivity of a quantum computer. The objective function is used to select an element from the set of all satisfying assignments that yields a quantum computation with e.g. a lower qubit requirement or duration.

The at-runtime part is performed close to the execution of the intended quantum computation, within minutes or seconds before the execution. During this at-runtime part, the error characterization experiments are executed and the quantum computation previously determined in the offline part is adapted to the obtained error characteristics. The error characteristics are determined by regular calibration performed by the operator of the quantum computer and the `Reset X Simul.` reset error characterization experiments (see Section 5.3). The method introduced in [NT23] is employed in combination with a custom cost function to efficiently compute a subset of qubits on the quantum computer given the error characteristics of a target quantum computation. The details of the developed SMT model (see Section 5.4.1) and the at-runtime part (see Section 5.4.2) are described in the remainder of this section.

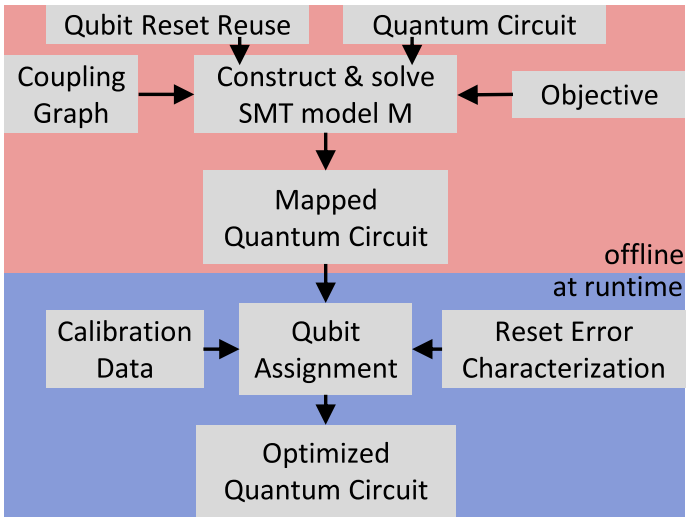


Figure 5.7: Individual steps of the developed qubit reuse compilation method.

5.4.1 Formal Modelling

The developed SMT model $\mathcal{M}$ for qubit reuse is determined from a given input quantum computation, qubit reuse compilation opportunity, an objective function and the swap insertion model $\mathcal{S}$ previously developed in [TC20]. The developed model $\mathcal{M}$ can be divided into a formal model $\mathcal{R}$ specifying the qubit reuse compilation opportunity and a formal model $\mathcal{S}$ specifying the accommodation of limited qubit connectivity by inserting swap gates.

The qubit reuse model $\mathcal{R}$ essentially extends typical swap insertion models [TC20, WBZ19] with qubits that can be 'unassigned' for a portion of the given quantum computation. A swap insertion model $\mathcal{S}$ must keep track of changes to the qubit assignment that have been induced by one or more swap gates. A qubit assignment defines which abstract qubit of a quantum computation is mapped to which physical qubit on a target quantum computer during which points of time of the quantum computation. To facilitate this, each abstract qubit starts with an initial assignment to a physical qubit of the quantum computer. In a swap insertion mode, this assignment may only change whenever swap gates are inserted into the quantum computation to accommodate a two-qubit gate that is not directly supported by the qubit connectivity of the target quantum computer. The qubit reuse model $\mathcal{R}$ extends this notion of qubit assignment by:

- Allowing an abstract qubit in a quantum computation to start without an initial assignment, i.e. to be unassigned.
- Disallowing swap gates on unassigned qubits.
- Specifying the change in qubit assignment through a reset operation: qubits that have been reset become unassigned while a different unassigned qubit becomes assigned.
- Disallowing reset operations on a qubit before the computation on that qubit was completed, i.e. the insertion of reset operations is only allowed after a measurement has occurred.

These conditions require the extension of the swap insertion model $\mathcal{S}$ by new variables and constraints that will be explained in the following text.

Model Variables

The qubit reuse model $\mathcal{R}$ takes a quantum computation with a set of qubits Q and a set of quantum gates G, a set of physical qubits on the quantum computer P, a set of measurements $M \subseteq G$ and the maximum considered depth T as an input. The following variables are part of model $\mathcal{R}$ for the given input:

- $A = \{a_{1,1}, ..., a_{|Q|,T}\}$—this set contains variables describing the assignment status of a qubit, i.e. $a_{q,t}$ evaluates to false only if the qubit q is not assigned to any physical qubit of the quantum computer at time step t else the variable $a_{q,t}$ evaluates to true.

- $L = \{l_1, ..., l_{|M|}\}$—this set contains variables describing potential reset locations i.e. the variable l_m evaluates to true if a reset operation is inserted after the measurement operation $m \in M$ on the same physical qubit else the variable evaluates to false. In the developed model, reset operations are only inserted after a measurement operator.

The maximum depth T can be given as an input or determined automatically by applying an efficient swap insertion algorithm [Qis23, LDX19, IBM23b] and determining the worst-case depth when qubit reuse was a maximal number of times in the input quantum computation. Since the swap insertion model $\mathcal{S}$ would always determine a solution with lower depth than the employed heuristic swap insertion algorithm, the determined depth can be used as an upper bound T.
A number of variables from the swap insertion model $\mathcal{S}$ occur in the same equations as variables in the qubit reuse model $\mathcal{R}$. These variables are described in the following:

- $\Pi = \{\pi_{1,1,1}, ...\pi_{|Q|,|P|,T}\}$—this is the set of variables that determine the qubit assignment at a specific time, i.e. the variable $\pi_{q,p,t}$ evaluates to true in a time step $t \in \{1, ..., T\}$ if an abstract qubit $q \in Q$ in the input quantum computation is mapped to the physical qubit $p \in P$ on the target quantum computer.

- $E = \{\sigma_{1,1,1}, ..., \sigma_{|P|,|P|,T}\}$—this is the set of variables that describe swap gate insertions, i.e. the variable $\sigma_{i,j,t}$ evaluates to true if a swap gate is inserted between physical qubits $i \in P$ and $j \in P$ at a time step $t \in \{1, ..., T\}$. As a consequence, the qubit assignment variables $\pi_{\_,i,t}$ and $\pi_{\_,j,t}$ corresponding to the participating qubits must also change accordingly.

- X_g—this is the set of variables corresponding to the location of a quantum gate g. For instance, let g be a single-qubit gate that needs to be executed on a physical qubit. The set $X_g = \{x_{g,1,1}, ..., x_{g,|P|,T}\}$ with variables $x_{g,p,t}$ that evaluate to true if the single-qubit gate gate $g \in G$ is executed at the time step $t \in \{1, ..., T\}$ on the

physical qubit p. Multi-qubit quantum gates are analogously associated with a set of variables X_g specifying the gate location on multiple physical qubits.

- $D_g = \{d_{g,1}, ..., d_{g,T}\}$—this is the set of variables that describe the time step at which a gate g is started to be computed, i.e. variable $d_{g,t}$ evaluates to true if gate g occurs at time step $t \in \{1, ..., T\}$.

Model Constraints

In the first time step of the quantum computation to be determined, assignment status variables $a_{\cdot,1}$ can be set arbitrarily, i.e. their assignment is not constrained. The application of a reset operation is required for a subsequent change in a qubit assignment status variable during a time step $t \in \{2, ..., T\}$. The following set of equations (or: constraints) require that a reset operation must be applied at the time step t and on a physical qubit $p \in P$ to which a qubit $q \in Q$ is assigned at time step $t + 1$, whenever a previously unassigned qubit q becomes assigned from one time step to another.

$$(\neg a_{q,t} \wedge a_{q,t+1}) \rightarrow \bigvee_{m \in M, p \in P} (l_m \wedge x_{m,p,t} \wedge d_{m,t} \wedge \pi_{q,p,t+1}) \,. \tag{5.1}$$

Reciprocally, if a qubit $q \in Q$ becomes unassigned at time step $t + 1$, the following constraints enforce a corresponding reset operation to occur at the previous time step (and vice versa):

$$(a_{q,t} \wedge \neg a_{q,t+1}) \leftrightarrow \left(l_{m_q} \wedge d_{m_q,t}\right) \tag{5.2}$$

where $m_q \in M$ is the measurement acting on the qubit $q \in Q$. The variables l_{m_q} and $d_{g,t}$ are used for consistency in the previous and subsequent equations.

No quantum gate may be computed at a time step t on a qubit $q \in Q$ if the qubit is not assigned at that time step $\neg q_{q,t}$. This yields a set of consequences that must be enforced to yield a correct quantum computation, i.e. a quantum computation where each quantum gate is exclusively assigned a physical location on the target quantum computer on a specific time step. Enforcing these consequences requires the following constraints to be true:

$$\neg a_{q,t} \rightarrow \neg d_{g,t}, \tag{5.3}$$

where $g \in G$ is a quantum gate that acts on the abstract qubit $q \in Q$ in a quantum computation.

In addition to the listed constraints, the swap insertion model $\mathcal{S}$ must be further relaxed to allow for changes to qubit assignment variables $\pi_{q,p,t}$ using reset operations instead of only using swap gates [TC20]. In previous swap insertion models, the assignment of abstract qubits in the quantum computation to physical qubits in the target quantum computer was always injective. This is not necessarily true anymore when qubit reuse is employed: as quantum gates may not occur on unassigned qubits, two abstract qubits $q, q' \in Q$ may be mapped to the same physical qubit $p \in P$ while still yielding a correct quantum computation as long as one of the qubits q, q' are unassigned. Consequently, the swap insertion model $\mathcal{S}$ is relaxed to permit these non-injective qubit assignments.

Evaluated Objective Functions

Three objective functions are evaluated with the developed model $\mathcal{M}$ to guide the selection of the satisfiable variable assignments by the SMT solver towards preferred quantum computations that are expected to perform better on a near-term quantum computer.

To minimize the duration of a quantum computation, the depth of the quantum computation is minimized by:

$$\min Z, Z \geq t \wedge d_{g,t}, \forall g \in G, \forall t \in \{1, ..., T\}, \tag{5.4}$$

where Z is an integer variable. The duration of a quantum computation and hence the incurred decoherence error is correlated with the depth of a quantum computation. Thus, if the corresponding depth is minimized the quantum computation is expected to incur fewer errors.

Another figure of merit that can be optimized with the qubit reuse compilation opportunity is the number of qubits required on the quantum computer to execute a given quantum computation. This figure can be minimized by the following objective function

$$\min \sum_{p \in P} \left(\vee_{q \in Q, t \in \{1,..,T\}} \pi_{q,p,t} \right), \tag{5.5}$$

where each physical qubit $p \in P$ of the quantum computer counts towards the qubit requirement if an abstract qubit $q \in Q$ was assigned to p at any time of the quantum computation, i.e. if $\pi_{q,p,t}$ evaluates to true at any time step.
The last figure of merit evaluated in this chapter is the number of inserted swap gates. The number of inserted swap gates can be optimized by

$$\min \sum_{e \in E} e, \tag{5.6}$$

where E is the set of swap insertions as defined previously.

Deriving the Optimized Quantum Circuit

The model $\mathcal{M}$ constraints the assignment to the model variables such that a satisfiable assignment yields a valid quantum computation that accommodates the qubit connectivity of a target quantum computer. As such the quantum computation uses each physical qubit on the quantum computer at most once for a quantum gate during a specific time step. Furthermore, any restrictions in qubit connectivity are resolved by either swap gates or qubit reuse. The timing and location of the inserted operations as well as of the quantum gates of the quantum computation can be directly derived by inspecting the model variable assignments, specifically the qubit assignment variables $\pi_{q,p,t}$, qubit assignment status variables $a_{q,t}$ and inserted reset operations variables l_m.

5.4.2 Qubit Assignment using Reset Error Characterization

After the developed rigorous compilation used qubit reuse to yield a quantum computation that can be performed on a target quantum computer directly, the reset error characterization and calibration data are obtained and used to select a subset of physical qubits on the quantum computer for computation. This subset of qubits is selected such that the quantum computation does not require further swap gate insertions or qubit reuse to accommodate the limited qubit connectivity. There are typically multiple such subsets of qubits as current and near-term quantum computers have a regular qubit topology and connectivity as shown in Figure 5.8 [AAB$^+$19, IBM23b, CZY$^+$20]. Thus, a compiled quantum computation can be computed on different sets of physical qubits in the quan-

tum computer. Methods such as [NT23], exploit this regular structure to efficiently find a suitable subset of qubits that minimizes a given cost function. This section extends these methods by selecting the number of reset operation repetitions per qubit while considering the error characterization yielded by both, the calibration data but also the performed reset error characterization experiments.

The optimized selection of a qubit subset is determined by the steps depicted in Algorithm 5.1. In the first step, the calibration data ξ_{x_g} collected by the operator of the target quantum computer is obtained. The operator of the quantum computer typically includes the fidelity of each gate g supported by the quantum computer at the location x in the calibration data ξ_{x_g}. As introduced in the previous section, the location x can either be a qubit in the quantum computer or the connection between two qubits. Then, the reset error characterization experiments are executed on the target quantum computer, yielding reset error $\xi'_{p,r}$ per qubit p and reset repetitions r. The `Reset X Simul.` reset error characterization experiment introduced in Section 5.3 is used for the evaluation of the developed rigorous compilation method. The reset error $\xi'_{p,r}$ obtained by the `Reset X Simul.` characterization experiment is considered in conjunction with the incurred idle time to determine the number of reset repetitions per qubit. Since the application of reset operations may lead to idle time and thus to additional decoherence on non-reset qubits we consider the duration $\epsilon_{p,r}$ of performing r reset operations on the physical qubit p to compute the value

$$\xi_{p,r} := e^{-\epsilon_{p,r}/T} \cdot (1 - \xi'_{p,r}) \tag{5.7}$$

that indicates the fidelity of applying the reset operation r times on physical qubit p on a quantum computer while subject to the decoherence time T. The number of reset applications R_p optimizing the reset fidelity for a specific physical qubit p is subsequently determined by computing $\arg\max_r \xi_{p,r}$ for a fixed qubit p.

In the next step, the method in [NT23] is invoked with the following cost function to perform the lines 5-7 in Algorithm 5.1:

$$\text{cost}(\bar{a}, \xi, R_p) := 1 - \prod_{x_g \in C} \xi_{x_g} \prod_{r_p \in C} \xi_{p,R_p}, \tag{5.8}$$

where C is the compiled quantum computation and x_g is a quantum gate g in C applied to the location x that determined by the qubit subset selection $\bar{a}$. Furthermore, the variable $p_r \in C$ of Equation (5.8) denotes a reset operation in the compiled quantum computation that is applied to physical qubit p according to the qubit subset selection $\bar{a}$. As such, the reset operation on physical qubit p is applied R_p times. Finally, the determined qubit subset selection $\bar{a}_f \in \bar{A}$ that yields the minimal cost $K_{\bar{a}}$ is computed. This qubit subset selection $\bar{a}_f$ is then applied to the compiled quantum computation C to yield a new quantum computation O where reset operations on qubit p are applied R_p times and the abstract qubits are assigned to physical qubits according to $\bar{a}_f$.

Algorithm 5.1: Qubit subset selection with reset operations

Input: A compiled quantum circuit C, the qubit connectivity H
Output: A compiled quantum circuit with optimized qubit selection

begin
  $\xi_{x_g} \leftarrow$ fidelity of quantum gate g at location x;
  $\xi_{p,r} \leftarrow$ fidelity of r resets on physical qubit p;
  $R_p \leftarrow$ derive the number of reset repetitions per qubit from $\xi_{p,r}$;
  $\bar{A} \leftarrow$ `subgraph_isomorphism`(graph(C), H);
  for $\bar{a} \in \bar{A}$ **do**
    $K_{\bar{a}} \leftarrow$ cost$(\bar{a}, \xi, R_p)$;
  end
  $\bar{a}_f \leftarrow \bar{a}$ with lowest cost $K_{\bar{a}}$;
  $O \leftarrow$ assign qubits in C as defined in $\bar{a}_f$ with reset applications as in R_p;
  return O
end

5.4.3 Example: 7-Qubit Bernstein-Vazirani

The developed compilation method is applied on a 7-Qubit Bernstein-Vazirani quantum computation, a heavy-hex qubit connectivity of the 27-qubit `ibmq_ehningen` quantum computer (see Figure 5.2) in this section. Assuming an objective function for minimizing the number of swap gates (see Equation (5.6)), the rigorous compilation may yield the quantum computation shown in Figure 5.8. This compiled quantum computation is then subject to the steps outlined in Algorithm 5.1, where first the gate fidelities are obtained [MAB+18] and the (Reset X Simul.) reset error characterization experiment (see Section 5.3) is performed yielding the data in Table 5.1.

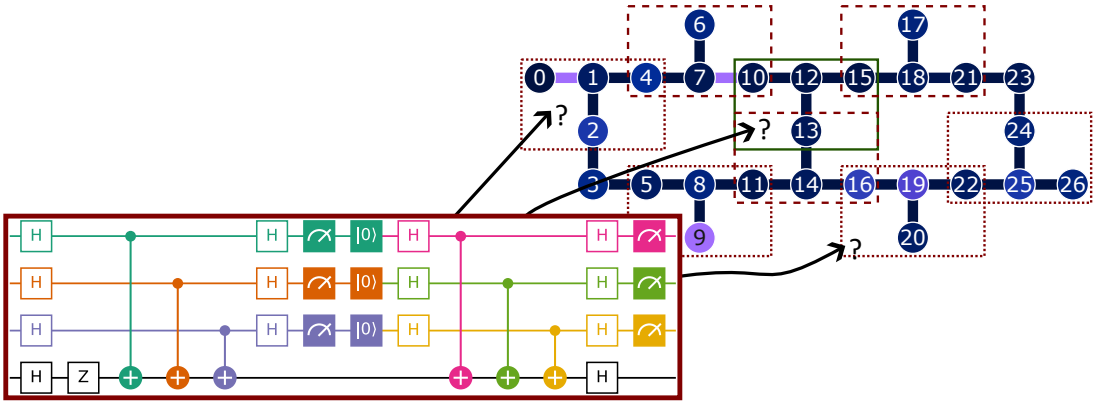


Figure 5.8: Mapping a 7-qubit Bernstein-Vazirani quantum circuit with three reused qubits (lower left, also see Figure 5.2b) to the coupling graph of `ibmq_ehningen` (upper right).

Applying Equation (5.7) on the reset operation duration, decoherence times of `ibmq_ehningen` and the reset error data as shown in rows one, three, and five of Table 5.1 yields the values given in rows two, four, and six of Table 5.1. A suitable number of reset applications can be derived from these values (last row in Table 5.1) that are used to yield an optimized quantum computation and to select a qubit subset for computation. The compiled quantum computation depicted in Figure 5.2 would require four physical qubits with three qubits connected to one common qubit. The heavy-hex qubit connectivity of `ibmq_ehningen` exhibits eight qubit subsets on which the compiled quantum computation can be performed without requiring swap gates (see Figure 5.8). By computing the custom cost function defined in Equation (5.8) on each of these qubit subsets, the qubit subset with the smallest cost, i.e. with the highest expectation of successful computation, can be determined. Let the qubit subset $(10, 13, 15, 12)$ yield the lowest cost as defined by Equation (5.8). The abstract qubits $(0, 1, 2, 3)$ in the quantum computation are assigned to the physical qubits $(10, 13, 15, 12)$ respectively. Furthermore, as given by the reset repetition number defined in the last row of Table 5.1, each reset operation occurring is applied R_p times before the quantum computation is executed on the target quantum computer.

5.5 Evaluation

The developed qubit reuse compilation method is evaluated on a set of quantum computations consisting of the Bernstein-Vazirani (BV) [BV97] quantum circuits, Hadamard ladder

Table 5.1: Reset error data considered for qubit subset selection.

qubit	p_0	p_1	p_2		p_{20}		p_{24}	p_{25}	p_{26}
$\xi'_{p,1}$	0.976	0.986	0.984		0.971		0.984	0.973	0.961
$\xi_{p,1}$	0.971	0.981	0.978		0.965		0.979	0.967	0.955
$\xi'_{p,2}$	0.996	0.993	0.998	...	0.980	...	0.999	0.978	0.994
$\xi_{p,2}$	0.985	0.983	0.988		0.969		0.988	0.967	0.983
$\xi'_{p,3}$	0.99	0.993	0.995		0.987		0.999	0.98	0.997
$\xi_{p,3}$	0.975	0.977	0.979		0.971		0.983	0.964	0.981
R_p	2	2	2		3		2	1	2

(H-ladder) [KW23] quantum circuits, and a quantum circuit computing the exclusive-or (XOR) function (xor5_254) using up to ten qubits [WGT+08]. The Hadamard ladder quantum circuit (as depicted on the right side of Figure 5.9) does, in contrast to the Bernstein-Vazirani quantum circuit (as depicted in Figure 5.1) and the exclusive-or quantum circuit (as depicted on the left side of Figure 5.9), only require linear qubit connectivity. As a consequence, the Hadamard ladder quantum circuits can be executed directly on near-term quantum computers such as `ibmq_ehningen` without requiring the insertion of swap gates or other auxiliary instructions. The BV and xor5_254 quantum circuits, however, require swap gate insertions depending on the size of the quantum circuit.

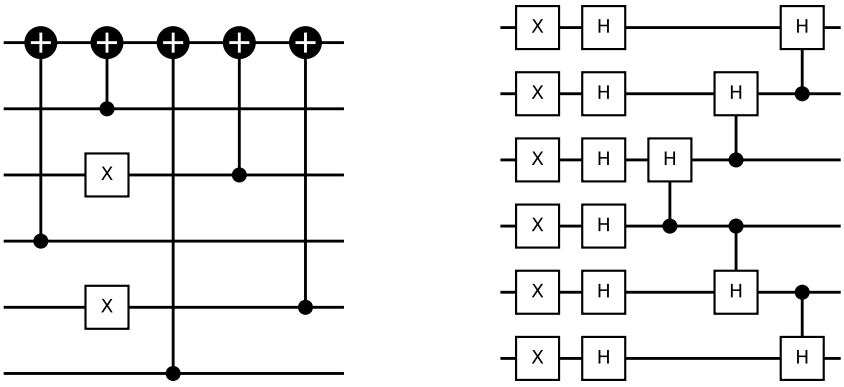


Figure 5.9: The xor5_254 quantum circuit computing an exclusive-or (XOR) function (left) and a 6-qubit Hadamard ladder quantum circuit (right).

The rigorous compilation method utilizing qubit reuse was evaluated with two different objective functions on the evaluated quantum circuits. The number of required swap gate insertions should be minimized by one of the evaluated objective functions. The second

evaluated objective function should minimize the number of required qubits or set this qubit requirement to a specific number.

After a quantum computation has been compiled by the rigorous compilation method and subsequently a suitable qubit subset was selected, the characteristics of the determined quantum computations are reported. The quantum circuits generated by Qiskit are used as a basis for comparison [Qis23]. The evaluated characteristics of the compiled quantum computations are the quantum circuit depth, the number of required swap gate insertions, the number of required qubits on the quantum computer, the Hellinger fidelity [Hel09], and the estimated success probability (ESP). The estimated success probability (ESP) is defined as the product of gate fidelities, measurement fidelities, and reset fidelities in a quantum computation [TQ19] and can be determined by inspecting a given quantum computation and corresponding fidelity data. The fidelity data required for computing the ESP is retrieved from the calibration data provided by the quantum computer operator during calibration and from the performed reset error characterization experiments (see Section 5.3) that are performed shortly before executing the evaluated quantum circuits. The Hellinger fidelity is computed by determining the overlap in measurement outcomes from an error-free simulation and an execution of the compiled quantum computation on a near-term quantum computer. In this evaluation, the 27-qubit IBM quantum computer `ibm_hanoi` was used for determining the ESP and Hellinger fidelity.

5.5.1 Impact of Compilation with Qubit Reuse on Characteristics

The impact of the compilation performed by Qiskit and the rigorous method developed in this chapter on the characteristics of the evaluated quantum computations is shown in Table 5.2. The data in the first five columns of the table is obtained by applying Qiskit to the evaluated quantum computation and is reported as absolute values. The remaining columns depict data obtained by applying the rigorous compilation method developed in this chapter with various objective functions to the evaluated quantum computations. Except for qubit columns, these columns contain values relative to the data reported by Qiskit. By applying the developed rigorous compilation method with the objective to minimize the number of swaps on a Bernstein-Vazirani quantum circuit with 7 qubits (BV7

- second row, columns six to nine), for instance, yields a 4-qubit quantum circuit with a 13% reduced duration, a 20% improved ESP, and swap gate insertions.

When minimizing the number of swap gate insertions with the developed compilation method (columns five to eight of Table 5.2), the swap gate insertions could be completely replaced by qubit reuse at no overhead in quantum circuit depth compared to the quantum computations compiled by Qiskit. In general, however, qubit reuse may have a marginal impact on swap gate insertions or may lead to an increase in swap gate insertions.

The 10-qubit Bernstein-Vazirani quantum circuit exhibited the largest improvement in ESP if the qubit requirement was minimized (last column, third row in Table 5.2), and the qubit reuse is maximal. The maximal qubit reuse had the same impact on swap gate insertions as the swap gate minimization objective (columns five to eight of Table 5.2). In contrast to the swap gate minimization objective, the quantum circuit depth is increased significantly, which may lead to larger errors due to decoherence. In the worst observed case of the evaluated quantum circuits, quantum circuit depth increased by 141% for the 10-qubit Bernstein-Vazirani quantum circuit.

Another interesting data point is the reuse of one qubit in the 4-qubit Bernstein-Vazirani quantum circuit (BV4) where the number of swap gates insertions was zero to begin with (row one, columns nine to twelve). Instead of a decrease in ESP due to the insertion of one reset operation when reusing one qubit, the ESP stays constant as a qubit subset with higher fidelities can be selected for quantum computation (row 1, column twelve). The adverse effect on the ESP when inserting reset operations without replacing swap gate

Table 5.2: The number of qubits (Q columns), quantum circuit depth (D columns), number of swap gate insertions (S columns), and ESP for the investigated quantum circuits.

	Qiskit				mininimal swaps				one reuse				maximal reuse			
circuit	Q	D	S	ESP	Q	D	S	ESP	Q	D	S	ESP	Q	D	S	ESP
BV4	4	10	0	0.93	4	1.10x	0x	1.0x	3	1.70x	0.00x	1.00x	2	2.60x	0x	0.99x
BV7	7	23	4	0.70	4	0.87x	0x	1.2x	6	0.83x	0.50x	1.17x	2	2.30x	0x	1.20x
BV10	10	33	9	0.54	4	0.85x	0x	1.4x	9	0.82x	0.78x	1.05x	2	2.42x	0x	1.41x
H-ladder3	3	8	0	0.96	3	1.00x	0x	1.0x	2	1.88x	0.00x	0.99x	2	1.89x	0x	0.99x
H-ladder5	5	11	0	0.93	5	1.00x	0x	1.0x	4	1.27x	0.00x	0.99x	2	2.18x	0x	0.95x
H-ladder7	7	14	0	0.86	7	1.00x	0x	1.0x	6	1.14x	0.00x	1.02x	2	2.43x	0x	0.95x
xor5_254	6	10	2	0.84	4	1.00x	0x	1.0x	5	0.80x	0.50x	1.03x	2	1.60x	0x	1.03x

insertions can only be partially mitigated by a better qubit subset selection in general. Reducing the qubit requirement of the 7-qubit or 5-qubit Hadamard ladder quantum circuit to two qubits, for instance, leads to an ESP that is reduced by 5% while the quantum circuit depth is more than doubled compared to the Qiskit compilation. While qubit reuse is in principle able to reduce the number of swap gate insertions significantly and higher-quality qubit subsets can be selected for some quantum circuits, qubit reuse can also have an adverse impact on other quantum circuit characteristics.

5.5.2 Impact of Increasing Qubit Reuse on Circuit Characteristics

The impact of increased qubit reuse on the quantum circuit characteristics and Hellinger fidelity of the 10-qubit Bernstein-Vazirani quantum circuit (BV10), 7-qubit Hadamard ladder (H-Ladder7), and the quantum circuit implementing the exclusive-or function (xor5_254) are reported in Table 5.3. Unsurprisingly, there is a pronounced increase in quantum circuit depth with an increased qubit reuse, for all evaluated quantum circuits. When the qubit requirement is decreased from three qubits to two qubits, the increase in quantum circuit depth is most pronounced at 82%, 26%, and 33% respectively for the BV10, H-Ladder7, and xor5_254 quantum circuits. From a different point of view, however, the developed method can reuse 7, 2 or 3 qubits compared to the circuit compilation performed by Qiskit, if the quantum circuit depth is allowed to be increased by 50%.
Furthermore, the qubit reuse must not be maximal for minimizing the number of swap gate insertions. No swap gate insertions on the heavy-hex qubit connectivity were necessary when the qubit requirement had been lowered to four for both the XOR and the Bernstein-Vazirani quantum circuit. For these quantum circuits, the maximum qubit reuse yields a qubit requirement of two for both quantum circuits. As seen for the case of the 7-qubit Hadamard ladder quantum circuit, the number of swap gate insertions can even increase with an increase in qubit reuse when the objective is to minimize quantum circuit depth after reaching the desired qubit reuse.
The developed rigorous compilation method utilizing qubit reuse was able to improve the Hellinger fidelity significantly. For instance, the Hellinger fidelity of the 10-qubit Bernstein-Vazirani quantum circuit was improved by 4.3x and by 1.16x for the xor5_254 quantum circuit. When using qubit reuse, the Hellinger fidelity can not always be im-

proved as shown in the data of the 7-qubit Hadamard ladder quantum circuit. An increase in Hellinger fidelity generally correlates more with a reduction in swap gate insertions due to qubit reuse and less with a change in ESP. For the xor5_254 quantum circuit, the developed method incurs a change in Hellinger fidelity that roughly correlates with the change in ESP for increased qubit reuse. However, such a pattern was not observed for the Hadamard ladder and Bernstein-Vazirani quantum circuits. The ESP increased by 40% compared to the Qiskit solution while the Hellinger fidelity improved significantly more by 330% for the Bernstein-Vazirani quantum circuit. Furthermore, the ESP decreases by 5% while the Hellinger fidelity decreases by 93% for the Hadamard ladder quantum circuit.

This may be caused by two potential effects. First, we report a large rate of outliers and occasionally a large variance for some qubits regarding the reset fidelity in Section 5.3.1. Thus, the state transformation specified by the reset operation may occasionally fail for single quantum circuits even if the average reset fidelity is sufficiently high. Second, the ESP metric assumes independent errors and does thus not capture e.g. additional errors on qubits neighbouring the reset qubit. Given that crosstalk errors have been reported to be incurred by reset operations [MDS22], a more detailed error metric may be required to yield a strong agreement with the results determined by a near-term quantum computer.

Table 5.3: Hellinger fidelity and circuit characteristics for no qubit reuse (Qiskit) to eight reused qubits (this chapter).

		reused qubits								
circuit	characteristic	0	1	2	3	4	5	6	7	8
BV10	depth	33	27	24	30	28	30	28	44	80
	swap	9	7	5	3	2	1	0	0	0
	ESP	54%	57%	49%	51%	73%	75%	76%	75%	76%
	fidelity	9%	18%	8%	11%	5%	6%	2%	12%	39%
H-Ladder7	depth	14	16	21	27	34	-	-	-	-
	swap	0	0	1	0	0	-	-	-	-
	ESP	86%	87%	84%	83%	82%	-	-	-	-
	fidelity	79%	74%	9%	2%	6%	-	-	-	-
xor5_254	depth	10	8	10	12	16	-	-	-	-
	swap	2	1	0	0	0	-	-	-	-
	ESP	84%	86%	87%	88%	86%	-	-	-	-
	fidelity	79%	88%	92%	90%	87%	-	-	-	-

5.6 Summary

The combined model allows to optimally trade-off the application of swap gates and qubit reuse for different costs of reset operations and swap gates. In this chapter, a set of reset error characterization experiments was introduced to quantify the reset fidelity on a 27-qubit quantum computer. Furthermore, a compilation method was developed that utilizes qubit reuse to determine a quantum computation with minimal quantum circuit depth, minimal number of swap gate insertions, or minimal qubit requirement. The developed rigorous compilation method is augmented by an efficient low-latency qubit subset selection that considers the varying fidelity of repeated reset operations as well as quantum gate and measurement fidelities. The qubit reuse compilation method is then evaluated on quantum computations with up to 10 qubits to determine the characteristics and Hellinger fidelity of thereby compiled quantum computations on a near-term quantum computer with heavy-hex connectivity.

The reset fidelity was revealed to vary highly from 67.5% to 100% by the introduced reset error characterization experiments. In addition, the reset operation was observed to incur errors that depend on the initial state of the reset qubit. A low-effort lower bound on reset fidelity for compilation purposes can be determined by simultaneously initializing the qubits to the $|1\rangle$-state and subsequently resetting as well as measuring all qubits. The developed rigorous compilation method was able to use qubit reuse to replace a significant portion of swap gate insertions and to increase the Hellinger fidelity by up to 4.3x. Finally, completely exhausting qubit reuse during compilation may not yield the best characteristics or Hellinger fidelity. The utilization of qubit reuse must be guided by the decrease in swap gate insertions and the increase in quantum circuit depth bundled with the subsequent ability to select a higher-quality qubit subset to yield quantum computations with an increased result quality.

Chapter 6

Topology Changes Induced by Rydberg Shifts

Near-term quantum computers are based on nascent technology platforms such as ion traps [ION23, LWF+17, QuT23, PDF+21], superconducting circuits [AAB+19, IBM23d, Rig23], neutral atoms [JCZ+00, Saf16, BL20] and NV centers [DMD+13, N+14]. For these technologies, the qubit connectivity is limited, i.e. only a subset of qubit-qubit interactions as defined by the topology of a quantum computer is supported at a time. If a quantum computation exhibits unrestricted qubit connectivity in its computational steps, the algorithm can be made compliant with the topology of a quantum computer by quantum circuit mapping. Typically, quantum circuit mapping requires the insertion of auxiliary instructions, such as swap gates, into the input quantum computation [CDD+19, DSB+22]. Then insertion of these auxiliary instructions can cause additional errors when the quantum computation is eventually executed on a target quantum computer. Reducing the overhead incurred by quantum circuit mapping is therefore crucial in order to extend the computational reach of current and near-term quantum computing technology [Pre18].

Quantum computing technologies based on Rydberg atoms are characterized by long decoherence times, high-fidelity quantum gates [MCS+20], multi-qubit interactions with more than two qubits [LKS+19] and the ability to change their topology during computation that allows for dynamic qubit connectivities [NLR+14, BDLL+16, EBK+16]. This nascent technology is therefore explored for near-term quantum computing with prototypes currently investigated by research groups [QRy23] and startups such as QUERA [QUE23] and Pasqal [Pas23]. However, the changes in topology supported by the Rydberg atoms platform have only started to be explored for quantum circuit mapping.

Topology changes do not manipulate the quantum state per se but induce a change in qubit connectivity. They are therefore a promising candidate for quantum circuit mapping as topology changes can have a lower error rate than swap gates that are typically used for quantum circuit mapping. Furthermore, one topology change may replace the need for multiple swap gates, thereby reducing the duration and thus errors due to decoherence of a compiled quantum computation. The computational reach of quantum computers may therefore be extended by supporting topology changes.
In this chapter, a rigorous compilation method is developed that exploits compilation opportunities available with topology changes for accommodating the limited qubit connectivity of a near-term quantum computer more efficiently:

- A formal model is developed that incorporates topology changes in one dimension and swap gate insertions in conjunction.
- The developed rigorous compilation method is evaluated on intermediate-scale quantum circuits.
- Technology parameters for near-term Rydberg quantum computers are characterized that enable a more efficient accommodation of limited qubit connectivity.

The remainder of this chapter is structured as follows. Quantum computers based on Rydberg atoms and topology changes supported by these devices are introduced in section 6.1. The formal model describing topology changes supported by the Rydberg atoms platform is introduced and integrated with a swap gate insertion model in section 6.2. The evaluation in section 6.3 demonstrates the potential for more efficient quantum circuit mapping in terms of quantum circuit depth and fidelity depending on different technology parameters. In section 6.4 this chapter is summarized. Parts of this chapter have been published as [BBP21].

6.1 Quantum Computing Based on Rydberg Atoms

In technology platforms based on Rydberg atoms, an array of individually trapped neutral atoms is used for realizing a quantum computer. Optical tweezers trap the neutral atoms in place and also allow for a deterministic loading of the atom array by physically moving

individual atoms [NLR+14, BDLL+16, EBK+16]. The single-qubit gates in such a technology platform are realized by individual optical addressing of atoms. For the realization of two-qubit gates, the atoms representing the interacting qubits of the two-qubit gates are excited into a Rydberg state, where the Van der Waals interaction is enhanced such that it induces a strong interaction [JCZ+00, GMW+09, UJH+09, LKS+19, GKG+19, MCS+20].

A major challenge of topology changes in the neutral atom technology platform is to avoid errors due to qubit dephasing and the coupling of the qubit to motional degrees of freedom. Such errors can be avoided when the movement of the atoms is restricted along a row of the two-dimensional atom array with a fixed ordering of the qubits. This setup is currently experimentally implemented in e.g. [QRy23]. Thus, while the arbitrary movement of atoms during a quantum computation is possible in principle, the movement of atoms is expected to be restricted during a quantum computation on near-term quantum computers based on Rydberg atoms. Therefore, the focus in this chapter is on one-dimensional displacements (one-dimensional shifts) where atoms are only moved along one row. These one-dimensional displacements are modelled by changing the topology graph of a target quantum computer.

Individual qubits and the qubit connectivity in a quantum computer are represented by a topology graph. The topology graph $G = (P, E)$ is defined as follows. For each physical qubit in the quantum computer, one vertex $v \in P$ is included in the topology graph. If a physical qubit p and a physical qubit u in the quantum computer can interact with each other, i.e. participate in a common multi-qubit gate, one edge $e = (p, u) \in E$ is included in the topology graph. Removing or adding at least one edge to the topology graph constitutes a topology graph change.

An example of one-dimensional topology displacements is described for an atom array with two rows and three columns as shown in Figure 6.1 where three topology changes denoted as *a*, *b* and *c* are depicted. The qubits in the first row of the depicted array are displaced to the right by one qubit in topology change *a*. In topology change *b*, qubit 0 and qubit 1 are displaced to the left by one qubit while qubit 2 keeps its position. Qubit 2 could also be displaced to the right by the same topology change *b*. The last topology change *c* is not permitted by the envisaged one-dimensional topology displacements since the relative positions of two qubits in the first row are swapped.

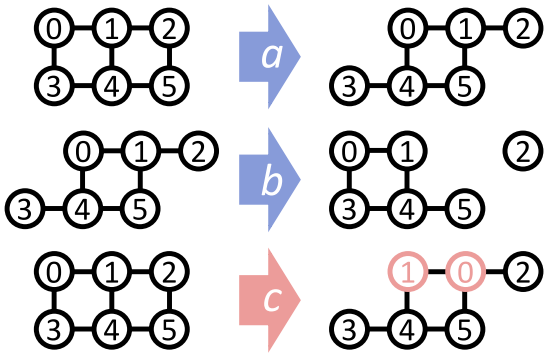


Figure 6.1: Example topology graph changes induced by one-dimensional displacements. The topology graph changes a and b are supported by one-dimensional displacements while the topology graph change c is not supported because it changes the relative order of qubits.

A subset of all possible topology graph changes can be supported by a quantum computer depending on the employed technology platform. Quantum computers based on technologies such as superconducting qubits and NV centers do not support deliberate changes in topology during a quantum computation and are therefore typically represented by a static topology graph for compilation purposes [N$^+$14, AAB$^+$19]. However, the topology can change by physically moving qubits [WKM21, Bar15] or changing other operational parameters [BKM16] in quantum computations based on photons, ion traps or neutral atoms. The topology can change arbitrarily in photonic quantum computing by placing mirrors [Bar15] or waveguides [Hun95]. Electromagnetic fields can be applied in ion trap quantum computers, to support arbitrary topology changes [BKM16]. Other realizations of quantum computations based on ion traps allow to change the topology in a limited way by physically moving qubits represented by ions [WKM21].

6.2 Rigorous Compilation with Topology Changes

In this section, a rigorous compilation method is described that incorporates topology changes induced by one-dimensional shifts into a satisfiability modulo theories (SMT) model $\mathcal{M}$. The formal model $\mathcal{M}$ is the union of a model $\mathcal{B}$ that describes topology changes supported by one-dimensional shifts and a swap gate insertion model $\mathcal{S}$.

The individual steps of the developed compilation method given a quantum computation U, maximal time step T, and a topology graph $G = (P, E)$ are depicted in Figure 6.2. In the first step, the topology graph G is extended to yield a graph $G' = (P, E')$ that includes the qubit connectivity supported via all valid topology graph changes. Then, the input quantum computation U, the extended topology graph G' and the maximum quantum circuit depth T are used to construct the model $\mathcal{M}$. An SMT solver [MB08] subsequently computes a satisfying assignment to the variables in model $\mathcal{M}$ subject to further objective functions such as the minimization of the quantum circuit depth or the maximization of the expected circuit fidelity.

The variable assignment returned by the solver is optimal with respect to the chosen objective function, yielding an optimal quantum computation. If the solver can not return a satisfiable assignment within the given maximum quantum circuit depth T, model $\mathcal{M}$ is modified by increasing T. The maximum quantum circuit depth T is increased iteratively until a valid assignment to $\mathcal{M}$ is found, i.e. the compiled quantum computation accommodates the limited qubit connectivity of the target quantum computer. In general, optimality is guaranteed with this approach if the value of the employed objective function can not improve by increasing the quantum circuit depth. Alternatively, an initial maximum quantum circuit depth T must be chosen that allows the solver to explore all qubit permutations and topology changes in the quantum circuit U.

In the following text, P is the set of physical qubits in the target quantum computer, Q are the abstract qubits in the given quantum computation U and T is the last time step considered. The topology assumed in this section is an array defined by R rows and C columns. The same identifiers R and C also specify a function that a physical qubit $p \in P$ to its corresponding row or column in the topology graph. Constraints defining the domain of a variable $v \in \mathcal{B} \cup \mathcal{S}$ have been omitted in the following text.

6.2.1 Topology Graph Extension

The developed compilation method first constructs the extended topology graph $G' = (V, E')$ according to the supported topology graph changes (see section 6.1). In this chapter, we assume a Rydberg atom quantum computer with an initial topology G with a two-dimensional qubit connectivity and qubits arranged in a $C \times R$ array. Arbitrary one-

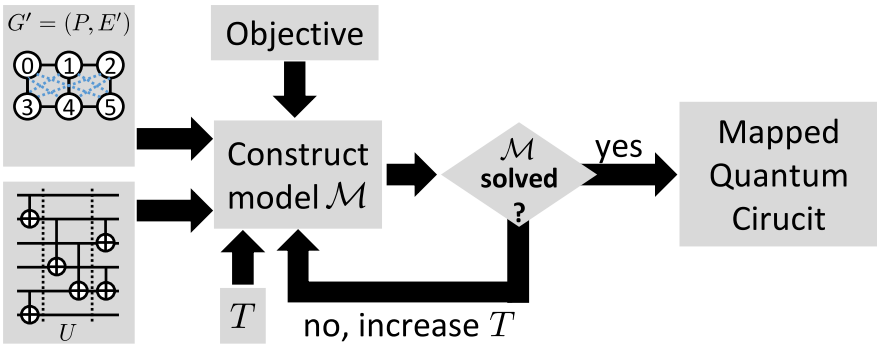


Figure 6.2: Individual steps of the developed rigorous compilation method given a quantum computation U, topology graph $G' = (P, E')$ and maximal considered time step T.

dimensional displacements in one row along the x-axis of the array are modelled while maintaining the relative order of qubits. The resulting extended topology graph G' is shown in figure 6.3. An arbitrary qubit q of row r can be connected with an arbitrary qubit u of a neighbouring row r' since the qubits can be displaced arbitrarily along the x-axis of the array. Since the relative position of qubits in the same row may not change, qubits q, q' in the same row can only interact directly if the qubits are connected in the original topology graph G. Thus, the edge set E' of the extended topology graph G' is defined by

$$E' = E \cup \{\{q, u\} \mid q \in r, u \in r'\}. \tag{6.1}$$

In general, $\mathcal{O}(C^2R)$ edges are included in G' for an initial topology given by an array with C columns and R rows.

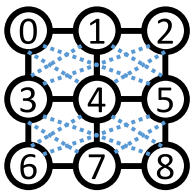


Figure 6.3: The topology graph representing the qubit connectivity enabled by one-dimensional topology displacements. Black edges denote the qubit connectivity in the initial topology and dashed, blue edges denote the additional qubit connectivity supported by suitable topology changes.

6.2.2 Modelling One-Dimensional Topology Displacements

In a subsequent step of the developed rigorous compilation method, the dynamics of the topology graph is defined, i.e. the way a topology graph can change from one time step to another. In addition, the edges of the extended topology graph G' that are available for computation in a specific time step must be defined. Given an extended topology graph $G' = (P, E')$ and the maximum considered quantum circuit depth T, one-dimensional displacements (see section 6.1) are defined in $\mathcal{B}$ to contain the following variables:

- $\eta = \{\eta_{1,1}, ..., \eta_{|P|,T}\}$—this set represents the displacement of a physical qubit $p \in P$ at a time step $t \in \{1, ..., T\}$. The variable $\eta_{p,t}$ evaluates to zero if the physical qubit p was not displaced with respect to its position in the original topology graph G. A positive value of $\eta_{p,t}$ indicates a displacement of physical qubit p to the right and a negative value indicates a displacement to the left at time step t.

- $EN = \{EN_{1,1,1}, ..., EN_{|P|,|P|,T}\}$—this set represents whether a specific qubit connectivity is valid in time step t. Specifically, $EN_{p,u,t}$ evaluates to true if the qubit connection between physical qubit p and physical qubit u is available for the computation of a multi-qubit gate at time step t.

Since the assumed one-dimensional topology displacements fix the order of qubits in the same row for all time steps $t \in T$, the following constraints are required for each qubit $p \in P$ and displacement $\eta_{p,t}$:

$$(\eta_{p,t} \leq \eta_{r,t}) \wedge (\eta_{l,t} \leq \eta_{p,t}), \tag{6.2}$$

where r (l) is in the set of physical qubits that are assigned a position right (left) to the physical qubit p, i.e. $r \in \{u \in P \mid (C(u) > C(p)) \wedge (R(u) = R(p))\}$ and $l \in \{u \in P \mid (C(u) < C(p)) \wedge (R(u) = R(p))\}$.
Next, the qubit connectivity at a time step t must be defined given by the individual qubit displacements at that time step:

$$EN_{p,u,t} := (\eta_{p,t} = \eta_{u,t}), \tag{6.3}$$

for physical qubits p, u that are in the same row – they are connected when they have been displaced by the same amount in a time step t. For physical qubits p, u that are in neighboring rows:

$$EN_{p,u,t} := ((C(p) + \eta_{p,t}) = (C(u) + \eta_{u,t})), \tag{6.4}$$

i.e. two physical qubits are connected if their displacements cause them to be in the same column of the array in a time step t.

Depending on the specific realization of the technology platform, a topology displacement may require a runtime of $t_d \in \{1, ..., T\}$ time steps. Consequently, no other operation may be computed on a physical qubit $p \in P$ in $t' \in \{t - t_d, ..., t\}$ if a change in displacement $\eta_{p,t} \neq \eta_{p,t+1}$ occurs on a qubit p between time step t and $t + 1$. The runtime t_d is a user-specified input to model $\mathcal{M}$ and can be e.g. defined by the duration of the displacement as measured experimentally on a quantum computer based on Rydberg atoms.

6.2.3 Formal Swap Gate Insertion Model

A swap gate insertion model $\mathcal{S}$ defines an assignment of abstract qubits Q in a quantum computation U to the physical qubits of a quantum computer P for each considered time step. This qubit assignment can be changed by inserting swap gates into the given quantum computation. Furthermore, each quantum gate $g \in U$ is assigned a location $l \in L = \mathcal{P}(P) \setminus \{\}$ for each time step $t \in \{1, ..., T\}$. The swap insertion model $\mathcal{S}$ described in [TC20] is used in this chapter.

The following variables are included in a swap insertion model $\mathcal{S}$:

- $\Pi = \{\pi_{1,1,1}, ..., \pi_{|Q|,|P|,T}\}$—this set of variables represents the qubit assignment in the input quantum computation U for each time step. Specifically, $\pi_{q,p,t}$ evaluates to true if the abstract qubit $q \in Q$ is assigned to the physical qubit $p \in P$ at time step t.
- X_g—this is the set of variables corresponding to the location of a quantum gate g. For instance, the location of a single-qubit gate g is represented by $X_g = \{x_{g,1,1}, ..., x_{g,|P|,T}\}$. The variable $x_{g,p,t}$ evaluates to true if the single-qubit gate g is executed at the time step t on physical qubit p. The location of multiple-qubit quantum gates is analogously specified by a set of variables where a location is a set of physical qubits instead.

- $D_g = \{d_{g,1}, ..., d_{g,T}\}$—this is the set of variables that describe the time step at which a gate g is started to be computed, i.e. variable $d_{g,t}$ evaluates to true if gate g occurs at time step t.

and constrained by the following conditions:

- Each quantum gate $g \in U$ is assigned a valid location in X_g and a time step d_g such that the dependency defined in the input quantum computation U is maintained.
- Each abstract $q \in Q$ in the quantum computation is assigned at most one physical qubit $p \in P$ in the quantum computer, i.e. the assignment $\pi_{q,t}$ is injective.
- If a quantum gate $g \in U$ is assigned a location in X_g at time d_g, the gate qubits Q' of quantum gate g must have been assigned to physical qubits P', i.e π_{q_i,p_i,d_g} must evaluate to true for all $q_i \in Q'$ and $p_i \in P'$.
- The qubit assignment $\pi_{q,p,t}$ may only change to $\pi_{q,u,t+1}$ in a subsequent time step if there exists a valid edge (p, u) in the topology graph G.

A swap gate must be inserted into the quantum circuit U whenever the qubit assignment changes ($\pi_{q,p,t} \neq \pi_{q,p,t+1} \neq$). Furthermore, quantum gates may have an individual runtime $t_i \in \{1, ..., T\}$ that must be considered, i.e. during the runtime of a quantum gate g, the corresponding qubit $p \in P$ is only occupied with quantum gate g.
When a swap insertion model $\mathcal{S}$ is combined with a model $\mathcal{B}$ for topology changes two more constraints are required:

- A quantum gate (or swap gate) may only be placed on qubits $p, u \in P$ in a certain time step t if $EN_{p,u,t}$ evaluates to true, i.e. the qubit connectivity between p and u is available in the topology at time t.
- Mutual exclusion of quantum computations, topology displacement and swap gates on a set of qubits at the same time step.

6.2.4 Optimization Objectives

Objective functions guide the solver to determine a satisfiable assignment to the variables of model $\mathcal{M}$ that yield a quantum computation with preferred characteristics. The objective functions investigated in this chapter minimize the number of inserted swap gates,

maximize the expected fidelity of the compiled quantum computation or minimize the depth of the quantum computation. These objectives are expected to reduce the error on the quantum computer incurred by the compiled quantum computation.

The depth of the quantum computation is minimized in the developed model $\mathcal{M}$ by the following function:

$$\min Z, Z \geq t \wedge d_{g,t}, \forall g \in U, \forall t \in \{1, ..., T\}, \tag{6.5}$$

where g is a quantum gate in the input quantum computation and Z is an ancilla integer variable. The expected fidelity of the quantum computation can be optimized using:

$$\max \sum_{c \in U} \log(f_c) + \sum_{s \in S} \log(f_s) + \sum_{d \in D} \log(f_d),$$

where f_c is the fidelity of a computation c, S is the set of swap gates, f_s is the fidelity of a swap gate $s \in S$, D is the set of one-dimensional topology displacements and f_d is the fidelity of a displacement $d \in D$. The minimization of inserted swap gates was not evaluated in this work.

Example — One-dimensional Displacements for Rigorous Compilation

In this section, the developed model $\mathcal{M}$ is demonstrated on a quantum computation that should accommodate limited qubit connectivity of a target quantum computer at a lower overhead by using one-dimensional displacements and swap gate insertions. Figure 6.4 shows a six-qubit quantum computation and a topology graph G defined by an array with three columns and two rows.

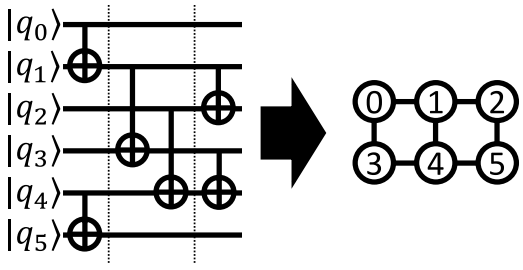


Figure 6.4: Example quantum computation (left) to be compiled for an example topology graph (right).

The given six-qubit quantum computation contains six two-qubit quantum gates out of which the two quantum gates between qubit 1 and qubit 3 (CX(1, 3)) and between qubit 4 and qubit 2 (CX(4, 2)) can not be computed on the specified topology directly. The given quantum computation is compiled using the rigorous method developed in this section to accommodate the limited qubit connectivity given by G using one-dimensional topology displacements and swap gate insertions.

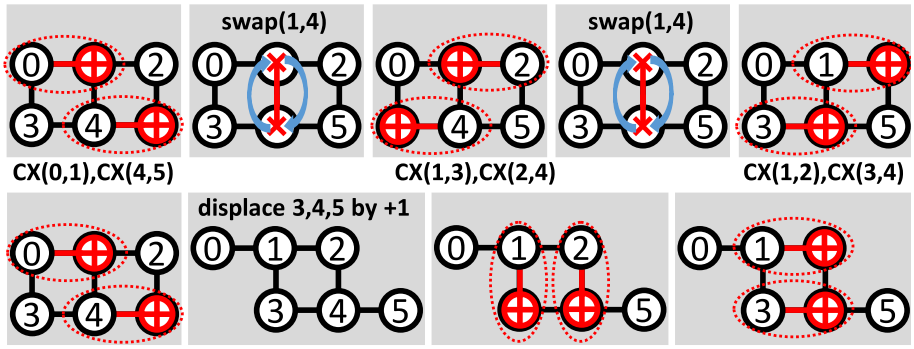


Figure 6.5: Accommodating the topology graph in Figure 6.4 with the quantum computation described in the same figure. For the top row, only swap gates are inserted while the bottom rows explore mapping with only one-dimensional displacements of the initial topology. Between these two rows, the two-qubit quantum gates executed in that time step are noted.

In the example shown in Figure 6.5, the runtime of the swap gate and the topology displacement is t_d while all other operations require unit time ($t_i = 1$). The two two-qubit gates to the left of the dashed line of the previous Figure 6.4 can be computed directly in the first time step of the given quantum computation. If the subsequent quantum gates should be mapped using only swap gate insertions, the quantum gates CX(1, 3) and CX(4, 2) can be computed in time step $t_d + 2$ by inserting either set of swap gates $s_0 = \{\text{swap}(1,4)\}$ or $s_1 = \{\text{swap}(1,0),\ \text{swap}(4,5)\}$. For both of these swap gate insertions, the resulting quantum circuit depth is $3+2t_d$. Let the swap gates in s_0 be performed until time step $t_d + 2$, then the remaining two-qubit gates can not be computed due to the limited qubit connectivity. The remaining two-qubit quantum gates would require another insertion of the swap gates in s_0, which leads to a resulting quantum circuit depth of $3+2t_d$. Let the swap gates in s_1 be performed until time step t_d+2, then the two-qubit

gate CX(0, 1) eventually requires another swap, which sets the resulting quantum circuit depth again to $3 + 2t_d$.

Now, the same quantum circuit mapping task is explored with one-dimensional topology displacements, i.e. qubits $3, 4, 5$ can be displaced by 1 ($\eta_{3,2+t_d} = \eta_{4,2+t_d} = \eta_{5,2+t_d} = 1$). Interactions between qubits $1, 3$ and qubits $2, 4$, i.e. CX(1, 3) and CX(2, 4), can be facilitated by this topology displacement in time step $2 + t_d$. The remaining two-qubit quantum gates can be computed in the subsequent time step, i.e. the resulting depth is $3 + t_d$. This example highlights the importance of exploiting one-dimensional topology displacements to reduce the depth of the mapped quantum circuit.

6.3 Evaluation

The improvement in quantum circuit depth and expected quantum circuit fidelity is investigated in this evaluation when one-dimensional topology displacements are considered in conjunction with swap gate insertions in order to map a given quantum computation. The improvement of these two metrics was evaluated on different technology parameters such as varying swap gate fidelity, as well as the topology displacement and the swap gates runtime.

The developed rigorous compilation method was evaluated on quantum circuits with up to 15 qubits and a maximum depth of 76 quantum gates. The set of evaluated quantum computations consisted of Bernstein-Vazirani (BV) [BV97], the quantum Fourier transform (QFT) [COP94] and quantum circuits that compute arithmetic functions [WGT+08]. In addition, random quantum circuits were generated and evaluated where multiple layers of two-qubit CX gates between random pairs of qubits were inserted. A topology graph of sufficient size was chosen for each evaluated quantum computation. Specifically, a quantum computation with $|Q|$ qubits was compiled to a topology graph with an array that contains R rows and C columns such that $|Q| \leq C \cdot R$ at a minimal C, R. The average runtime of the solver used in the developed rigorous compilation method was roughly 20 minutes.

6.3.1 Quantum Circuit Depth

In order to explore different technology parameters, each quantum computation was evaluated with a swap gate and topology displacement runtimes that range from one to four time steps. The reduction in quantum circuit depth when using topology displacements in conjunction with swap gates is shown in Figure 6.6.

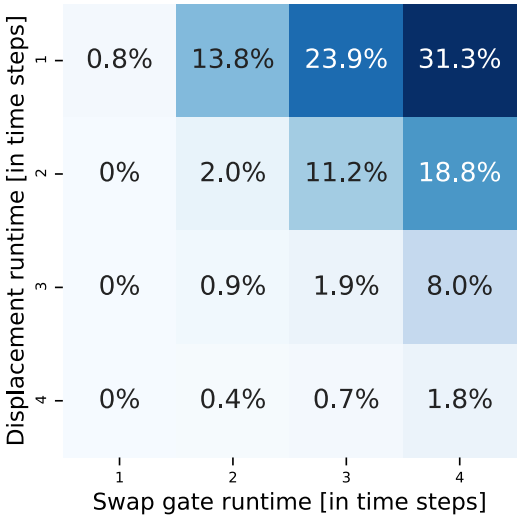


Figure 6.6: Reduction in quantum circuit depth for different one-dimensional topology displacement and swap gate runtimes.

The average reduction in quantum circuit depth ranges from 0.8% to 2% if the runtime of swap gates and topology displacements is equal. The mapping procedure will typically fall back to inserting swap gates at a particular point in the quantum computation if the swap runtime is smaller than the topology displacement runtime. However, slower topology displacements may still be viable if one displacement can replace multiple swap gates in the quantum computation. This does not appear to be the case for the evaluated quantum computations, whenever the swap runtime equals one time step and the topology displacement runtime is larger than one time step. When the topology displacement runtime is four time steps and the swap runtime is two time steps, a reduction in quantum circuit depth of 0.4% on average can be observed.

In contrast, if the topology displacement runtime is lower than the swap runtime, a large average reduction in quantum circuit depth of up to 31.3% can be observed. The ratio r between the swap gate runtime and topology displacement runtime has a major impact

on the exact quantum circuit depth improvement. The reduction in quantum circuit depth is 16.3% on average if this ratio is 2. However, r is not the only factor that determines the reduction in quantum circuit depth, as can be observed on the diagonal entries of the matrix in Figure 6.6. The ratio of the two runtimes is always 1 on the diagonal, i.e. swap gates and topology displacements take the same number of time steps. Still, the reduction in quantum circuit depth ranges from 0.8% to 2%. This difference is caused by the structure of the quantum computation, i.e. the number and position of single- and multi-qubit gates.

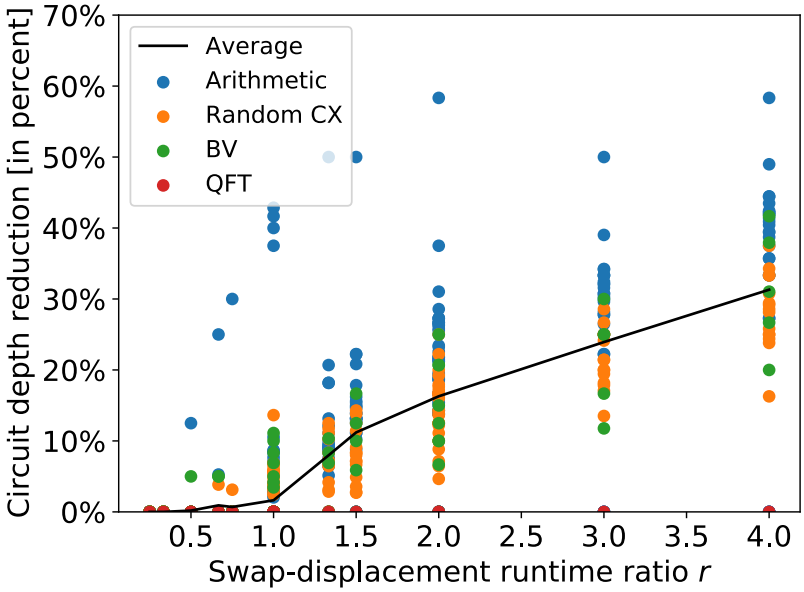


Figure 6.7: Reduction in quantum circuit depth for individual quantum circuits and different runtime ratios r.

Figure 6.7 emphasizes the impact of the circuit structure on the reduction in quantum circuit depth. For instance, the depth of QFT quantum circuits can not be reduced by the developed method at any evaluated combination of swap gate runtime and topology displacement runtime. Other quantum circuits such as arithmetic quantum circuits, however, exhibit a reduction in quantum circuit depth of up to 58.3%. If the runtime ratio r is 1, the quantum circuit depth reduction is up to 45%. The structure of the quantum circuit becomes insignificant for the reduction of quantum circuit depth when the runtime ratio r decreases. For instance, the maximum reduction in quantum circuit depth is roughly 13%, if the swap gate is twice as fast as the topology displacement. Topology displacements do

not have a tangible impact on the quantum circuit depth reduction if the swap gates have an even faster runtime.

6.3.2 Fidelity

In this subsection, the expected fidelity of a quantum circuit as given by Section 6.2.4 is investigated. Since we expect the modelled restriction on the movement of atoms through optical tweezers as given by one-dimensional topology displacements to not have a significant impact on the quantum state of the qubits, the fidelity for these topology displacements was set to 1. Successively, a swap gate fidelity of $f_s \in \{0.999, 0.995, 0.99, 0.97, 0.95\}$ was used when compiling the evaluated quantum computations. The improvement in expected quantum circuit fidelity is shown in Figure 6.8 for the developed rigorous compilation method that uses swap gates and one-dimensional topology displacements in contrast to only using noisy swap gates as in previous work. By using topology displacements in addition to swap gates a maximal fidelity improvement of 29% could be yielded at a swap gate fidelity of 0.95. However, the expected quantum circuit fidelity did not improve for all quantum circuits at all evaluated swap gate fidelities, i.e. no swap gate could be replaced by a topology displacement for some quantum circuits. The fidelity improved only by 0.3% on average for the largest swap gate fidelity of (0.999). However, the quantum circuit fidelity improved by 15% on average, for the lowest evaluated swap gate fidelity (0.95). These results highlight the benefit of one-dimensional topology displacements at the right technology parameter.

6.4 Summary

A rigorous compilation method was developed in this chapter that is able to exploit one-dimensional topology displacements available in near-term quantum architectures based on Rydberg atoms in conjunction with swap gate insertions. The developed method incurred a quantum circuit depth reduction of up to 58% and an improvement in expected quantum circuit fidelity of up to 29%, for the evaluated quantum circuits and technology parameters. One-dimensional topology displacements demonstrated the ability to improve quantum circuit mapping significantly if the swap gate fidelity is lower than 0.999 or the

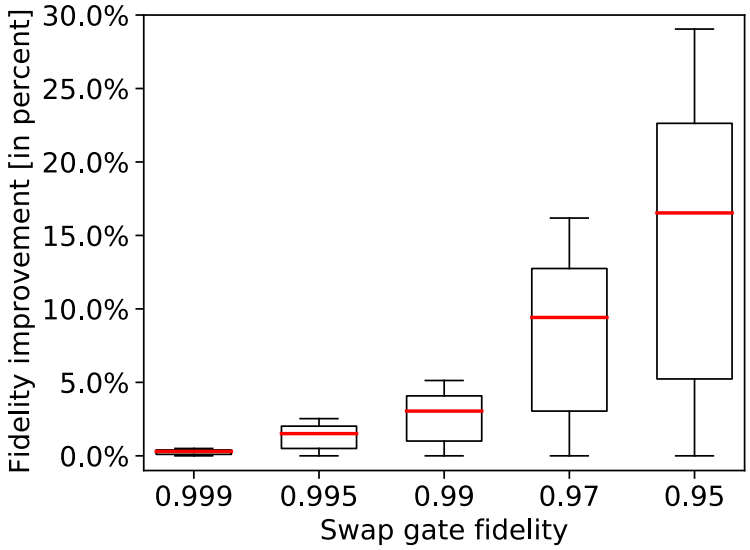


Figure 6.8: Improvement in expected quantum circuit fidelity for different swap gate fidelities $f_s \in \{0.999, 0.995, 0.99, 0.97, 0.95\}$.

swap gate runtime is not much lower than the topology displacement runtime. The developed rigorous compilation method has been demonstrated to be tractable on quantum circuits for near-term quantum computers based on Rydberg atoms. Furthermore, the technology parameters provided to the experimentalists of the Rydberg quantum computer at the project [QRy23] can help to shape future generations of term Rydberg architectures.

Chapter 7

Quantum Circuit Partitioning

Effective near-term quantum computing is hindered by the scarcity of qubits, high error rates, and limited qubit connectivity [BDG+22, Pre18, CDD+19]. These challenges can be addressed by employing quantum circuit partitioning to divide a quantum computation into individual parts that can be computed separately, either sequentially on a single quantum computer or in parallel on multiple individual quantum computers [UPR+23, LMH+23, PS23, MF21a, MF21b, PHOW20, BPS23, Ped23, HWY23, BSS16, EMG+22, TTS+21, ALRS+20, ARS+21]. Typically these individual parts exhibit a reduced qubit connectivity requirement and smaller qubit footprint, thus also reducing the incurred error and extending the computational reach [TTS+21, ALRS+20, ARS+21]. For instance, quantum circuit partitioning could be used to yield smaller (sub) computations, for an otherwise infeasible quantum computation, each of which can be performed on a near-term quantum computer.

However, quantum circuit partitioning generally incurs an exponential increase in quantum runtime, i.e. computation time on the quantum computer. This quantum runtime increase may prevent the application of quantum circuit partitioning at scale and may diminish potential quantum advantages. While employing quantum circuit partitioning, it is therefore crucial to rigorously minimize the quantum runtime overhead to extend the scope and reduce the cost of partitioning.

Specifically, the individual steps of quantum circuit partitioning are:

1. Determine a suitable partitioning for a given quantum circuit that yields sufficiently small computations.

2. By using circuit cutting or knitting techniques, resolve the dependencies between partitions to enable a separate computation [PS23, MF21b, MF21a, PHOW20, Ped23, HWY23, LMH+23, BPS23].

3. Compute the partitioned quantum computation by executing the individual subcircuits contained in the partitions.

4. Determine the result of the original quantum computation by recombining the results of the subcircuit computations.

For a class of recent dependency resolution techniques (see Sections 7.1 and 7.2 for details), the last step of quantum circuit partitioning can be performed in polynomial time [PS23, BPS23]. The overhead of the third step is mainly determined by steps one and two of quantum circuit partitioning. For step two, recent advances in quantum information yielded optimal dependency resolution techniques [PS23, BPS23, Ped23, HWY23]. These recent works have not been considered in previous work [TTS+21, ALRS+20, ARS+21, PHOW20] when a suitable partitioning is determined in the first step, thus incurring a higher cost for partitioning.
This section develops a rigorous quantum circuit partitioning method that utilizes the dependency resolution techniques introduced in [PS23, BPS23] to address the first step of quantum circuit partitioning by:

- Extending the recent dependency resolution technique in [BPS23] by introducing a novel construction for dependency resolution at minimal cost without requiring ancilla qubits for single wires.

- Enabling an optimal partitioning of quantum circuits by developing a formal model on satisfiability modulo theories (SMT).

- Constructing and evaluating the developed formal model for partitioning sparse quantum circuits [WGT+08, Mas21, FGG14, KMT+17, GHZ89].

The remainder of this chapter is organized as follows. The choice of gate cuts and wire cuts in quantum circuit partitioning is introduced in Section 7.1. The dependency resolution via quantum circuit knitting [PS23] and the novel construction for single wire

dependency resolution is described in Section 7.2. The overhead of existing dependency resolution methods is reviewed and previous quantum circuit partitioning methods [TTS+21, PHOW20, ARS+21] are compared in detail during Section 7.3. Then, the developed formal model and corresponding preprocessing steps are described in Section 7.4 and evaluated in Section 7.5. Parts of this chapter have been published as [BPK23a].

7.1 Quantum Circuit Partitioning with Gate Cuts and Wire Cuts

Quantum gates and qubit wires introduce dependencies into a quantum circuit that enforce the execution on a single connected partition, i.e. a set of connected qubits where arbitrary interactions can be performed after inserting auxiliary instructions [TC20, ZPW18, LDX19, HZW21, DSB+22, BBP21]. A gate dependency is the consequence of a multi-qubit gate that requires qubits interacting in the same multi-qubit quantum gates to be connected on the same partition. A wire dependency describes the continuity of qubit assignment in time, i.e. the qubit assignment between two quantum gates is constant. Without quantum circuit partitioning, the wire or gate dependencies in a quantum computation define the number of connected qubits required for computation. Quantum circuit partitioning allows for the reduction of that number by resolving these dependencies by operations local to the individual partitions. A wire cut resolves a wire dependency and a gate cut resolves a gate dependency. The following sections will discuss dependency resolutions and the different types of cuts in detail while this section is limited to describing an example for motivating this chapter.

An example partitioning of a quantum circuit into two partitions induced by resolving dependencies using different combinations of cuts (green lines) is depicted in Figure 7.1. The example quantum circuit consists of two sets of CNOT quantum gates arranged in distinct structures. Specifically, k_w CNOT quantum gates are arranged in a harp-like structure and k_v CNOT quantum gates are arranged in a ladder-like structure. Parts of the quantum circuits where partitioning incurs a prohibitive cost due to a high density of two-qubit quantum gates are coloured red (top) and blue (bottom). A different combination of cuts is represented by green lines in Figure 7.1 where a dashed line depicts using only gate

cuts, a dotted line depicts only using wire cuts and a solid line depicts a combined use of both types of cuts.

The result of the quantum circuit partitioning using any of the indicated cuts is two separate partitions, as will be described in more detail in subsequent sections. The selected cuts have a significant impact on the number of required cuts for the example partitioning. Using only wire cuts, the number of required cuts is $2 + 2k_v$, $k_w + k_v$ when only using gate cuts and $\min\{2, k_w\} + k_v$ when both types of cuts are used. The number of qubits required on a quantum computer is reduced by a selection indicated by a green line, i.e. the example quantum circuit required $L + M$ qubits initially which is reduced to $\max\{L, M\}$ qubits.

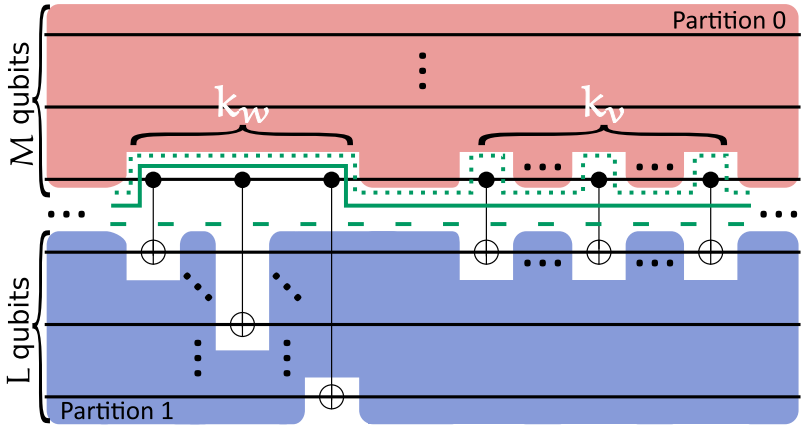


Figure 7.1: An example quantum circuit partitioned into two parts using different combinations of cuts depicted by green lines. While partitioning within the red (top) and blue (bottom) coloured blocks incur prohibitive costs, using only gate cuts (dashed line), only wire cuts (dotted line), or both (solid line) yields the desired partitioning.

Depending on the structure of a quantum circuit, using only wire cuts can occasionally yield a lower number of cuts for partitioning than only using gate cuts (and vice versa). For an arbitrary quantum circuit, however, a minimal cost partitioning can only be determined when both types of cuts are employed. Furthermore, the minimal cost of a cut varies significantly depending on the type of gates cut, the availability of ancilla qubits, classical communication [PS23, BSS16, EMG+22, MF21a, MF21b, PHOW20, TTS+21, ALRS+20, ARS+21, LMH+23, UPR+23, BPS23, Ped23, HWY23].

7.2 Gate Cuts and Wire Cuts via Quantum Circuit Knitting

During quantum circuit partitioning, qubits and quantum gates are assigned to distinct smaller partitions that then can subsequently be executed separately from each other. In general, partitioning will thus violate gate and/or wire dependencies that need to be resolved by methods such as circuit knitting, i.e. the effects of such violations are overcome [PS23, BSS16, EMG$^+$22, MF21a, MF21b, PHOW20, TTS$^+$21, ALRS$^+$20, ARS$^+$21, LMH$^+$23, UPR$^+$23, BPS23, Ped23, HWY23].

A (connected) quantum circuit can be divided into separate partitions after these dependencies have been resolved. The result of the unpartitioned quantum circuit computation is then estimated by executing quantum subcircuits defined on the determined partitions. The individual computation results (samples) of these quantum subcircuits are combined to estimate the result of the unpartitioned quantum circuit. A quantum subcircuit contains the qubits and quantum gates assigned to their corresponding partition and potentially additional operations inserted by the employed dependency resolution method. Quantum subcircuits can be computed separately from each other, i.e. quantum gates do not occur between quantum subcircuits that are at most connected to each other via classical communication [PS23, BPS23, IBM22]. Furthermore, quantum subcircuits typically require a lower number of qubits and duration individually than the unpartitioned quantum circuit. The total quantum runtime of a partitioned quantum circuit is dominated by the number of times these subcircuits must be executed, i.e. the sampling overhead grows exponentially for every resolved dependency. Thus, a crucial objective for quantum circuit partitioning is therefore the minimization of the sampling overhead.

In quantum circuit knitting (see Figure 7.2) [PS23, MF21b, MF21a], a gate is cut by representing the unitary channel $\mathcal{U}$ corresponding to a quantum gate g by a quasiprobability decomposition (QPD)

$$\mathcal{U} = \sum_i a_i \mathcal{F}_i, \tag{7.1}$$

with local operations (including classical communication) $\mathcal{F}_i$ and weights a_i that indicate how the computation of $\mathcal{U}$ is estimated. Each time a partition is computed that includes the quantum gate g, the quantum gate g is randomly replaced by operations $\mathcal{F}_i$ yielding a distinct quantum subcircuit with a probability that depends on a_i. The sampling overhead

scales as κ^2 with $\kappa := \sum_i |a_i|$ for each cut gate. Different QPDs may decompose the same quantum gate at an individual κ-factor [PS23], where the γ-factor is the minimal κ-factor of a gate. The magnitude of the γ-factor depends on the entanglement generated by a quantum gate as, as quantified by the robustness of entanglement [VT99, PS23], i.e. a more entangling gate incurs a larger γ factor. The result of the original quantum circuit is estimated by a simple sum of the measurement outcomes of the subcircuits weighted by their corresponding a_i [PS23].

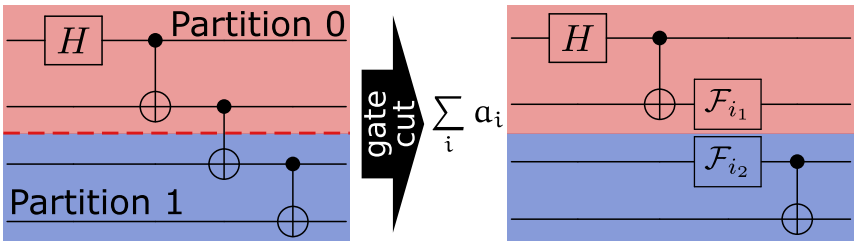


Figure 7.2: Using quantum circuit knitting to employ a gate cut for computing a 4-qubit GHZ state on two partitions.

The sampling overhead can be reduced in general [PS23, Ped23, HWY23] when multiple gate cuts are realized simultaneously, i.e. simultaneous cuts, via quantum gate teleportation that require ancilla qubits and classical communication. Quantum gate teleportation [NC11] is a protocol that realizes a quantum gate by consuming a preexisting Bell state between the qubits of the quantum gate in addition to local operations. Thus, a quantum gate that prevents a partitioning, can be replaced by a quantum gate teleportation construction where the preparation of the shared preexisting Bell state is cut. The sampling overhead of cutting simultaneous Bell state preparation is lower than individually generating and cutting these Bell states as entanglement within one partition can be exploited [PS23] (see Section 7.3 for details). Preparing these Bell states, however, incurs one additional ancilla qubit per partition and the quantum gate teleportation protocols require classical communication between partitions.

Wire Dependency Resolution via Gate Cuts

In this section, a construction is shown that allows the transformation of a wire dependency to a gate dependency that can then be cut by regular quantum circuit knitting (see

previous Figure 7.2) [PS23]. In Figure 7.3, a set of quantum gates (indicated by a boxed vertical arrow) is inserted at the location of a wire cut. The inserted set of quantum gates transfers the state left to the wire cut to an ancilla qubit that is used in subsequent operations. In a subsequent step, a gate cut is applied to the inserted set of quantum gates, thus removing the newly introduced gate dependencies. By resolving these gate dependencies, the qubit at the left of the wire cut and the qubit to the right of the wire cut can be in different partitions. Note that in the absence of qubit reuse techniques [HJC+22, BPK23b] (see the right of Figure 7.3), the sum of overall used qubits increases by one for each wire cut.

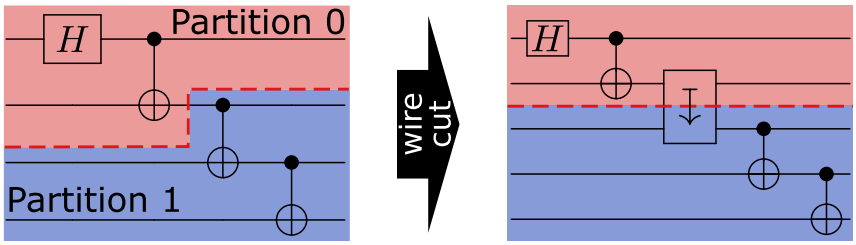


Figure 7.3: Resolving a wire dependency via gate cuts to divide a 4-qubit GHZ state into two partitions.

This functionality can be realized in different ways to replace the wire dependency with gate dependencies: e.g. by inserting a swap gate, a quantum teleportation circuit [BPS23], or the quantum circuit described in Figure 7.4, which we call a 'move' circuit. Different costs in terms of ancilla qubits, sampling overhead, and the requirement for classical communication are incurred by these options. The lowest possible sampling overhead (see Section 7.3) is incurred by the 'move' circuit and the quantum teleportation circuit. However, they also require the ability to perform classical communication between separate partitions to yield the minimal sampling overhead and the quantum teleportation circuit furthermore requires one extra ancilla qubit.

We now show that the 'move' quantum circuit transfers the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ of a qubit to a fresh ancilla qubit in the $|0\rangle$ state. In the 'move' quantum circuit, a Bell measurement and a subsequent phase correction are applied [NC11]. The two-qubit state $|\psi\rangle|0\rangle$ is transformed after the first CNOT gate in the 'move' quantum circuit to

$$|\psi_1\rangle = \alpha|00\rangle + \beta|11\rangle\,. \tag{7.2}$$

Applying the Hadamard gate to the top qubit leads to

$$|\psi_2\rangle = \alpha\left(|0\rangle + |1\rangle\right)|0\rangle + \beta\left(|0\rangle - |1\rangle\right)|1\rangle\,. \tag{7.3}$$

Measuring the top qubit in the computational basis yields either a zero or a one with equal probability. When measuring a 'zero' on the top qubit, the state of the bottom qubit becomes:

$$\langle 0| \cdot |\psi_2\rangle = \alpha(0+1)\,|0\rangle + \beta(0+1)\,|1\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{7.4}$$

and measuring a 'one' on the top qubit, the state of the bottom qubit is

$$\langle 1| \cdot |\psi_2\rangle = \alpha(0+1)\,|0\rangle + \beta(0-1)\,|1\rangle = \alpha\,|0\rangle - \beta\,|1\rangle\,. \tag{7.5}$$

Thus, the state $|\psi\rangle$ can be prepared on the bottom qubit by correcting the phase, i.e. applying a Z-gate on the bottom qubit depending on the measurement result on the top qubit.

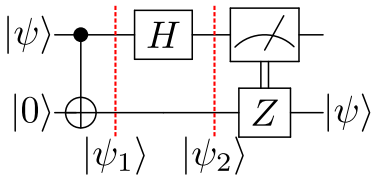


Figure 7.4: The 'move' quantum circuit transfers the state $|\psi\rangle$ from the top qubit to the bottom qubit without further ancilla qubits; $|\psi_1\rangle$ and $|\psi_2\rangle$ are the states corresponding to Equation (7.2) and Equation (7.3), respectively. By inserting this circuit, a wire cut can be reduced to gate cutting one single CNOT gate.

7.3 Related Work

The overhead incurred by the gate and wire dependency resolution methods is summarized in this section. Furthermore, previous quantum circuit partitioning methods [TTS+21, PHOW20, ALRS+20] are compared to the method developed in this chapter. Previous dependency resolution techniques [PS23, BSS16, EMG+22, MF21a, MF21b, PHOW20, TTS+21, ALRS+20, ARS+21, LMH+23, UPR+23, BPS23, Ped23, HWY23, LMH+23] can be distinguished by their application to gate cuts [UPR+23, BSS16,

EMG$^+$22, MF21b] or wire cuts [PHOW20, MF21a, BPS23, Ped23, HWY23, LMH$^+$23, TTS$^+$21, ALRS$^+$20, ARS$^+$21]. These works can further be distinguished by assuming the availability of classical communication (CC) and ancilla qubits to reduce the overhead of dependency resolution [PS23, BPS23, LMH$^+$23, HWY23] and not requiring classical communication [MF21a, MF21b, PHOW20, TTS$^+$21, ALRS$^+$20, ARS$^+$21].

The minimal sampling overhead of dependency resolution techniques is summarized in Table 7.1. The sampling overhead varies significantly depending on the availability of CC as well as ancilla qubit insertions, allowing for reductions of up to 75% for k arbitrary wire cuts. Furthermore, the exact type and number of cuts also have an impact on the sampling overhead.

A multiplicative sampling overhead of 9 (e.g. this work and [BPS23, HWY23, Ped23]) is incurred for a single individual wire cut when CC and ancilla qubits are available and 16 otherwise [PHOW20, TTS$^+$21, ALRS$^+$20, BPS23, Ped23, HWY23, LMH$^+$23]. Recent approaches in [Ped23, HWY23] and in this chapter achieve this sampling overhead for a single wire cut without additional ancilla qubits. k arbitrary wire cuts incur a sampling overhead of 16^k without ancilla qubits and CC and the minimal sampling overhead of $(2^{k+1}-1)^2$ otherwise [PS23, Ped23, HWY23].

The sampling overhead of cutting quantum gates is also given in Table 7.1. In this chapter, we review CNOT quantum gates, swap gates, and the conditional-rotation (CR) quantum gate. The multiplicative sampling overhead is 9^k [MF21a], 49^k [MF21b], and $(1+2|\sin(\theta)|)^{2k}$ [PS23] respectively for cutting k CNOT, swap and conditional-rotation quantum gates without CC and ancilla qubits. Otherwise, the sampling overhead is reduced to $(2^{k+1}-1)^2$, 16^k, and to less than 4^k for simultaneously cutting k arbitrary CNOT, swap, and conditional-rotation quantum gates respectively [PS23]. The minimal sampling overhead of other classes of quantum gates is largely unknown [PS23], as determining a QPD and the corresponding γ-factor for an arbitrary quantum gate is computationally hard.

Related Work on Quantum Circuit Partitioning

In previous quantum circuit partitioning methods such as [TTS$^+$21, ALRS$^+$20, PHOW20], dependencies are resolved by using only wire cuts. Furthermore, these methods assume

no further ancilla qubit insertions and no available CC. As established in the previous Table 7.1, however, CC and ancilla qubits dominate the cost of partitioning along the number of applied cuts. Specifically, wire cuts incur a sampling overhead of 16^k in these methods, while the theoretical minimum scales as $\mathcal{O}(4^k)$ with CC and ancilla qubits. In addition, these quantum circuit partitioning methods also incur a larger number of cuts depending on the structure of the quantum circuit (see Figure 7.1) compared to the combined use of gate cuts and wire cuts in this work. Previous work on partitioning also does not consider gate cuts, thus they are not able to exploit a potential low sampling overhead when gate cutting e.g. CR gates.

Another difference is observed in the runtime of the classical postprocessing required for quantum circuit partitioning. The classical postprocessing step is a major bottleneck for previous work [TTS+21, ALRS+20, ARS+21], where an exponential classical runtime overhead is incurred in addition to the exponential increase in quantum runtime, thereby essentially limiting the applicability of such methods. In the method developed in this chapter, quantum circuit partitioning is based on quantum circuit knitting that allows for an efficient (polynomial time) classical postprocessing [PS23].

In summary, previous quantum circuit partitioning methods require a larger number of cuts at a larger increase in classical runtime and quantum runtime.

Table 7.1: Minimal sampling overhead of cutting depending on cut type, ancilla qubits and classical communication (CC).

	Wire Cuts		k Gate Cuts		
	Single	k Arbitrary	CNOT	swap	CR(θ)
CC and Ancilla	9	$(2^{k+1}-1)^2$	$(2^{k+1}-1)^2$	16^k	$(\leq 4)^k$
No CC, no Ancilla	16	16^k	9^k	49^k	$(1+2\lvert\sin(\theta)\rvert)^{2k}$

Number of Available Cuts for Quantum Circuit Partitioning

In addition to the availability of CC and ancilla qubits, the applicability of quantum circuit partitioning depends significantly on the number of required cuts. The quantum runtime of quantum circuit partitionings requiring k (x-axis) cuts is shown on the y-axis of Fig-

ure 7.5. Furthermore, the sampling frequencies supported by a quantum computer and multiplicative sampling overheads incurred by a cut are also varied while assuming eight thousand measurements as a basis. The investigated sampling overheads can e.g. be incurred by cutting k CNOT gates simultaneously with CC and ancilla qubits ($(2^{k+1}-1)^2$) or by cutting single wires at a minimum cost of 9^k (e.g. achieved by the construction in this chapter). The cost of wire cutting (16^k) in previous quantum circuit partitioning work is also investigated [TTS+21, ALRS+20, PHOW20].

Assuming a total quantum runtime of one day at a sampling frequency of 1 kHz, 1 MHz, and 1 GHz, 5, 10, and 15 cuts, respectively would be able to be accommodated at the minimal sampling overhead per CNOT cut. At a sampling overhead of 9^k (16^k) per cut, however, only 4, 7, 10 (3, 5, 8) cuts can be accommodated at the assumed sampling frequencies. A realistic repetition rate for current quantum computers is in the order of roughly 1 kHz to 20 kHz [Goo22, IBM23b].

A runtime-intensive and rigorous quantum circuit partitioning approach is warranted at the limited number of cuts available at even large quantum computing budgets.

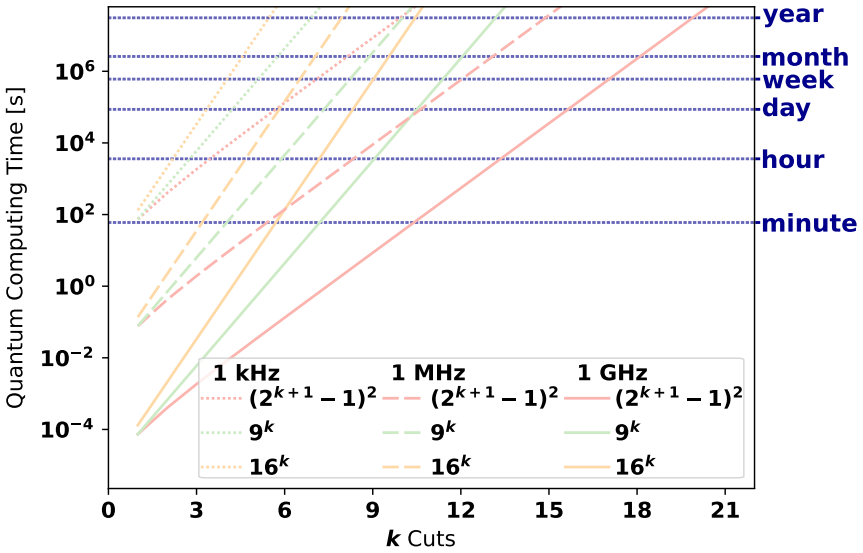


Figure 7.5: Quantum runtime required for k cuts incurring a multiplicative sampling overhead of $(2^{k+1}-1)^2$, 9^k and 16^k at different sampling frequencies.

7.4 Rigorous Quantum Circuit Partitioning

The developed satisfiability modulo theories model (SMT), the preprocessing steps, model variables, model constraints, and evaluated objective functions are described in this section. In Figure 7.6, the individual steps of the developed rigorous quantum circuit partitioning method are shown. A given quantum circuit is input to a preprocessing step first where a representation based on two-qubit quantum gates is generated and where a cutting graph is determined that is a basis for the developed formal model. The SMT model is then constructed from the determined cutting graph, given a quantum circuit, a description of the QPDs including their γ-factor and a specification of the available quantum computing resources in terms of qubits, quantum time, and sampling frequency. These quantum computation characteristics are used to determine a suitable partition size and the maximum incurred sampling overhead. The developed formal model is then input to an SMT solver [MB08] that determines an assignment to model variables subject to a given objective function. This variable assignment completely specifies a quantum circuit partitioning that is minimal with respect to the given QPD if that partitioning is possible with the given quantum computation characteristics. Otherwise, the solver proves that such a quantum circuit is not possible.

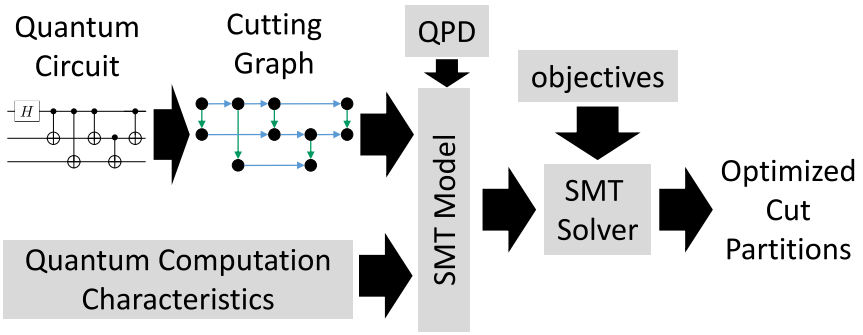


Figure 7.6: Individual steps of the rigorous quantum circuit compilation method.

7.4.1 Preprocessing

A given quantum circuit is first compiled to two-qubit gates [Qis23] before a cutting graph $\mathcal{G} = (V, G \cup W)$ is derived. The cutting graph represents the effect of cuts on the partitioning of a given quantum circuit, i.e. a wire cut or gate cut removes an edge from $\mathcal{G}$; if

the graph becomes disconnected by a set of cuts, the disconnected (disjoint) vertex subsets correspond to computations on distinct partitions. Essentially, a minimum weight cut [CLRS22] must be determined on the cutting graph while satisfying a set of additional constraints corresponding to simultaneous cuts and the size of a partition.

The cutting graph $\mathcal{G}$ consists of two edge sets $G \cup W$ where W represents wire cuts and G represents quantum gate cuts. In the first step, single-qubit gates are removed from the input quantum circuit and thus from the cutting graph. Then, for a two-qubit quantum gate g with qubits g_u and g_v, vertices g_u and g_v are inserted into the vertex set of graph $\mathcal{G}$. Furthermore, an edge corresponding to cutting that gate is inserted $s_g = (g_u, g_v) \in G$. An edge representing a wire cut is added in a subsequent step to the edge set W for each overlapping qubit between a pair of consecutive two-qubit gates (h, k), where k is an immediate successor of h. Specifically, for each vertex $a \in s_h$ and vertex $b \in s_k$ that act on the same qubit, an edge $(a, b) \in W$ is added. In Figure 7.7, this preprocessing step is exemplarily applied to a quantum circuit with blue wire cut edges W and green gate cut edges G.

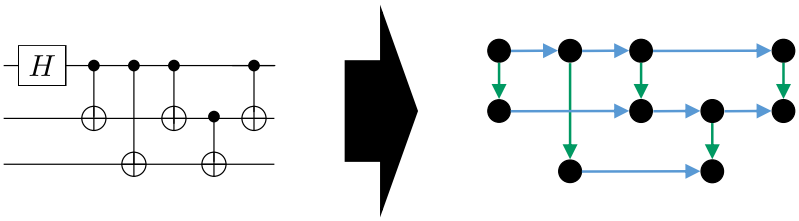


Figure 7.7: A given quantum circuit (left) is transformed into a cutting graph (right). Blue edges represent wire cuts and green edges represent gate cuts.

7.4.2 Model Variables

Given the cutting graph $\mathcal{G} = (V, E)$ with $E := (G \cup W)$ and a set of partitions P, the variables of the developed SMT model are:

- $O = \{o_{1,1}, ..., o_{|V|,|P|}\}$—this set of variables represents the assignment of a gate qubit vertex v to a partition p such that $o_{v,p}$ evaluates to true if the gate qubit vertex v is assigned to partition p.

- $C = \{c_1, ..., c_{|E|}\}$—this set of variables represents the selection of a cut represented by edge e for achieving a specific partitioning, i.e. c_e evaluates to true if the gate cut or wire cut represented by edge e is selected.

- $B = \{b_1, ..., b_{|E|}\}$—this set of variables indicates when a cut represented by edge e is cut simultaneously using a quantum gate teleportation protocol (see Section 7.3), i.e. b_e evaluates to true if the cut at edge e is simultaneous.

An upper bound for the maximum number of partitions $|P|$ can be determined by e.g. the size of the vertex set V. In this case, every vertex would be assigned a different partition, which also upper bounds the sampling overhead.

7.4.3 Model Constraints

The exact cut positions are determined by a mismatch in variables $o_{v,p} \in O$ for two adjacent vertices, as these variables define the assignment of gate qubit vertices to partitions. Therefore, the assignment to a cut variable $c_{e=(u,v)}$ is completely determined by

$$c_{e=(u,v)} := \bigvee_{p \in P} o_{u,p} \neq o_{v,p}, \tag{7.6}$$

where $e = (u, v) \in E$ and each vertex $v \in V$ is assigned to exactly one partition. This cardinality constraint is enforced via two sets of constraints where the first set enforces that no vertex is assigned to two different partitions

$$o_{v,p} \rightarrow \neg o_{v,p'} \qquad p, p' \in P \text{ with } p \neq p' \tag{7.7}$$

and the second set of constraints enforces that at each vertex is assigned at least one partition

$$\bigvee_{p \in P} o_{v,p}. \tag{7.8}$$

Selecting a cut to be simultaneous is encoded by variables b_e that imply c_e, i.e. b_e can be chosen arbitrarily as long as c_e evaluates to true and the partition size is not exceeded.
The number of partitions in the determined solution is given by the assignment to variables $o_{v,p} \in O$. These variables also determine the partition size, i.e. number of qubits Q_p in a

partition p, together with the assignments to cut variables c_e and b_e:

$$Q_p := \sum_{v \in I \subseteq V} o_{v,p} + \sum_{e=(u,v) \in W} c_e \wedge o_{v,p} + \sum_{e=(u,v) \in E} b_e \wedge (o_{v,p} \vee o_{u,p}), \tag{7.9}$$

where I is the set of vertices without incoming edges from set W, i.e. the vertices corresponding to the first two-qubit quantum gate on each qubit. Note that wire cuts (represented by edge set W) inherently increase the number of qubits in a partition p.

Constraints defining the domain of the model variables have been omitted in this description. A partitioning with wire cuts only can be obtained by requiring cut variables c_g with $g \in G \subseteq E$ to evaluate to false.

7.4.4 Objective Functions

In this chapter, two objective functions have been evaluated in different settings. The first objective is devised to minimize the incurred quantum runtime, i.e. increase in samples. The maximal partition size, i.e. the required number of qubits in a quantum computer for executing a quantum circuit, is minimized by the second objective. Specifically, the quantum runtime is minimized by

$$S := \prod_{e \in E} \gamma_e^2 (c_e \wedge \neg b_e) \prod_{e \in E} \gamma_{e^{(k)}}^2 b_e, \tag{7.10}$$

where $\gamma_{e^{(k)}}^2$ represents the individual multiplicative overhead of a cut in k simultaneous cuts and $\gamma_{e^k}^2$ represents the multiplicative overhead of an individual cut represented by edge $e \in E$. The γ-factors used in this work are reported in Table 7.1. When the set of satisfiable assignments to the SMT model is minimized using the objective S in Equation (7.10) a minimum cost partitioning is determined. The objective in Equation (7.10) can be linearized by introducing auxiliary variables and by applying the logarithm. The maximum partition size can be minimized by the objective

$$\min Q, \text{with } Q \geq Q_p \qquad \forall p \in P, \tag{7.11}$$

where Q_p is defined by Equation (7.9).

7.5 Evaluation

Three aspects of the developed rigorous quantum circuit partitioning method are evaluated in this section. The impact of a decreasing partition size on the quantum runtime is quantified. A decreasing partition size, i.e. partitions with a decreasing number of qubits, implies that quantum computers with a decreasing number of qubits are able to perform a given quantum computation. In a second evaluation, the point of view is reversed; the impact of the available quantum runtime budget on the ability to decrease the partition size is investigated. This section is then concluded by demonstrating a reduction in quantum runtime for the developed rigorous quantum circuit partitioning method compared to the quantum runtime incurred by previous quantum circuit partitioning work [TTS+21, ALRS+20, PHOW20]

The used SMT solver was given a timeout of one hour for one quantum circuit in the set of evaluated quantum circuits. The evaluated quantum circuits had up to 40 qubits, depths of up to 51 quantum gates and consisted of arithmetic functions [WGT+08, Mas21], GHZ state preparation [GHZ89], hardware-efficient ansatz circuits [KMT+17], and QAOA quantum circuits for solving the maximum cut problem. The input to the maximum cut problem consists of random connected graphs where an additional number of random edges was inserted that amount to 10%, 30%, 50% and 100% of the vertex set size. Qiskit was used to compile the evaluated quantum circuits into two-qubit during the preprocessing step [Qis23]. The quantum runtime budget was assumed to be one day at a sampling frequency of 1 MHz, which equals roughly 10^{11} samples unless otherwise noted.

7.5.1 Quantum Runtime for a Decreasing Partition Size

The overhead in the number of samples is shown in Figure 7.8 for a decrease in partition size. More precisely, the number of qubits in a partition is reduced by a fraction of 2, 3, and 4 compared to the number of qubits in the given quantum computation. For each of these qubit fractions, an additional 10%, 30%, or 60% ancilla qubits were allowed on that partition. These ancilla qubits allow the utilisation of wire cuts to a larger extent

and reduce the quantum runtime by enabling simultaneous cuts. For instance, given a connected four-qubit quantum circuit that should be partitioned into two-qubit partitions. Gate cuts can accomplish such a partitioning in principle as they do not require ancilla qubits. However, when employing wire cuts, which inherently require ancilla qubits, one partition would be required to have at least three qubits due (see Figure 7.3). Note that the number of partitions is at least $2, 3$, or 4 when dividing a given quantum computation into partitions with qubit fractions of $2, 3$, and 4.

The largest quantum runtime for all investigated partition sizes was incurred when partitioning without ancilla qubits. At a qubit fraction of two, the lowest quantum runtime is observed. This is in part because a lower number of cuts are expected to be required for suitable partitioning on larger partitions. But also in part because simultaneous cuts can be applied most often when a given computation is only divided into two partitions. The largest cost in quantum runtime is incurred for the smallest partition size at a qubit fraction of four. In these cases, the sampling overhead can not be reduced further by simultaneous cuts.

The developed rigorous quantum circuit partitioning method was demonstrated to be able to reduce the number of qubits required for the evaluated quantum computations by a fraction of two, three and four. Hereby, the computational reach of quantum computers could be extended on the given quantum runtime budget. In addition, ancilla qubits are crucial for reducing the quantum runtime overhead of quantum circuit partitioning for specific partition sizes.

7.5.2 Reduction in Qubits for Increasing Computing Budget

The average reduction in partition size is shown in Figure 7.9 for various quantum runtime budgets given by sampling frequency of 1 kHz to 10 MHz and a quantum runtime of one hour to one month. Each cell of the array in Figure 7.9 shows the average reduction in partition size over all evaluated quantum computations. The average reduction in partition size ranges from 40.6% to 61.6% depending on the given quantum runtime budget. Thus, while the largest reduction in qubit number requires large quantum computing budgets, significant reductions can be yielded at a limited quantum runtime budget. Thus, the developed rigorous quantum circuit partitioning method can double the size of quantum

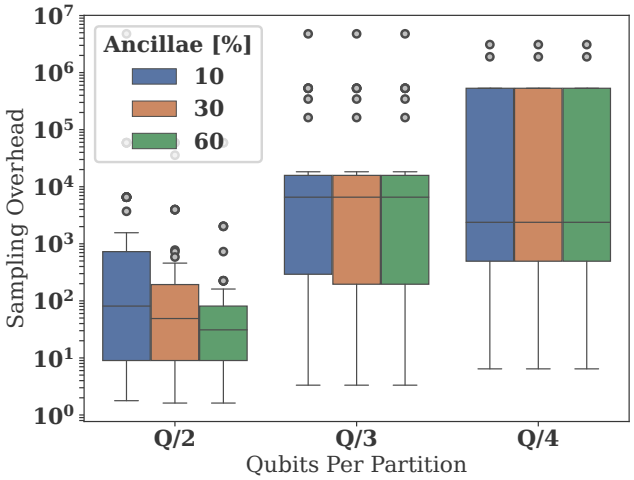


Figure 7.8: Sampling overhead for partitions with at most $Q/2, Q/3, Q/4$ qubits and an additional $10\% - 60\%$ ancilla qubits where Q qubits are required for the unpartitioned quantum computation.

computations available on near-term quantum computers at a limited quantum runtime budget.

7.5.3 Advantage of Combining Gate Cuts and Wire Cuts

In this section, the reduction in sampling overhead yielded by the method developed in this chapter is compared to previous quantum circuit partitioning work relying only on wire cuts without classical communication [TTS$^+$21, ALRS$^+$20, ARS$^+$21]. The developed rigorous quantum circuit partitioning method is configured to halve the number of qubits required for a quantum computation once using wire cuts only and once using wire cuts and gate cuts in conjunction. When using only wire cuts for quantum circuit partitioning, classical communication (CC) and thus simultaneous cuts are expected to not be available. The sampling overhead of using only wire cuts without CC is shown on the left of Figure 7.10 and compared to the method developed in this chapter (right). The ancilla qubits available to a partition range from 10%, 30%, and 60%. The developed rigorous quantum circuit partitioning method is able to significantly reduce the sampling overhead by employing (potentially simultaneous) gate cuts and wire cuts. For roughly 56% of the evaluated quantum circuits, the developed method was able to reduce the sampling

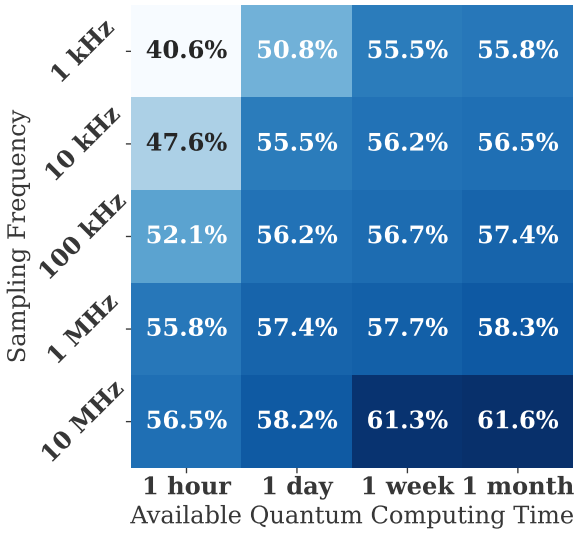


Figure 7.9: Reduction in number of qubits required for a given quantum computation at varying quantum runtime budgets.

overhead by 73% on average compared to previous work relying on only wire cuts for partitioning [TTS+21, ALRS+20, ARS+21].

When only employing wire cuts, a larger amount of ancilla qubits per partition is required to partition larger quantum computations with a larger number of cuts. A larger number of ancilla qubits also reduces the sampling overhead of the developed method through simultaneous cuts, allowing for otherwise prohibitive quantum circuit partitionings. Specifically, using only wire cuts at 10% ancilla qubits resulted in 41% of infeasible quantum circuit partitionings, i.e. the qubit requirement could not be halved for 41% evaluated quantum circuits at the given quantum runtime budget. When 30% ancilla qubits are available, the number of infeasible partitionings reduces to 14% and further to 8% when 60% ancilla qubits are available. In general, the combined use of gate cuts and wire cuts required fewer ancilla qubits to partition the evaluated quantum circuits within the given quantum computing budget. Using 10% ancilla qubits, yielded only 3% infeasible quantum circuit partitionings. This reduced to 2% for 30% ancilla qubits or more. In addition to enabling a halving in qubit requirement for a larger portion of the evaluated quantum circuit, the respective quantum circuit partitioning incurred a lower quantum runtime.

The importance of using wire cuts and gate cuts in combination with CC and additional ancilla qubits per partition is highlighted by these results.

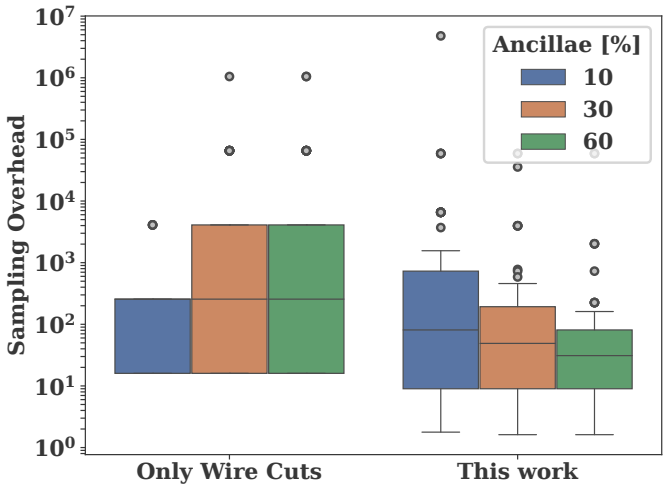


Figure 7.10: Sampling overhead of partitioning when relying on using only wire cuts at minimal individual cost (previous work [TTS+21, ALRS+20, ARS+21, PHOW20]) compared to the combined use of gate cuts and wire cuts with $10\% - 60\%$ ancilla qubits per partition and classical communication.

The results in Figure 7.10 were obtained by the method developed in this work with (simultaneous) gate cuts disabled. This restriction allows for comparison with the capabilities of previous partitioning work where (simultaneous) gate cuts are not used and the typical objective is the reduction of classical postprocessing runtime [TTS+21, ALRS+20, ARS+21]. The methods developed in previous work on quantum circuit partitioning were not directly invoked [TTS+21, ALRS+20, ARS+21]. The minimal sampling overhead for wire cuts [BPS23] was nevertheless used as a basis for comparison.

7.6 Summary

A rigorous quantum circuit partitioning method that combines wire cuts and gate cuts to soften qubit-number, error, and latency limitations of today's and near-term quantum computers was developed in this chapter. In the developed method, recent approaches on gate cutting and wire cutting are considered that use classical communication in addition to ancilla qubits to reduce the incurred sampling overhead further. Furthermore, a 'move' quantum circuit was constructed that can be used to transform a wire dependency to

a gate cut using quantum circuit knitting with minimal overhead and no cost in ancilla qubits.

A 73% reduction in quantum runtime was demonstrated by the developed rigorous quantum circuit partitioning method on average for 56% of the evaluated quantum circuits over previous work that only uses wire cuts [TTS+21, ALRS+20, PHOW20]. At the same time, the developed method was able to partition 98% of the evaluated quantum circuits within the given qubit and runtime budget whereas previous partitioning work successfully partitioned only 86% of the evaluated quantum circuits. Using the developed method, the number of required qubits was reduced by 41% at a quantum runtime budget of one hour at a sampling frequency of 1 kHz. At much larger quantum runtime budgets of one month at a sampling frequency of 10 MHz, the reduction in qubit number was raised to 60%.

The developed rigorous quantum circuit partitioning method was demonstrated to be effective in extending the computational reach of near-term quantum computers by reducing the number of qubits required for a given quantum computation. Furthermore, the scope of quantum circuit partitioning was significantly extended as the developed partitioning method requires less classical and quantum runtime for partitioning by using a recent approach to dependency resolution and employing gate cuts in conjunction with wire cuts.

In the future, the developed method may be extended further by considering more nuanced quantum circuit characteristics such as the duration or the circuit mapping effort when the cuts necessary for a quantum circuit partitioning are selected.

Chapter 8

Quantum Circuit Adaptation

Numerous technology platforms satisfy the requirements for universal quantum computing. However, as full fault tolerance has not been achieved by any of these technology platforms yet, quantum computations are limited in duration by gate fidelities and qubit decoherence times. Therefore reducing the gate count, transforming operations to incur less error, and reducing qubit idle time, during the adaptation of a quantum circuit is highly relevant for these noisy near-term quantum computers to achieve quantum advantage.

A quantum computation must be decomposed into a set of single- and two-qubit gates [BB02, BDD$^+$02] that is supported by the quantum computer intended for execution. Typically, technology platforms for quantum computers incur a higher error from two-qubit quantum gates than single-qubit quantum gates. To improve overall result quality, quantum circuit adaptation can be used for technology platforms that support multiple two-qubit quantum gates. In this chapter, an adaptation from the semiconductor spin qubit technology platform to the superconducting qubit technology is chosen as an example. The used semiconductor spin qubit platform supports three two-qubit gates natively that are incorporated and studied in a satisfiability modulo theories (SMT) model in varying parameter regimes for quantum circuit adaptation.

Depending on the applied physical control, the two-qubit swap, conditional phase (CPHASE), and conditional rotation (CROT) gates can be realized on the used semiconductor spin qubit technology platform. Except for the swap gate, the other two two-qubit gates can form a universal set of quantum gates with single-qubit rotations. The swap

two-qubit quantum gate is useful for accommodating the limited qubit connectivity of the investigated spin qubit platform.

In this chapter, we first describe these various two-qubit quantum gates and discuss their cost in terms of error and duration that will inform the developed SMT model Section 8.1. Subsequently, an overview of the employed quantum circuit substitution techniques is given in section 4.4. A detailed description of the developed rigorous compilation including the developed SMT model is then provided together with an example for illustration in Section 8.2. The developed rigorous compilation method is evaluated in Section 8.3. Parts of this chapter have been published as [BKNB23].

8.1 Two-qubit Gates in Semiconductor Spin Qubits

In this section, three two-qubit quantum gates are summarized as described for a quantum computer based on the semiconductor spin qubit technology platform [PRE+22, ZCL+20, WBB+14]. The semiconductor spin qubit technology platform chosen for comparison realizes the CPHASE quantum gate, the CROT quantum gate and the swap quantum gate (see Section 2.1.3 for a description of quantum gates) [PRE+22]. Several options for the realization of these two-qubit quantum gates are available in the semiconductor spin qubit technology platform [PRE+22]. In general, the realization of two-qubit quantum gates on this technology platform can be distinguished into diabatic and adiabatic approaches as well as single pulses and composite pulses. In adiabatic approaches, the operational conditions are slowly changed to allow a desired change in the state of the qubits [PRE+22, BF28, Kat50, XLL+21]. In contrast to adiabatic approaches, diabatic approaches quickly change operational conditions to induce a desired change in qubit state [PRE+22, BF28, Kat50, XLL+21]. Furthermore, the change in qubit state defined by a quantum gate can be realized by a single continuous pulse or by multiple pulses that implement the state change in conjunction [PRE+22].

Depending on the used control schemes and underlying material, a specific approach to the realization of a quantum gate may exhibit better characteristics. For instance, the adiabatic approach may yield higher fidelities but also incur a larger gate runtime than a diabatic approach for a specific spin qubit technology platform [PRE+22, XLL+21] while

a single pulse may yield a lower runtime than composite pulses, they may also induce a lower fidelity. Depending on the characteristics of the quantum computation and the quantum computer one quantum gate realization may be preferable to another. For instance, a faster realization of a quantum gate may occasionally be preferable to a longer realization even if the faster realization yields a lower quantum gate fidelity, namely if decoherence errors dominate the result quality of a quantum computation due to short decoherence times or a long computation runtime. In this chapter, we demonstrate how the different characteristics of such quantum gate realizations can be exploited to increase the overall result quality of a quantum computation.

Furthermore, implementing a specific state transformation requires a different number of two-qubit quantum gates depending on the choice of two-qubit quantum gates. For instance, two CNOT quantum gates are required to realize an arbitrary conditional rotation quantum gate whereas only one CROT gate would be sufficient [NC11]. Thus, a quantum gate realization may be preferable to others even if its duration and fidelity are worse.

A summary of the fidelities and durations of different quantum gate realization following the description in [PRE+22] is reported in Table 8.1 with gate times D_0. The (diabatic) CZ quantum gate is denoted as CZ (CZ_{db}), the conditional rotation gate is denoted as CROT, the (composite) SWAP quantum gate is denoted by SWAP ($SWAP_c$). Since different materials or control schemes may be required for a scaled-up spin qubit technology platform, we explore different quantum gate runtimes D_1 in Table 8.1 that are investigated in this chapter as a further comparison.

Table 8.1: Investigated quantum gate durations and fidelities.

	SU(2)	CZ	CZ_{db}	CROT	$SWAP_d$	$SWAP_c$
Fidelity	0.999	0.999	0.99	0.994	0.99	0.999
Duration D_0 [ns]	30	152	67	660	19	89
Duration D_1 [ns]	30	151	7	660	9	13

8.2 Rigorous Quantum Circuit Adaptation

The steps of the proposed rigorous compilation method for adapting a quantum computation from one technology platform to another are shown in Figure 8.1. First, the set

of two-qubit blocks, their dependencies and their cost in terms of fidelity and duration are determined for an input quantum computation. Then, every given quantum circuit substitution rule is evaluated on the two-qubit blocks of the quantum computation. The determined blocks, block characteristics, the specified quantum circuit substitution rules and given objective function are used to construct a satisfiability modulo theory (SMT) model in a third step. An assignment to the variables of this SMT model is then computed by an SMT solver such that the given objective function is optimized [MB08]. The determined variable assignment then directly defines a quantum circuit adaptation using a selection of the given quantum circuit substitution rules.

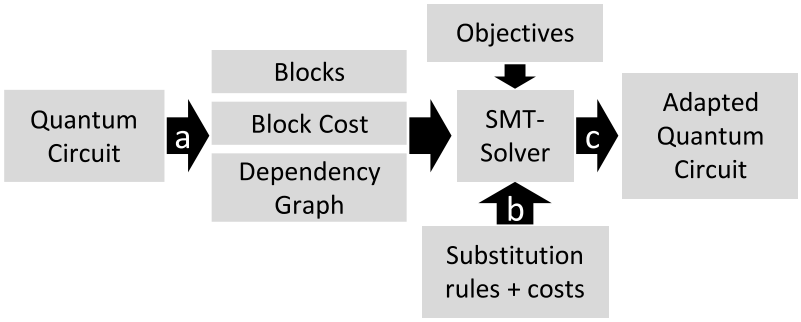


Figure 8.1: Individual steps of the developed quantum circuit adaptation method for given quantum computation with a preprocessing stage (a), substitution rule evaluations (b), construction and solving of an SMT model (c).

The following sections describe the preprocessing steps, the evaluation of specified quantum circuit substitutions, then the construction of the SMT model is shown and an explicit example is given for adapting a quantum circuit generated for IBM quantum computers [IBM23b] to the semiconductor spin technology platform specified in [PRE+22, YTO+18].

8.2.1 Preprocessing

The preprocessing stage includes three successive steps. The given quantum computation is first partitioned into two-qubit blocks that only include quantum gates interacting on the same pair of qubits. The block dependency is given by a dependency graph that contains each block b as a vertex and an edge $a = (b', b)$ if block b' must be computed before block b.

In the second step, each quantum gate in the given quantum computation is directly substituted by the basis gates of the target technology platform. This basis gate substitution can be performed by using an equivalence library that can be generated manually [Qis23] or automatically [PPJ+21].

After the basis gate substitution, the cost of each block is evaluated in terms of block duration and block fidelity. The block duration quantifies the time the target quantum computer needs to execute the block, i.e. the sum of quantum gate durations along the critical path in the block. The product of each quantum gate fidelity in a block defines the block fidelity. This single pass basis gate substitution provides a trivial adaptation whose cost is used as a reference in subsequent steps.

8.2.2 Evaluation of Substitution Rules

The input quantum circuit is evaluated with respect to each specified substitution rule before the SMT model is constructed in a subsequent step. For each substitution s applicable to the given quantum circuit, the evaluation of that substitution rule yields the set of substituted gates p_s, the set of substitution gates g_s, the affected blocks b_s and the cost of the substitution w_s.

A substitution rule can be a gate equivalency included in an equivalency library via the basis gate translation approach, a specific template used in the template optimization step, or a decomposition method that decomposes a block to the basis gates of the target technology platform. These substitution rules can be defined manually by a domain expert as a set of templates or gate equivalencies [IMM+22], derived automatically [PPJ+21] for the basis gates of the target technology platform, or be part of a general decomposition method such as the KAK decomposition [NKS06].

Gate equivalency or template substitution rules can be evaluated in polynomial runtime [IMM+22]. For evaluating substitution rules based on unitary decomposition, the unitary matrix of each two-qubit block must be computed first in order to evaluate the cost of a decomposition. Despite requiring an exponential runtime in the number of qubits n for determining the unitary matrix of an n-qubit block in general, the runtime overhead is not significant for the $n = 2$ qubit blocks in this method.

8.2.3 Formal Model for Quantum Circuit Adaptation

In this section, an SMT model is derived from the data yielded by the preprocessing stage and the substitution rule evaluation. After determining a solution to the SMT model, a quantum circuit adaptation for a target technology platform is yielded. The constructed SMT model consists of Boolean or integer variables, constraints that set the relation between those variables and the definition of linear objective functions. A solution to the SMT model consists of an assignment to the variables of the developed SMT model that satisfies all model constraints and that is optimal with respect to the defined model assumptions and objective function. In this work the Z3 solver software was used to determine a solution to the SMT model [MB08].

Model Variables

Given a quantum computation with substitutions S, blocks B and a dependency graph $G = (V, A)$ with vertices V and edges A consists of variables:

- $C = \{c_1, ..., c_{|S|}\}$—this set of variables represents available substitutions for quantum circuit adaptation, i.e. the resulting adaptation only contains a substitution $s \in S$ if c_s evaluates to true.
- $E = \{e_1, ..., e_{|B|}\}$—this set of variables represents the individual starting time of each block, i.e. the time at which the computation of a block is started on the target quantum computer.
- $D = \{d_1, ..., d_{|B|}\}$—this set of variables represents the individual duration of each block, i.e. the amount of time a block requires for computation on the target quantum computer
- $F = \{f_1, ..., f_{|B|}\}$—this set of variables represents the individual fidelity of each block, i.e. the product of quantum gate fidelities in the block.

Model Constraints

The assignment to variables C, E, D, and F is constrained such that satisfying all constraints yields a valid and optimized quantum circuit adaptation. In the following, these

constraints are described in detail. In a conflict-free selection of substitutions, two selected substitutions may not replace the same set of quantum gates:

$$\neg c_s \vee \neg c_{s'} \qquad \forall s, s' \in S : p_s \cap p_{s'} \neq \emptyset, \tag{8.1}$$

where p_s and $p_{s'}$ are the sets of quantum gates that will be substituted by substitutions s and s', respectively. The symbol $\neg$ refers to the logic negation while the symbol $\wedge$ ($\vee$) corresponds to the logic conjunction (disjunction). Furthermore, performing a quantum computation must obey the block dependency defined in graph G. Thus, the block b may only be performed on a target quantum computer if the computation of any preceding block b' has been concluded:

$$e_b \geq e_{b'} + d_{b'} \qquad \forall b, b' \in B : a_{b',b} \in A, \tag{8.2}$$

where $a_{b',b}$ is a dependency graph edge that defines the dependency between blocks b' and b, $e_{b'}$ is the time step at which block b' starts to be computed and $d_{b'}$ is the time required to compute block b'. In the last set of constraints, the block duration time and block fidelity of each block are set depending on the selected substitutions. The duration d_b of a block b is set by:

$$d_b := D(b) + \sum_{s \in S'} \mathbb{D}(s) \wedge c_s, \tag{8.3}$$

where $D(\cdot)$ is a function that returns the initial duration of a block after the direct basis translation performed in the preprocessing step and that returns the duration of a quantum gate. In addition, $\mathbb{D}(\cdot)$ gives the reduction in duration that can be achieved by a substitution. The reduction in block duration is defined by

$$\mathbb{D}(s) = \sum_{g \in g_s} D(g) - \sum_{p \in p_s} D(p), \tag{8.4}$$

where g_s is the set of quantum gates introduced by a substitution s and p_s is the set of quantum gates replaced by substitution s. The fidelity f_b of a block b is set in a similar manner by:

$$f_b := \log(F(b)) + \sum_{s \in S} \mathbb{F}(s) \wedge c_s \tag{8.5}$$

where $F(\cdot)$ returns the fidelity of a quantum gate or a block as given by the direct basis translation in the preprocessing stage, and $\mathbb{F}(\cdot)$ gives the improvement in fidelity achieved by a substitution. The improvement in fidelity is defined by:

$$\mathbb{F}(s) = \sum_{g \in g_s} \log(F(g)) - \sum_{p \in p_s} \log(F(p)), \tag{8.6}$$

where g_s is again the set of quantum gates introduced by substitution s and p_s is the set of quantum gates replaced by substitution s.

Note that the SMT solver does not need to compute functions $D(\cdot)$ and $\log(F(\cdot))$, i.e. these functions are not included in the developed SMT model. Instead, the individual value of every substitution s, quantum gate g and block b in the adaptation determined by the direct basis translation performed in the preprocessing stage is computed before the SMT model is constructed. Furthermore, only one duration and start time are registered for a two-qubit block in the developed model. This may introduce ambiguities in the order of single-qubit quantum gates when the qubit idle time or quantum circuit duration is minimized and in templates or two-qubit blocks where the duration on one qubit is different to the other.

Objective Functions

The last element of the developed SMT model is objective functions that guide the selection of substitutions towards quantum circuit adaptations that improve the probability of computing the correct result on a near-term quantum computer. In this chapter, objective functions are investigated that improve the fidelity or, the qubit idle time of the adapted quantum circuit individually and or in conjunction. The qubit idle time has been selected as a metric to be reduced as idling qubits have been observed to be a significant source of error [JJAB+21]. During idle time, the state of a qubit is assumed to decay exponentially, i.e. the state of a qubit is unaffected by the idle time with probability:

$$e^{-d/T}, \tag{8.7}$$

where d is the idle time and T is the coherence time of the target quantum computer. The fidelity of a quantum circuit adaptation is maximized by:

$$\max \sum_{b} f_b, \tag{8.8}$$

where f_b is defined in Equation (8.5). Minimizing the qubit idle time in the quantum circuit adaptation is represented by the objective function:

$$\max \ -\frac{Q \cdot \mathrm{D} - \sum_b d_b}{T}, \tag{8.9}$$

where Q is the qubit number and D is the total duration of a quantum computation. The combination of these objectives is also investigated and given by:

$$\max \sum_{b} \log(f_b) - \frac{Q \cdot D - \sum_b d_b}{T}. \tag{8.10}$$

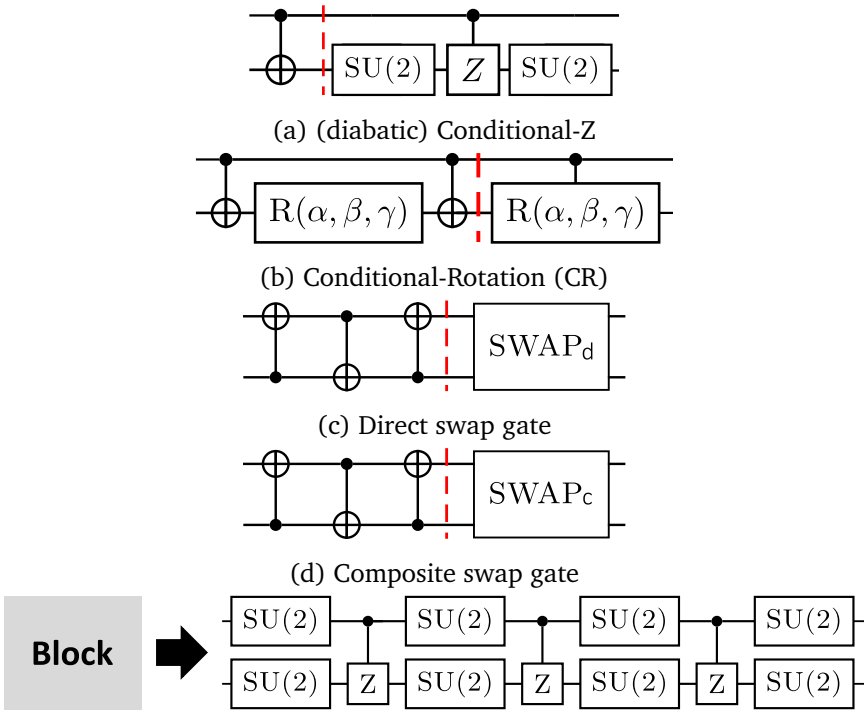
Retrieving the Adaptation from a Satisfiable Assignment

After a satisfiable assignment to the SMT model variables has been computed, a substitution $s \in S$ is included in a quantum circuit adaptation if c_s is set in the assignment, i.e. evaluates to true. The quantum circuit adaptation is generated by successively applying the substitution s to the given quantum computation by substituting quantum gates p_s in the quantum circuit with g_s in the substitution. Any quantum gate not included in the set of selected substituted quantum gates p_s is best adapted by the basis translation performed in the preprocessing step. Thus, it is not required to select a substitution $s \in S$ such that each quantum gate is in a set of substituted gates p_s as defined by the developed SMT model.

Example: Adapting from IBM quantum computers to Semiconductor Spin Qubits

In this section the adaptation of a quantum computation given in the basis of an IBM quantum computer [IBM23b] to the semiconductor spin qubit technology platform in [PRE+22]. Figure 8.3 shows the example quantum computation and Table 8.1 (D_0)

describes the quantum gate characteristics supported by the example target technology platform [PRE+22]. The example technology platform supports arbitrary single-qubit quantum gates in SU(2), a two-qubit controlled-Z (CZ) gate that is also used for KAK decompositions, two-qubit conditional rotation gates along an arbitrary axis and swap gates. In this example, two swap gate realisations $SWAP_d$ and $SWAP_c$ are considered. The $SWAP_c$ swap gate requires a larger runtime but also exhibits a higher fidelity than the $SWAP_d$ swap gate realization. The swap gate $SWAP_d$ or the swap gate $SWAP_c$ may be preferable in an adaptation depending on the structure of the quantum computation, the employed objective function and characteristics of the target quantum computer such as the decoherence time.



(a) (diabatic) Conditional-Z

(b) Conditional-Rotation (CR)

(c) Direct swap gate

(d) Composite swap gate

(e) KAK decomposition using CZ and single-qubit gates

Figure 8.2: Substitution rules for adapting quantum computation generated for IBM quantum computers [IBM23b] to a semiconductor spring technology platform [PER+20].

Figure 8.3 shows the results of the quantum circuit adaptation steps in this example. First, the preprocessing stage yields a quantum computation that is partitioned into two-qubit blocks with reference costs as given by a direct basis gate translation (see Figure 8.2a). In the next step, the applicable substitution rules described in figure 8.2 are evaluated on

the given quantum computation. The result of the evaluation is ten substitution matches,

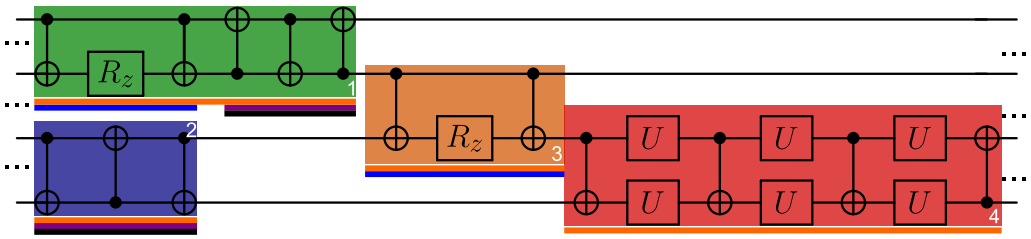


Figure 8.3: Example quantum circuit adaptation for a quantum computation in the IBM quantum computer basis gate set. Continuous lines indicate quantum gates substituted by the same substitution rule. An orange line corresponds to a KAK decomposition, a blue line to conditional rotation gates, and violet and black lines to different swap gate realizations.

where each block could be substituted using the KAK decomposition (orange line), the parts of block 1 and block 3 could be substituted by the conditional-rotation gate (blue line), and SWAP_d (violet line) as well as swap_c (black line) could each be applied once in block 1 and block 2. As an example, we set the duration of block 1 by:

$$d_1 = 965 + (573 - 965) \wedge c_0 + (660 - 422) \wedge c_1 + (19 - 543) \wedge c_2 + (67 - 543) \wedge c_3, \quad (8.11)$$

where the reference duration of block 1 is 965ns, c_0, c_1, c_2, c_3 corresponds to whether the KAK decomposition (c_0), the conditional-rotation substitution (c_1), the swap substitution SWAP_d (c_2) or SWAP_c (c_3) is applied. The characteristics of other blocks are computed in a similar way and subsequently input to the developed SMT model (see Section 8.2). As guided by the employed objective function, different substitutions may be selected during a quantum circuit adaptation. In this example, the duration of the given quantum computation should be minimized. The duration of block 1 could be reduced by 392ns using a KAK decomposition, while the duration would be increased by 238ns using the conditional-rotation quantum gate substitution. Furthermore, SWAP_d quantum gate substitution would reduce the duration by 524ns and SWAP_c substitution reduces the duration by 476ns.

The set of substitutions $\{s_0, s_2, s_3\}$ and $\{s_0, s_1\}$ are in conflict, i.e. they substitute the same set of quantum gates and can not be selected simultaneously. Therefore, the duration of block 1 is reduced most when applying the KAK substitution s_0.
Similarly for an arbitrary quantum computation, the values and constraints are input to an SMT model that is solved by an SMT solver whose result informs the selection of substitutions and thus the quantum circuit adaptation.

8.3 Evaluation

In this section, the developed rigorous quantum circuit adaptation is evaluated on the introduced semiconductor spin technology platform. The increase in quantum circuit fidelity and Hellinger fidelity, and the decrease in qubit idle time is investigated for quantum volume circuits [CBS+19] and random circuits containing quantum gates from the templates shown in Figure 8.2 with up to four qubits and a quantum circuit depth of up to 160. As given in Table 8.1, two sets of quantum gate characteristics D_0, D_1 were selected for the evaluation. The developed SMT model is compared to individually employing the quantum circuit substitution approaches in Section 4.4. Specifically, a KAK decomposition using CZ and diabatic CZ gates, template optimization with two cost functions targeting quantum circuit fidelity and qubit idle time, and a direct basis translation that replaces each non-supported two-qubit quantum gate in the given quantum computation with a supported CZ gate. The fidelity objective SAT F given in Equation (8.8), the idle time objective SAT R given in Equation (8.9), and the combined objective SAT P as given in Equation (8.10) were investigated in the evaluation. The direct basis translation is chosen as a reference for comparison in the following results. A given quantum computation was compiled by Qiskit [Qis23] to yield a quantum circuit suitable for computation on IBM backends before employing the developed rigorous quantum circuit adaptation.

8.3.1 Circuit Fidelity Increase and Qubit Idle Time Decrease

In this section, we evaluated the ability of the developed rigorous quantum circuit adaptation to decrease the qubit idle time and increase the quantum circuit fidelity as given by the product of individual quantum gate fidelities. In the following results, the fidelity

and idle time of the quantum circuit achieved by direct basis translation are chosen as a baseline. The SMT approach yields the largest improvement in quantum circuit fidelity of up to 15% as depicted in Figure 8.4. Performing quantum circuit adaptation By only using a KAK decomposition with (diabatic) CZ gates the quantum circuit fidelity decreases when performing quantum circuit adaptation additional single-qubit gates may be introduced that would be missing in e.g. template optimization. Furthermore, the decrease in quantum circuit fidelity is also caused by the utilisation of the diabatic CZ gate in the KAK decomposition that has a lower gate fidelity as the second realization of the CZ gate used in direct basis translation. Figure 8.5 shows the decrease in qubit idle time for

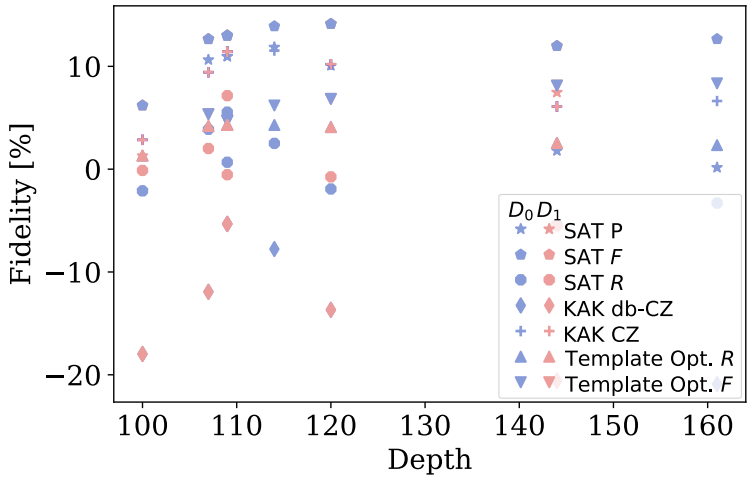


Figure 8.4: Impact of quantum circuit adaptation on the quantum circuit fidelity as given by the product of gate fidelities.

the evaluated quantum circuits as yielded by the investigated quantum circuit adaptation and substitution techniques. The developed rigorous quantum circuit adaptation method yields the highest decrease in qubit idle time for all but the smallest quantum circuit where single-qubit quantum gates may dominate the change in qubit idle time.

8.3.2 Hellinger Fidelity and Qubit Idle Time

In this section, the impact of the developed approach on the qubit idle time and the Hellinger fidelity is investigated. The Hellinger fidelity is determined by performing a simulation subject to errors incurred by a depolarization channel that corresponds to the individual gate fidelities and thermal relaxation due to qubit idle time [Qis23]. As given

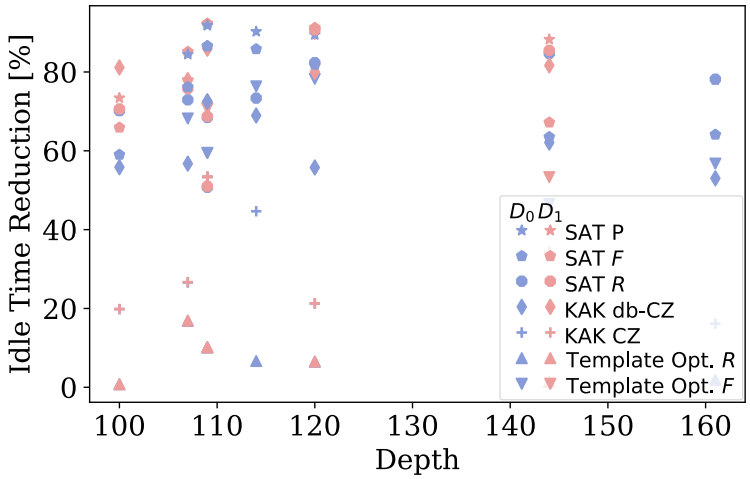


Figure 8.5: Decrease in qubit idle time yielded by the investigated quantum circuit adaptation methods.

in [PRE+22], a $T_2 = 2900$ ns decoherence and a T_1 decoherence that is three orders of magnitudes larger was used for simulation. In Figure 8.4, the decrease in qubit idle time is

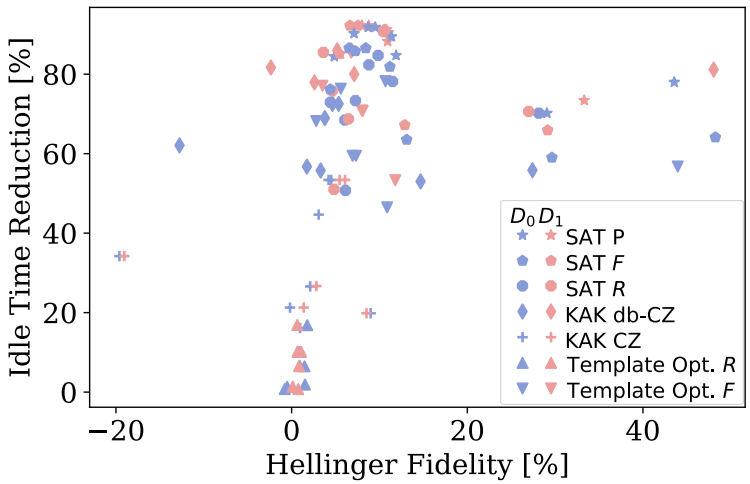


Figure 8.6: Change in Hellinger fidelity and qubit idle time yielded by the investigated quantum circuit adaptation methods.

shown on the y-axis and the change in Hellinger fidelity is given on the x-axis. The highest decrease in qubit idle time and the largest increase in Hellinger fidelity are yielded by the developed rigorous quantum circuit adaptation method. The KAK unitary decomposition and template optimization yield good results occasionally but lead to worse results than the developed rigorous approach in most cases.

8.4 Summary

In this chapter, a rigorous quantum circuit adaptation method was developed and evaluated on a technology platform with multiple universal quantum gate sets that are used in combination to yield quantum computations with a higher circuit fidelity or smaller qubit idle time. The developed rigorous compilation method was demonstrated to be well suited for dealing with multiple two-qubit gates and to yield a decrease in qubit idle time of up to 87% and an increase in Hellinger fidelity of up to 40% compared to a quantum circuit substitution based on direct basis translation.

Part III

Conclusions and Future Work

Chapter 9

Conclusions

Quantum computing holds the potential to solve certain problems with an advantage, notably within cryptography, material sciences or drug design, that are infeasible on classical computers at a large scale. However, near-term quantum computing is affected by numerous challenges that are rooted in the employed nascent quantum computing technology platforms. Thus, the scale at which relevant problems can be addressed by a near-term quantum computer is limited, restricting the ability to yield economic or scientific value.

In this work, four challenges were identified for near-term quantum computing and the methods developed in this work address all four of them as follows. The impact of limited qubit connectivity on near-term quantum computations can be reduced by the rigorous compilation methods developed in Chapters 5 to 7, leading to a reduction of quantum circuit duration by up to 45% as well as an improvement in state fidelity by up to 30% with topology shifts and up to a factor of 4.3x with qubit reuse. Furthermore, the scarcity of qubits was addressed by quantum circuit partitioning and qubit reuse, yielding a reduction in the required number of qubits of up to 5x with qubit reuse and up to 40% with quantum circuit partitioning depending on the structure of the quantum computation. Specifically for the challenge of heterogeneous operations incurred by the multitude of available quantum computing technology platforms, the rigorous compilation method in Chapter 8 improves the state fidelity by up to 40% and reduces the qubit idle time by up to 87% when translating quantum computations between technology platforms. By reducing the required number of qubits, the duration of a quantum computation, and the number of operations in a quantum computation and by considering additional sources of

errors such as qubit idle time, all of the developed rigorous compilation methods demonstrated an increase in state fidelity and thus a (partial) avoidance of ubiquitous errors in near-term devices.

The identified compilation opportunities and the developed rigorous quantum compilation methods have been shown to improve quantum computations in various ways. The result quality of quantum computations can be increased while the error rate stays unchanged on the quantum computer (or allowing a higher error rate on a quantum computer at the same result quality). The duration of an individual quantum computation can be decreased by quantum circuit partitioning, finding suitable adaptations or lowering the effort of accommodating the limited qubit connectivity of near-term quantum computers. Thus, a smaller coherence time on a quantum computer can be tolerated by applying the developed compilation methods. Furthermore, quantum computations that are infeasible due to an insufficient number of available qubits on a target quantum computer can still be computed by applying qubit reuse or quantum circuit partitioning.

The developed rigorous compilation methods are based on formal modelling that was demonstrated to be applicable to near-term quantum computations with up to 40 qubits and a quantum circuit depth of up to 160 quantum gates in this work. However, determining a satisfiable assignment to a formal model and optimizing over such assignments incurs an exponential runtime overhead for the solving procedures in general, limiting the scope of rigorous compilation methods.

A key to the scalability of the solving procedures to problems of relevant sizes is an appropriate preparation of the formal model including the following aspects. First, determining a suitable level of abstraction for the considered compilation opportunity and quantum circuit structure as well as an efficient formulation of the resulting formal model is imperative. For instance in Chapter 7, the exact qubit assignment within a partition is omitted to reduce the solving effort while the typical quantum circuit graph structure is extended to allow an accurate representation of the quantum circuit partitioning problem. Semantically equivalent formal models can exhibit a significant difference in solving procedure runtime depending on their exact formulation, justifying the exploration of constraint formulations such as in [Sin05] for Boolean cardinality constraints.

Furthermore, the employed optimization can have a large impact on the solving procedure runtime. For certain objective functions, an optimization can be completely replaced by successively solving larger problem instances. This resulted in a lower solving procedure runtime for the rigorous compilation methods in Chapters 6 and 7. For other objective functions concerning multiple metrics, a larger solving procedure runtime was observed when these metrics were combined into one objective function as a weighted sum as in Chapter 8 instead of successively optimizing each metric as in Chapters 5 and 7. Finally, the quantum circuit characteristics considered by the optimization routine can often be quantified on different levels of detail. For instance, replacing exact (Chapter 8) by uniform quantum gate durations resulted in significant simplifications and solving procedure runtime in Chapters 5 to 7.

In summary, the computational reach of near-term quantum computers was demonstrated to be extended by identifying suitable compilation opportunities and incorporating them into rigorous compilation methods. Specifically, an improvement in duration, number of operations, qubit requirement and incurred errors of near-term quantum computations was shown. Furthermore, the ability of rigorous compilation methods to find the global optimum enables the investigation into the incorporated compilation opportunities themselves. In Chapter 6, this investigation yielded suitable technology parameters for the incorporated competing compilation opportunities that informed the development of future neutral-atom quantum computers. While even with an increased computational reach, a runtime speedup was not demonstrated by using quantum computers over their best classical counterparts, rigorous compilation methods still enable the exploration of quantum computations that are otherwise infeasible in the near term. Thus, possibly paving the way to experimental demonstrations on near-term quantum computers that are solving practical problems with a speedup or with other practical advantages.

Ongoing and Future Research

Error characterization experiments on near-term quantum computers continue to unearth novel error mechanisms on quantum computers [KW23, JJAB$^+$21]. While considering a simple error model is typically sufficient for error analysis [BDP21, AAB$^+$19] and quan-

tum error correction [FMMC12], identifying compilation opportunities for novel error mechanisms and subsequently modelling them for incorporation into rigorous compilation methods can have a tremendous impact on the result quality [MMMJA20, BKNB23].
In traditional computer systems, suitable levels of abstraction have been determined for the individual steps necessary to realize computations with a potentially abstract specification. In quantum computing, however, these layers of abstraction are still being explored with different tradeoffs available for different classes of quantum computations. For instance, in permutation-aware synthesis [LYW+23] or for quantum computations entirely represented by phase-polynomials, the synthesis step can consider circuit mapping (or: routing) step [MvdGD23, NGM20, VMB22] to yield quantum computations that incur less errors. Here, rigorous compilation can help to analyze suitable abstraction layers for different classes of quantum computations accurately by evaluating different abstractions during rigorous compilation in conjunction.
Furthermore, large quantum computing systems are expected to become viable in the near future using a modular approach [IBM23c], i.e. separate quantum computers are connected via a low number of classical or quantum links. Rigorous compilation methods can help to utilize these links efficiently for enabling the resolution of a problem instance that would not be resolvable on any of the separate quantum computers.
The expected increase in quantum computer size will limit the scope of rigorous compilation methods. The scope of rigorous compilation may be extended in the immediate future by developing more efficient models or solver strategies. A further increase in quantum computer size to thousands of qubits will, however, require the development of heuristic or hybrid approaches for the compilation of quantum computations that consider full quantum error correction schemes in the long term.

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the 43rd Annual ACM symposium on Theory of Computing (STOC'11)*, pages 333–342, 2011. doi:10.1364/qim.2014.qth1a.2.

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

[Aar15] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015. doi:10.1038/NPhys3272.

[ABDK+21] Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and Quantum implications of Huang's sensitivity theorem. In *Proceedings of the 53rd ACM SIGACT Annual Symposium on Theory of Computing (STOC'21)*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

[AGDLHG05] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005. doi:10.1126/Science.1113479.

[AL99] Daniel S. Abrams and Seth Lloyd. Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors. *Physical Review Letters*, 83(24):5162–5165, 1999. doi:10.1103/PhysRevLett.83.5162.

[ALRS+20] Thomas Ayral, François-Marie Le Régent, Zain Saleem, Yuri Alexeev, and Martin Suchara. Quantum divide and compute: Hardware demonstrations and noisy simulations. In *Proceedings of the 12th IEEE Computer Society Annual Symposium on VLSI (ISVLSI'20)*, pages 138–140, 2020. doi:10.1109/ISVLSI49217.2020.00034.

[AMOL22] Philipp Aumann, Tim Menke, William D Oliver, and Wolfgang Lechner. Circuitq: an open-source toolbox for superconducting circuits. *New Journal of Physics*, 24(9):093012, 2022. doi:10.1088/1367-2630/ac8cab.

[ARS+21] Thomas Ayral, François-Marie Le Régent, Zain Saleem, Yuri Alexeev, and Martin Suchara. Quantum divide and compute: exploring the effect of different noise sources. *SN Computer Science*, 2(3):132, 2021. doi:10.1007/s42979-021-00508-9.

[ASWD13] Nabila Abdessaied, Mathias Soeken, Robert Wille, and Rolf Drechsler. Exact template matching using boolean satisfiability. In *Proceedings of the 43rd IEEE International Symposium on Multiple-Valued Logic (ISMVL'13)*, pages 328–333, 2013. doi:10.1109/ISMVL.2013.26.

[B+18] Panagiotis Kl. Barkoutsos et al. Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. *Physical Review A*, 98:022322, 2018. doi:10.1103/PhysRevA.98.022322.

[Bar15] Stefanie Barz. Quantum computing with photons: Introduction to the circuit model, the one-way quantum computer, and the fundamental principles of photonic experiments. *Phys B-At Mol Opt*, 48(8):083001, 2015. doi:10.1088/0953-4075/48/8/083001.

[BB02] Jean-Luc Brylinski and Ranee Brylinski. Universal quantum gates. *Mathematics of quantum computation*, 79, 2002. doi:10.1201/9781420035377.pt2.

[BBD+09] Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009. doi:10.1038/NPhys1157.

[BBD+23] Sebastian Brandhofer, Daniel Braun, Vanessa Dehn, Gerhard Hellstern, Matthias Hüls, Yanjun Ji, Ilia Polian, Amandeep Singh Bhatia, and Thomas Wellens. Benchmarking the performance of portfolio optimization with QAOA. *Quantum Information Processing*, 22(1):1–27, 2023. doi:10.1007/s11128-022-03766-5.

[BBMKLC20] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020. doi:10.1021/ACS.ChemRev.9b00829.

[BBP21] Sebastian Brandhofer, Hans Peter Büchler, and Ilia Polian. Optimal mapping for near-term quantum architectures based on Rydberg atoms. In *Proceedings of the 40th IEEE/ACM International Conference On Computer-Aided Design (ICCAD'21)*, 2021. doi:10.1109/ICCAD51958.2021.9643490.

[BC23] Tomislav Begušić and Garnet Kin-Lic Chan. Fast classical simulation of evidence for the utility of quantum computing before fault tolerance, 2023. doi:10.48550/arXiv.2306.16372.

[BDD+02] Michael J. Bremner, Christopher M. Dawson, Jennifer L. Dodd, Alexei Gilchrist, Aram W. Harrow, Duncan Mortimer, Michael A. Nielsen, and Tobias J. Osborne. Practical scheme for quantum computation with any two-qubit entangling gate. *Physical Review Letters*, 89:247902, 2002. doi:10.1103/PhysRevLett.89.247902.

[BDG+22] Sergey Bravyi, Oliver Dial, Jay M Gambetta, Dario Gil, and Zaira Nazario. The future of quantum computing with superconducting qubits. *Journal of Applied Physics*, 132(16):160902, 2022. doi:10.1063/5.0082975.

[BDLL+16] Daniel Barredo, Sylvain De Léséleuc, Vincent Lienhard, Thierry Lahaye, and Antoine Browaeys. An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays. *Science*, 354(6315):1021–1023, 2016. doi:10.1126/Science.aah3778.

[BDP21] Sebastian Brandhofer, Simon Devitt, and Ilia Polian. ArsoNISQ: Analyzing Quantum Algorithms on Near-Term Architectures. In *Proceedings of the 26th IEEE European Test Symposium (ETS'21)*, pages 1–6, 2021. doi:10.1109/ETS50041.2021.9465414.

[BDWP21] Sebastian Brandhofer, Simon Devitt, Thomas Wellens, and Ilia Polian. Noisy intermediate-scale quantum (NISQ) computers: How they work, how they fail, how to test them? In *Proceedings of the 39th IEEE VLSI Test Symposium (VTS'21)*, 2021. doi:10.1109/VTS50974.2021.9441047.

[BF28] M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180, 1928. doi:10.1007/BF01343193.

[BFP23] Armin Biere, Mathias Fleury, and Florian Pollitt. CaDiCaL vivinst, IsaSAT, Gimsatul, Kissat, and TabularaSAT entering the SAT competition 2023. *SAT Competition*, page 14, 2023.

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018. doi:10.1126/Science.aar3106.

[BHvM09] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009. ISBN: 978-1-586-03929-5.

[Bjø11] Nikolaj Bjørner. Engineering theories with z3. In *Proceedings of the 9th Asian Symposium on Programming Languages and Systems (ASPLAS'11)*, pages 4–16, 2011. doi:10.1007/978-3-642-25318-8.

[BKM07] Manfred Broy, Ingolf H. Krüger, and Michael Meisinger. A formal model of services. *ACM Transactions on Software Engineering and Methodology*, 16(1):5–es, 2007. doi:10.1145/1189748.1189753.

[BKM16] Kenneth R Brown, Jungsang Kim, and Christopher Monroe. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Information*, 2(1):1–10, 2016. doi:10.1038/npjQI.2016.34.

[BKNB23] Sebastian Brandhofer, Jinwoong Kim, Siyuan Niu, and Nicholas T. Bronn. SAT-Based Quantum Circuit Adaptation. In *Proceedings of the ACM/IEEE Conference on Design, Automation Test in Europe (DATE'23)*, 2023. doi:10.23919/DATE56975.2023.10137140.

[BL17] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017. doi:10.1038/Nature23461.

[BL20] Antoine Browaeys and Thierry Lahaye. Many-body physics with individually controlled Rydberg atoms. *Nature Physics*, 16(2):132–142, 2020. doi:10.1038/s41567-019-0733-z.

[BN20] Nikolaj Bjørner and Lev Nachmanson. Navigating the universe of Z3 theory solvers. In *Proceedings of the 23rd Brazilian Symposium on Formal Methods: Foundations and Applications (SBMF'20)*, pages 8–24, 2020. doi:10.1007/978-3-030-63882-5.

[BPF15] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. νZ - An Optimizing SMT Solver. In *Proceedings of the 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'15)*, pages 194–199, 2015. doi:10.1007/978-3-662-46681-0.

[BPK23a] Sebastian Brandhofer, Ilia Polian, and Kevin Krsulich. Optimal partitioning of quantum circuits using gate cuts and wire cuts. *IEEE Transactions on Quantum Engineering*, 2023. doi:10.1109/TQE.2023.3347106.

[BPK23b] Sebastian Brandhofer, Ilia Polian, and Kevin Krsulich. Optimal qubit reuse for near-term quantum computers. In *Proceedings of the 4th IEEE International Conference on Quantum Computing and Engineering (QCE'23)*, 2023. doi:10.1109/QCE57702.2023.00100.

[BPS23] Lukas Brenner, Christophe Piveteau, and David Sutter. Optimal wire cutting with classical communication, 2023. doi:10.48550/arXiv.2302.03366.

[Bri17] Stephen Brierley. Efficient implementation of quantum circuits with limited qubit interactions. *Quantum Info. Comput.*, 17(13–14):1096–1104, 2017. doi:10.26421/QIC17.13-14-2.

[BSK+17] Hannes Bernien, Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551(7682):579–584, 2017. doi:10.1038/Nature24622.

[BSS16] Sergey Bravyi, Graeme Smith, and John A. Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6:021043, 2016. doi:10.1103/PhysRevX.6.021043.

[BSW15] Martin Bromberger, Thomas Sturm, and Christoph Weidenbach. Linear integer arithmetic revisited. In *Proceedings of the 25th International Conference on Automated Deduction Automated Deduction (CADE'15)*, pages 623–637, 2015. doi:10.1007/978-3-319-21401-6.

[BTW+23] Elisa Bäumer, Vinay Tripathi, Derek S. Wang, Patrick Rall, Edward H. Chen, Swarnadeep Majumder, Alireza Seif, and Zlatko K. Minev. Efficient long-range entanglement using dynamic circuits, 2023. doi:10.48550/arXiv.2308.13065.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/s0097539796300921.

[BvL05] Samuel L. Braunstein and Peter van Loock. Quantum information with continuous variables. *Reviews of Modern Physics*, 77:513–577, 2005. doi:10.1103/RevModPhys.77.513.

[BW20a] Lukas Burgholzer and Robert Wille. Improved DD-based Equivalence Checking of Quantum Circuits. In *Proceedings of the 25th Asia and South Pacific Design Automation Conference (ASP-DAC'20)*, pages 127–132, 2020. doi:10.1109/ASP-DAC47756.2020.9045153.

[BW20b] Lukas Burgholzer and Robert Wille. The Power of Simulation for Equivalence Checking in Quantum Computing. In *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC'20)*, pages 1–6, 2020. doi:10.1109/DAC18072.2020.9218563.

[CBS+19] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019. doi:10.1103/PhysRevA.100.032328.

[CDD+19] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah. On the qubit routing problem, 2019. doi:10.48550/arXiv.1902.08091.

[CDM02] J.W. Cangussu, R.A. DeCarlo, and A.P. Mathur. A formal model of the software test process. *IEEE Transactions on Software Engineering*, 28(8):782–796, 2002. doi:10.1109/TSE.2002.1027800.

[CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022. ISBN: 978-0-262-04630-5.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM symposium on Theory of computing (STOC'71)*, pages 151–158, 1971. doi:10.1145/800157.805047.

[COP94] D. COPPERSMITH. An approximate fourier transform useful in quantum factoring. *IBM Research Report*, pages RC–19642, 1994. doi:10.48550/arXiv.quant-ph/0201067.

[CRAG18] Y. Cao, J. Romero, and A. Aspuru-Guzik. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, 62(6):6:1–6:20, 2018. doi:10.1147/JRD.2018.2888987.

[CTI+21] Antonio D Córcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K Minev, Jerry M Chow, and Jay M Gambetta. Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits. *Physical Review Letters*, 127(10):100501, 2021. doi:10.1103/PhysRevLett.127.100501.

[CTV17] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 2017. doi:10.1038/Nature23460.

[CZY+20] Christopher Chamberland, Guanyu Zhu, Theodore J. Yoder, Jared B. Hertzberg, and Andrew W. Cross. Topological and subsystem codes on low-degree graphs with flag qubits. *Physical Review X*, 2020. doi:10.1103/PhysRevX.10.011022.

[DBE95] David Elieser Deutsch, Adriano Barenco, and Artur Ekert. Universality in quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 449(1937):669–677, 1995. doi:10.1098/RSPA.1995.0065.

[DC08] Arnab Das and Bikas K. Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061–1081, 2008. doi:10.1103/RevModPhys.80.1061.

[DCKFF23] Matthew DeCross, Eli Chertkov, Megan Kohagen, and Michael Foss-Feig. Qubit-reuse compilation with mid-circuit measurement and reset. *Physical Review X*, 13(4):041057, 2023. doi:10.1103/PhysRevX.13.041057.

[DiV00] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000. doi:https://doi.org/fnmhpz.

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962. doi:10.1145/368273.368557.

[DLSP22] Rohit Dilip, Yu-Jie Liu, Adam Smith, and Frank Pollmann. Data compression for quantum machine learning. *Physical Review Research*, 4(4):043007, 2022. doi:10.1103/PhysRevResearch.4.043007.

[dMB08] Leonardo de Moura and Nikolaj Bjørner. Model-based theory combination. *Electronic Notes in Theoretical Computer Science*, 198(2):37–49, 2008. doi:10.1016/j.entcs.2008.04.079.

[DMD+13] Marcus W. Doherty, Neil B. Manson, Paul Delaney, Fedor Jelezko, Jörg Wrachtrup, and Lloyd C. L. Hollenberg. The nitrogen-vacancy colour centre in diamond. *Physics Reports*, 528(1):1–45, 2013. doi:10.1016/J.PhysRep.2013.02.001.

[DMN13] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013. doi:10.1088/0034-4885/76/7/076001.

[DSB+22] Dhruv Devulapalli, Eddie Schoute, Aniruddha Bapat, Andrew M Childs, and Alexey V Gorshkov. Quantum routing with teleportation, 2022. doi:10.48550/arXiv.2204.04185.

[DSC19] Nike Dattani, Szilard Szalay, and Nick Chancellor. Pegasus: The second connectivity graph for large-scale quantum annealing hardware, 2019. doi:10.48550/arXiv.1901.07636.

[DST+20] Marc G. Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. Towards Optimal Topology Aware Quantum Circuit Synthesis. In *Proceedings of the 1st IEEE International Conference on Quantum Computing and Engineering (QCE'20)*, pages 223–234, 2020. doi:10.1109/QCE49297.2020.00036.

[DTDQ21] Poulami Das, Swamit Tannu, Siddharth Dangwal, and Moinuddin Qureshi. ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling. In *Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'21)*, pages 950–962, 2021. doi:10.1145/3466752.3480059.

[EBK+16] Manuel Endres, Hannes Bernien, Alexander Keesling, Harry Levine, Eric R Anschuetz, Alexandre Krajenbrink, Crystal Senko, Vladan Vuletic, Markus Greiner, and Mikhail D Lukin. Atom-by-atom assembly of defect-free one-dimensional cold atom arrays. *Science*, 354(6315):1024–1027, 2016. doi:10.1126/Science.aah3752.

[EBL18] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical Quantum Error Mitigation for Near-Future Applications. *Physical Review X*, 8(3):031027, 2018. doi:10.1103/PhysRevX.8.031027.

[EMG+22] Andrew Eddins, Mario Motta, Tanvi P. Gujarati, Sergey Bravyi, Antonio Mezzacapo, Charles Hadfield, and Sarah Sheldon. Doubling the size of quantum simulators by entanglement forging. *PRX Quantum*, 3:010309, 2022. doi:10.1103/PrXQuantum.3.010309.

[End01] Herbert B Enderton. *A mathematical introduction to logic*. Elsevier, 2001. ISBN: 978-0-122-38452-3.

[ETE21] Nathan Earnest, Caroline Tornow, and Daniel J Egger. Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware. *Physical Rev R*, 3(4):043088, 2021. doi:10.1103/physrevresearch.3.043088.

[EWG+18] Daniel J Egger, Max Werninghaus, Marc Ganzhorn, Gian Salis, Andreas Fuhrer, Peter Müller, and Stefan Filipp. Pulsed reset protocol for fixed-frequency superconducting qubits. *Physical Review Applied*, 10(4):044030, 2018. doi:10.1103/PhysRevApplied.10.044030.

[EWL+21] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866):227–232, 2021. doi:10.1038/s41586-021-03582-4.

[EWP+19] Alexander Erhard, Joel J Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban A Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer Blatt. Characterizing large-scale quantum computers via cycle benchmarking. *Nature communications*, 10(1):5347, 2019. doi:10.1038/s41467-019-13068-7.

[F+18] Richard P Feynman et al. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 2018. doi:10.1007/BF02650179.

[Fan07] Yale Fan. A generalization of the deutsch-jozsa algorithm to multi-valued quantum logic. In *Proceedings of the 37th IEEE International Symposium on Multiple-Valued Logic (ISMVL'07)*, page 12, 2007. doi:10.1109/ISMVL.2007.3.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. doi:10.48550/arXiv.1411.4028.

[FMMC12] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012. doi:10.1103/PhysRevA.86.032324.

[FotUN17] Food and Agriculture Organization of the United Nations. World fertilizer trends and outlook to 2020, 2017.

[FSG09] Austin G. Fowler, Ashley M. Stephens, and Peter Groszkowski. High-threshold universal quantum computation on the surface code. *Physical Review A*, 80(5):052312, 2009. doi:10.1103/PhysRevA.80.052312.

[fSidIB20] Bundesamt für Sicherheit in der Informationstechnik (BSI). Cryptographic mechanisms: Recommendations and key lengths. *BSI—Technical Guideline*, 2020.

[GBD+19] Pranav Gokhale, Jonathan M. Baker, Casey Duckering, Natalie C. Brown, Kenneth R. Brown, and Frederic T. Chong. Asymptotic improvements to quantum circuits via qutrits. In *Proceedings of the 46th International Symposium on Computer Architecture (ISCA'19)*, pages 554–566, 2019. doi:10.1145/3307650.3322253.

[GE21] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 2021. doi:10.22331/q-2021-04-15-433.

[GHZ89] Daniel M Greenberger, Michael A Horne, and Anton Zeilinger. Going beyond Bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[Gid21] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, 2021. doi:10.22331/q-2021-07-06-497.

[GJW+23] L C G Govia, P Jurcevic, C J Wood, N Kanazawa, S T Merkel, and D C McKay. A randomized benchmarking suite for mid-circuit measurements. *New Journal of Physics*, 25(12):123016, 2023. doi:10.1088/1367-2630/ad0e19.

[GKG+19] T. M. Graham, M. Kwon, B. Grinkemeyer, Z. Marra, X. Jiang, M. T. Lichtman, Y. Sun, M. Ebert, and M. Saffman. Rydberg-mediated entanglement in a two-dimensional neutral atom qubit array. *Physical Review Letters*, 123:230501, 2019. doi:10.1103/PhysRevLett.123.230501.

[GMT+12] John P Gaebler, Adam M Meier, Ting Rei Tan, Ryan Bowler, Yiheng Lin, David Hanneke, John D Jost, JP Home, Emanuel Knill, Dietrich Leibfried, et al. Randomized benchmarking of multiqubit gates. *Physical review letters*, 108(26):260503, 2012. doi:10.1103/PhysRevLett.108.260503.

[GMW+09] Alpha Gaëtan, Yevhen Miroshnychenko, Tatjana Wilk, Amodsen Chotia, Matthieu Viteau, Daniel Comparat, Pierre Pillet, Antoine Browaeys, and Philippe Grangier. Observation of collective excitation of two individual atoms in the Rydberg blockade regime. *Nature Physics*, 5(2):115–118, 2009. doi:10.1038/NPhys1183.

[Goo22] Google Quantum AI. Quantum Computer Datasheet. https://quantumai.google/hardware/datasheet/weber.pdf, 2022.

[Goo23] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, 2023. doi:10.1038/s41586-022-05434-1.

[Gro97] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, 1997. doi:10.1103/PhysRevLett.79.325.

[GZ17] Weiwei Gong and Xu Zhou. A survey of SAT solver. *AIP Conference Proceedings*, 1836(1):020059, 2017. doi:10.1063/1.4981999.

[Hel09] E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909. doi:10.1515/crll.1909.136.210.

[HJC+22] Fei Hua, Yuwei Jin, Yanhao Chen, Suhas Vittal, Kevin Krsulich, Lev S. Bishop, John Lapeyre, Ali Javadi-Abhari, and Eddy Z. Zhang. Exploiting qubit reuse through mid-circuit measurement and reset, 2022. doi:10.48550/arXiv.2211.01925.

[HJG+20] Adam Holmes, Sonika Johri, Gian Giacomo Guerreschi, James S. Clarke, and A. Y. Matsuura. Impact of qubit connectivity on quantum algorithm performance. *Quantum Science and Technology*, 5(2):025009, 2020. doi:10.1088/2058-9565/ab73e0.

[Hun95] Robert G Hunsperger. *Integrated optics*. Springer, 1995. ISBN: 978-3-662-03159-9.

[HWY23] Hiroyuki Harada, Kaito Wada, and Naoki Yamamoto. Doubly optimal parallel wire cutting without ancilla qubits, 2023. doi:10.48550/arXiv.2303.07340.

[HZW21] Stefan Hillmich, Alwin Zulehner, and Robert Wille. Exploiting quantum teleportation in quantum circuit mapping. In *Proceedings of the 26th ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC'21)*, pages 792–797, 2021. doi:10.1145/3394885.3431604.

[IBM21] IBM Quantum. The IBM Quantum heavy hex lattice, 2021.

[IBM22] IBM Quantum. Bringing the full power of dynamic circuits to qiskit runtime. https://research.ibm.com/blog/quantum-dynamic-circuits, 2022.

[IBM23a] IBM Quantum. About calibration jobs. https://docs.quantum-computing.ibm.com/admin/calibration-jobs, 2023.

[IBM23b] IBM Quantum. IBM Quantum Experience. https://quantum-computing.ibm.com/, 2023.

[IBM23c] IBM Quantum. IBM Quantum Roadmap. https://ibm.com/roadmaps/quantum/, 2023.

[IBM23d] IBM Quantum. *IBM Quantum Systems*, Accessed 30th December, 2023.

[IMM+22] Raban Iten, Romain Moyard, Tony Metger, David Sutter, and Stefan Woerner. Exact and practical pattern matching for quantum circuit optimization. *ACM Transactions on Quantum Computing*, 3(1), 2022. doi:10.1145/3498325.

[ION23] IONQ. *IONQ Quantum Systems*, Accessed 30th December, 2023.

[JBP22] Yanjun Ji, Sebastian Brandhofer, and Ilia Polian. Calibration-aware transpilation for variational quantum optimization. In *Proceedings of the 3rd IEEE International Conference on Quantum Computing and Engineering (QCE'22)*, pages 204–214, 2022. doi:10.1109/QCE53715.2022.00040.

[JCZ+00] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin. Fast quantum gates for neutral atoms. *Physical Review Letters*, 85:2208–2211, 2000. doi:10.1103/PhysRevLett.85.2208.

[JDM13] Dejan Jovanović and Leonardo De Moura. Cutting to the chase: Solving linear integer arithmetic. *Journal of automated reasoning*, 51:79–108, 2013. doi:10.1007/s10817-013-9281-x.

[JJAB+21] Petar Jurcevic, Ali Javadi-Abhari, Lev S Bishop, Isaac Lauer, Daniela F Bogorin, Markus Brink, Lauren Capelluto, Oktay Günlük, Toshinari Itoko, Naoki Kanazawa, et al. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Science and Technology*, 6(2):025020, 2021. doi:10.1088/2058-9565/abe519.

[Kat50] Tosio Kato. On the Adiabatic Theorem of Quantum Mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950. doi:10.1143/JPSJ.5.435.

[KEA+23] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023. doi:10.1038/s41586-023-06096-3.

[KEBH+23] Naoki Kanazawa, Daniel J. Egger, Yael Ben-Haim, Helena Zhang, William E. Shanks, Gadi Aleksandrowicz, and Christopher J. Wood. Qiskit Experiments: A Python package to characterizeand calibrate quantum computers. *Journal of Open Source Software*, 8(84):5329, 2023. doi:10.21105/joss.05329.

[Kit03] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003. doi:10.1016/S0003-4916(02)00018-0.

[KLR+08] Emanuel Knill, Dietrich Leibfried, Rolf Reichle, Joe Britton, R Brad Blakestad, John D Jost, Chris Langer, Roee Ozeri, Signe Seidelin, and David J Wineland. Randomized benchmarking of quantum gates. *Physical Review A*, 77(1):012307, 2008. doi:10.1103/PhysRevA.77.012307.

[KLZ98] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation. *Science*, 279(5349):342–345, 1998. doi:10.1126/Science.279.5349.342.

[KMN+07] Pieter Kok, William J Munro, Kae Nemoto, Timothy C Ralph, Jonathan P Dowling, and Gerard J Milburn. Linear optical quantum computing with photonic qubits. *Reviews of modern physics*, 79(1):135, 2007. doi:10.1103/revmodphys.79.135.

[KMT+17] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017. doi:10.1038/Nature23879.

[KMW21] Oliver Keszocze, Naser Mohammadzadeh, and Robert Wille. Exact Physical Design of Quantum Circuits for Ion-Trap-based Quantum Architectures. In *Proceedings of the ACM/IEEEDesign, Automation & Test in Europe Conference & Exhibition (DATE'21)*, pages 344–349, 2021. doi:10.23919/DATE51398.2021.9474188.

[KTC+19] Abhinav Kandala, Kristan Temme, Antonio D. Córcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019. doi:10.1038/s41586-019-1040-7.

[KW23] Andreas Ketterer and Thomas Wellens. Characterizing crosstalk of superconducting transmon processors. *Physical Review Applied*, 20(3):034065, 2023. doi:10.1103/PhysRevApplied.20.034065.

[LAG22] Junde Li, Mahabubul Alam, and Swaroop Ghosh. Large-scale quantum approximate optimization via divide-and-conquer. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1–1, 2022. doi:10.1109/TCAD.2022.3212196.

[Lan81] Carl E. Landwehr. Formal Models for Computer Security. *ACM Computing Surveys*, 13(3):247–278, 1981. doi:10.1145/356850.356852.

[LB20] Frank Leymann and Johanna Barzen. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology*, 5(4):044007, 2020. doi:10.1088/2058-9565/abae7d.

[LDX19] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the qubit mapping problem for nisq-era quantum devices. In *Proceedings of the 24th ACM/IEEE International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*, pages 1001–1014, 2019. doi:10.1145/3297858.3304023.

[LG22] Zhenning Liu and Alexandru Gheorghiu. Depth-efficient proofs of quantumness. *Quantum*, 6:807, 2022. doi:10.22331/q-2022-09-19-807.

[LKS+19] Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Tout T. Wang, Sepehr Ebadi, Hannes Bernien, Markus Greiner, Vladan Vuletić, Hannes Pichler, and Mikhail D. Lukin. Parallel implementation of high-fidelity multiqubit gates with neutral atoms. *Physical Review Letters*, 123:170503, 2019. doi:10.1103/PhysRevLett.123.170503.

[LKT+23] Wan-Hsuan Lin, Jason Kimko, Bochen Tan, Nikolaj Bjørner, and Jason Cong. Scalable optimal layout synthesis for nisq quantum processors. In *Proceedings of the 60th ACM/IEEE Design Automation Conference (DAC'23)*, pages 1–6, 2023. doi:10.1109/DAC56929.2023.10247760.

[LMH+23] Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J. O'Riordan, Thomas R. Bromley, Juan Miguel Arrazola, and Nathan Killoran. Fast quantum circuit cutting with randomized measurements. *Quantum*, 7:934, 2023. doi:10.22331/q-2023-03-02-934.

[LPMMS16] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016. doi:10.1007/s10601-015-9183-0.

[LWF+17] Bjoern Lekitsch, Sebastian Weidt, Austin G. Fowler, Klaus Mølmer, Simon J. Devitt, Christof Wunderlich, and Winfried K. Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3(2), 2017. doi:10.1126/sciadv.1601540.

[LYW+23] Ji Liu, Ed Younis, Mathias Weiden, Paul Hovland, John Kubiatowicz, and Costin Iancu. Tackling the qubit mapping problem with permutation-aware synthesis. In *Proceedings of the 4th IEEE International Conference on Quantum Computing and Engineering (QCE'23)*, 2023. doi:10.1109/QCE57702.2023.00090.

[MAB+18] David C. McKay, Thomas Alexander, Luciano Bello, Michael J. Biercuk, Lev Bishop, Jiayin Chen, Jerry M. Chow, Antonio D. Córcoles, Daniel Egger, Stefan Filipp, Juan Gomez, Michael Hush, Ali Javadi-Abhari, Diego Moreda, Paul Nation, Brent Paulovicks, Erick Winston, Christopher J. Wood, James Wootton, and Jay M. Gambetta. Qiskit backend specifications for OpenQASM and OpenPulse experiments, 2018. doi:10.48550/arXiv.1809.03452.

[MAdCB20] Swarnadeep Majumder, Leonardo Andreta de Castro, and Kenneth R. Brown. Real-time calibration with spectator qubits. *npj Quantum Information*, 6(1):1–9, 2020. doi:10.1038/s41534-020-0251-y.

[Man80] Yuri Manin. Computable and uncomputable. *Sovetskoye Radio, Moscow*, 128:28, 1980.

[Mar21] John M. Martinis. Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays. *npj Quantum Information*, 7(1):1–9, 2021. doi:10.1038/s41534-021-00431-0.

[Mas21] Dmitri Maslov. Reversible logic synthesis benchmarks page. *http://www.cs.uvic.ca/maslov/*, 2021.

[MB08] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08)*, pages 337–340, 2008. doi:10.1007/978-3-540-78800-3.

[MCS+20] Ivaylo S Madjarov, Jacob P Covey, Adam L Shaw, Joonhee Choi, Anant Kale, Alexandre Cooper, Hannes Pichler, Vladimir Schkolnik, Jason R Williams, and Manuel Endres. High-fidelity entanglement and detection of alkaline-earth Rydberg atoms. *Nature Physics*, 16(8):857–861, 2020. doi:10.1038/s41567-020-0903-z.

[MDS22] Allen Mi, Shuwen Deng, and Jakub Szefer. Securing reset operations in NISQ quantum computers. In *Proceedings of the 29th ACM SIGSAC Conference on Computer and Communications Security (CCS'22)*, pages 2279–2293, 2022. doi:10.1145/3548606.3559380.

[MF21a] Kosuke Mitarai and Keisuke Fujii. Constructing a virtual two-qubit gate by sampling single-qubit operations. *New Journal of Physics*, 23(2):023021, 2021. doi:10.1088/1367-2630/abd7bc.

[MF21b] Kosuke Mitarai and Keisuke Fujii. Overhead for simulating a non-local channel with local channels by quasiprobability sampling. *Quantum*, 5:388, 2021. doi:10.22331/q-2021-01-28-388.

[MGE12] Easwar Magesan, Jay M Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Physical Review A*, 85(4):042311, 2012. doi:10.1103/PhysRevA.85.042311.

[MHL+13] Antonio Morgado, Federico Heras, Mark Liffiton, Jordi Planes, and Joao Marques-Silva. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013. doi:10.1007/s10601-013-9146-2.

[MHW23] Kevin Mato, Stefan Hillmich, and Robert Wille. Compression of Qubit Circuits: Mapping to Mixed-Dimensional Quantum Systems. In *Proceedings of the 2nd IEEE*

International Conference on Quantum Software (QSW'23), pages 155–161, 2023. doi:10.1109/QSW59989.2023.00027.

[Mil07] Paolo Milazzo. *Qualitative and quantitative formal modeling of biological systems*. PhD thesis, Ph. D. Thesis, University of Pisa, 2007.

[MMD03] D Michael Miller, Dmitri Maslov, and Gerhard W Dueck. A transformation based algorithm for reversible logic synthesis. In *Proceedings of the 40th ACM/IEEE Design Automation Conference (DAC'03)*, pages 318–323, 2003. doi:10.1145/775832.775915.

[MMD+21] Zlatko K Minev, Thomas G McConkey, Jeremy Drysdale, Priti Shah, Dennis Wang, Marco Facchini, Grace Harper, John Blair, Helena Zhang, Nick Lanzillo, Sagarika Mukesh, Will Shanks, Chris Warren, and Jay M Gambetta. Qiskit Metal: An Open-Source Framework for Quantum Device Design & Analysis, 2021. doi:10.5281/zenodo.4618153.

[MMMJA20] Prakash Murali, David C McKay, Margaret Martonosi, and Ali Javadi-Abhari. Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In *Proceedings of the 25th ACM/IEEE International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'20)*, pages 1001–1016, 2020. doi:10.1145/3373376.3378477.

[MMT+21] Zlatko K. Minev, Thomas G. McConkey, Maika Takita, Antonio D. Corcoles, and Jay M. Gambetta. Circuit quantum electrodynamics (cqed) with modular quasi-lumped models, 2021. doi:10.48550/arXiv.2103.10344.

[Mon16] David Monniaux. A survey of satisfiability modulo theory. In *Proceedings of the 18th International Workshop on Computer Algebra in Scientific Computing (CASC'16)*, pages 401–425, 2016. doi:10.1007/978-3-319-45641-6.

[MPB+16] Doug T McClure, Hanhee Paik, Lev S Bishop, Matthias Steffen, Jerry M Chow, and Jay M Gambetta. Rapid driven reset of a qubit readout resonator. *Physical Review Applied*, 5(1):011001, 2016. doi:10.1103/PhysRevApplied.5.011001.

[MS08] Igor L. Markov and Yaoyun Shi. Simulating Quantum Computation by Contracting Tensor Networks. *SIAM Journal on Computing*, 38(3):963–981, 2008. doi:10.1137/050644756.

[MT06] D.M. Miller and M.A. Thornton. QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits. In *Proceedings of the 36th IEEE International Symposium on Multiple-Valued Logic (ISMVL'06)*, pages 30–30, 2006. doi:10.1109/ISMVL.2006.35.

[MVBS04] Mikko Möttönen, Juha J Vartiainen, Ville Bergholm, and Martti M Salomaa. Quantum circuits for general multiqubit gates. *Physical review letters*, 93(13):130502, 2004. doi:10.1103/PhysRevLett.93.130502.

[MvdGD23] Arianne Meijer-van de Griend and Ross Duncan. Architecture-aware synthesis of phase polynomials for nisq devices. *Electronic Proceedings in Theoretical Computer Science*, 394:116–140, 2023. doi:10.4204/eptcs.394.8.

[N+14] Kae Nemoto et al. Photonic architecture for scalable quantum information processing in diamond. *Physical Review X*, 4(3):31022, 2014. doi:10.1103/PhysRevX.4.031022.

[Nad23] Alexander Nadel. Solving huge instances with Intel (R) SAT solver. In *Proceedings of the 26th International Conference on Theory and Applications of Satisfiability Testing (SAT'23)*, 2023. doi:10.4230/LIPIcs.SAT.2023.17.

[NB22] Nina Narodytska and Nikolaj Bjørner. Analysis of core-guided MaxSAT using cores and correction sets. In *Proceedings of the 25th International Conference on Theory and Applications of Satisfiability Testing (SAT'22)*, 2022. doi:10.4230/LIPIcs.SAT.2022.26.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. ISBN: 978-1-107-00217-3.

[NGM20] Beatrice Nash, Vlad Gheorghiu, and Michele Mosca. Quantum circuit optimizations for NISQ architectures. *Quantum Science and Technology*, 5(2):025010, 2020. doi:10.1088/2058-9565/ab79b1.

[Nie06] Michael A Nielsen. Cluster-state quantum computation. *Reports on Mathematical Physics*, 57(1):147–161, 2006. doi:10.1016/s0034-4877(06)80014-5.

[NKS06] Yumi Nakajima, Yasuhito Kawano, and Hiroshi Sekigawa. A new algorithm for producing quantum circuits using KAK decompositions. *Quantum Info. Comput.*, 6(1):67–80, 2006. doi:10.26421/QIC6.1-5.

[NLR+14] F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys. Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries. *Physical Review X*, 4:021034, 2014. doi:10.1103/PhysRevX.4.021034.

[NMD21] Philipp Niemann, Luca Mueller, and Rolf Drechsler. Combining SWAPs and Remote CNOT Gates for Quantum Circuit Transformation. In *Proceedings of the 24th Euromicro Conference on Digital System Design (DSD'21)*, pages 495–501, 2021. doi:10.1109/DSD53832.2021.00080.

[NO79] Greg Nelson and Derek C Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979. doi:10.1145/357073.357079.

[NPS+20] Shin Nishio, Yulu Pan, Takahiko Satoh, Hideharu Amano, and Rodney Van Meter. Extracting success from IBM's 20-qubit machines using error-aware compilation. *ACM Journal on Emerging Technologies in Computing Systems*, 16(3):1–25, 2020. doi:10.1145/3386162.

[NT23] Paul D Nation and Matthew Treinish. Suppressing quantum circuit errors due to system variability. *PRX Quantum*, 4(1):010327, 2023. doi:10.1103/PrXQuantum.4.010327.

[NTS22a] Siyuan Niu and Aida Todri-Sanial. Analyzing strategies for dynamical decoupling insertion on IBM quantum computer, 2022. doi:10.48550/arXiv.2204.14251.

[NTS22b] Siyuan Niu and Aida Todri-Sanial. Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers. *IEEE Transactions on Quantum Engineering*, 3:1–10, 2022. doi:10.1109/TQE.2022.3203153.

[Oli93] Pamela E. Oliver. Formal Models of Collective Action. *Annual Review of Sociology*, 19(1):271–300, 1993. doi:10.1146/AnnuRev.So.19.080193.001415.

[OOCG20] F. Orts, G. Ortega, E. F. Combarro, and E. M. Garzón. A review on reversible quantum adders. *Journal of Network and Computer Applications*, 170:102810, 2020. doi:10.1016/j.jnca.2020.102810.

[Opp80] Derek C Oppen. Complexity, convexity and combinations of theories. *Theoretical computer science*, 12(3):291–302, 1980. doi:10.1016/0304-3975(80)90059-6.

[Pas23] Pasqal. *Pasqal*, Accessed 30th December, 2023.

[PCZ22] Feng Pan, Keyang Chen, and Pan Zhang. Solving the Sampling Problem of the Sycamore Quantum Circuits. *Physical Review Letters*, 129(9):090502, 2022. doi:10.1103/PhysRevLett.129.090502.

[PDF+21] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis. Demonstration of the trapped-ion quantum CCD computer architecture. *Nature*, 592(7853):209–213, 2021. doi:10.1038/s41586-021-03318-.

[Ped23] Edwin Pednault. An alternative approach to optimal wire cutting without ancilla qubits, 2023. doi:10.48550/arXiv.2303.08287.

[PER+20] Luca Petit, HGJ Eenink, M Russ, WIL Lawrie, NW Hendrickx, SGJ Philips, JS Clarke, LMK Vandersypen, and M Veldhorst. Universal quantum logic in hot silicon qubits. *Nature*, 580(7803):355–359, 2020. doi:10.1038/s41586-020-2170-7.

[PF21] Archimedes Pavlidis and Emmanuel Floratos. Arithmetic Circuits for Multilevel Qudits Based on Quantum Fourier Transform. *Physical Review A*, 103(3):032417, 2021. doi:10.1103/PhysRevA.103.032417.

[PHOW20] Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125:150504, 2020. doi:10.1103/PhysRevLett.125.150504.

[PMS+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213, 2014. doi:10.1038/NComms5213.

[PPJ+21] Jessica Pointing, Oded Padon, Zhihao Jia, Henry Ma, Auguste Hirth, Jens Palsberg, and Alex Aiken. Quanto: Optimizing quantum circuits with automatic generation of circuit identities, 2021. doi:10.48550/arXiv.2111.11387.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. doi:10.22331/q-2018-08-06-79.

[PRE+22] Luca Petit, Maximilian Russ, Gertjan H. G. J. Eenink, William I. L. Lawrie, James S. Clarke, Lieven M. K. Vandersypen, and Menno Veldhorst. Design and integration of single-qubit rotations and two-qubit gates in silicon above one Kelvin. *Communications Materials*, 3(1):1–7, 2022. doi:10.1038/s43246-022-00304-9.

[PS23] Christophe Piveteau and David Sutter. Circuit knitting with classical communication. *IEEE Transactions on Information Theory*, 2023. doi:10.1109/TIT.2023.3310797.

[PWD16] Alexandru Paler, Robert Wille, and Simon J Devitt. Wire recycling for quantum circuit optimization. *Physical Review A*, 94(4):042337, 2016. doi:10.1103/PhysRevA.94.042337.

[QBW23a] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. Mqt predictor: Automatic device selection with device-specific circuit compilation for quantum computing, 2023. doi:10.48550/arXiv.2310.06889.

[QBW23b] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. Reducing the compilation time of quantum circuits using pre-compilation on the gate level. In *Proceedings of the 4th IEEE International Conference on Quantum Computing and Engineering (QCE'23)*, 2023. doi:10.1109/QCE57702.2023.00091.

[Qis23] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023. doi:10.5281/zenodo.2573505.

[QRy23] QRydDemo. *QRydDemo*, Accessed 30th December, 2023.

[QUE23] QUERA. *QUERA*, Accessed 30th December, 2023.

[QuT23] QuTech. *Quantum Inspire Quantum Systems*, Accessed 30th December, 2023.

[RDH14] Md Mazder Rahman, Gerhard W Dueck, and Joseph D Horton. An algorithm for quantum template matching. *ACM Journal on Emerging Technologies in Computing Systems*, 11(3):1–20, 2014. doi:10.1145/2629537.

[RH07] Robert Raussendorf and Jim Harrington. Fault-Tolerant Quantum Computation with High Threshold in Two Dimensions. *Physical Review Letters*, 98(19):190504, 2007. doi:10.1103/PhysRevLett.98.190504.

[Ric68] D. Richardson. Some undecidable problems involving elementary functions of a real variable. *The Journal of Symbolic Logic*, 33:514–520, 1968. doi:10.2307/2271358.

[Rig23] Rigetti. *Rigetti Quantum Systems*, Accessed 30th December, 2023.

[RMH+14] Lee A. Rozema, Dylan H. Mahler, Alex Hayat, Peter S. Turner, and Aephraim M. Steinberg. Quantum Data Compression of a Qubit Ensemble. *Physical Review Letters*, 113(16):160504, 2014. doi:10.1103/PhysRevLett.113.160504.

[RPL+19] Kenneth Rudinger, Timothy Proctor, Dylan Langharst, Mohan Sarovar, Kevin Young, and Robin Blume-Kohout. Probing Context-Dependent Errors in Quantum Processors. *Physical Review X*, 9(2):021045, 2019. doi:10.1103/PhysRevX.9.021045.

[RRG+22] Kenneth Rudinger, Guilhem J Ribeill, Luke CG Govia, Matthew Ware, Erik Nielsen, Kevin Young, Thomas A Ohki, Robin Blume-Kohout, and Timothy Proctor. Characterizing midcircuit measurements on a superconducting qubit using gate set tomography. *Physical Review Applied*, 17(1):014014, 2022. doi:10.1103/PhysRevApplied.17.014014.

[RS20] Shesha Raghunathan and Leon Stok. Eda and quantum computing: a symbiotic relationship? *IEEE Design & Test*, 37(6):71–78, 2020. doi:10.1109/MDAT.2020.3016586.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

[RWS+17] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the national academy of sciences*, 114(29):7555–7560, 2017. doi:10.1073/PNAS.1619152114.

[Saf16] Mark Saffman. Quantum computing with atomic qubits and rydberg interactions: progress and challenges. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 49(20):202001, 2016. doi:10.1088/0953-4075/49/20/202001.

[SBM06] Vivek V Shende, Stephen S Bullock, and Igor L Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006. doi:10.1109/TCAD.2005.855930.

[Sch19] Maximilian Schlosshauer. Quantum decoherence. *Physics Reports*, 831:1–57, 2019. doi:10.1016/J.PhysRep.2019.10.001.

[SER+23] V. V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret. Real-time quantum error correction beyond break-even. *Nature*, 616(7955):50–55, 2023. doi:10.1038/s41586-023-05782-6.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th IEEE Annual Symposium on Foundations of Computer Science (SFCS'93)*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700.

[Sin05] Carsten Sinz. Towards an optimal cnf encoding of boolean cardinality constraints. In *Proceedings of the Principles and Practice of Constraint Programming (CP'05)*, 2005. doi:10.1007/11564751.

[SKK23] Movahhed Sadeghi, Soheil Khadirsharbiyani, and Mahmut Taylan Kandemir. Quantum circuit resizing, 2023. doi:10.48550/arXiv.2301.00720.

[SM13] Mehdi Saeedi and Igor L. Markov. Synthesis and optimization of reversible circuits—a survey. *ACM Computing Surveys*, 45(2):21:1–21:34, 2013. doi:10.1145/2431211.2431220.

[SNC09] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT'09)*, pages 244–257, 2009. doi:10.1007/978-3-642-02777-2.

[SPR+20] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. Detecting crosstalk errors in quantum information processors. *Quantum*, 4:321, 2020. doi:10.22331/q-2020-09-11-321.

[SSCP18] Marcos Yukio Siraichi, Vinícius Fernandes dos Santos, Sylvain Collange, and Fernando Magno Quintão Pereira. Qubit allocation. In *Proceedings of the 16th ACM/IEEE International Symposium on Code Generation and Optimization (CGO'18)*, pages 113–125, 2018. doi:10.5753/CTD.2020.11368.

[Sto21] Leon Stok. Eda and quantum computing: The key role of quantum circuits. In *Proceedings of the 24th ACM International Symposium on Physical Design (ISPD '21)*, page 111, 2021. doi:10.1145/3439706.3446900.

[TBG17] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error Mitigation for Short-Depth Quantum Circuits. *Physical Review Letters*, 119(18):180509, 2017. doi:10.1103/PhysRevLett.119.180509.

[TBLC22] Bochen Tan, Dolev Bluvstein, Mikhail D Lukin, and Jason Cong. Qubit mapping for reconfigurable atom arrays. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD'22)*, pages 1–9, 2022. doi:10.1145/3508352.3549331.

[TC20] Bochen Tan and Jason Cong. Optimal layout synthesis for quantum computing. In *Proceedings of the 39th IEEE/ACM International Conference on Computer-Aided Design (ICCAD'20)*, 2020. doi:10.1145/3400302.3415620.

[TQ19] Swamit S. Tannu and Moinuddin K. Qureshi. Not all qubits are created equal: a case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the 24th ACM/IEEE International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*, pages 987–999, 2019. doi:10.1145/3297858.3304007.

[Tse83] Grigori S Tseitin. On the complexity of derivation in propositional calculus. *Automation of reasoning: 2: Classical papers on computational logic 1967–1970*, pages 466–483, 1983. doi:10.1007/978-3-642-81955-1.

[TSSF24] Joseph Tindall, Miles Stoudenmire, Dries Sels, and Matthew Fishman. Efficient tensor network simulation of IBM's eagle kicked ising experiment. *Bulletin of the American Physical Society*, 2024. doi:10.48550/arXiv.2306.14887.

[TTS+21] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. CutQC: using small quantum computers for large quantum circuit evaluations. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'21)*, pages 473–486, 2021. doi:10.1145/3445814.3446758.

[Tuc05] Robert R. Tucci. An introduction to Cartan's KAK decomposition for QC programmers, 2005. doi:10.48550/arXiv.quant-ph/0507171.

[UJH+09] E Urban, Todd A Johnson, T Henage, L Isenhower, DD Yavuz, TG Walker, and M Saffman. Observation of rydberg blockade between two atoms. *Nature Physics*, 5(2):110–114, 2009. doi:10.1038/NPhys1178.

[UPR+23] Christian Ufrecht, Maniraman Periyasamy, Sebastian Rietsch, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. Cutting multi-control quantum gates with ZX calculus. *Quantum*, 7:1147, 2023. doi:10.22331/q-2023-10-23-1147.

[vdW20] John van de Wetering. ZX-calculus for the working quantum computer scientist, 2020. doi:10.48550/arXiv.2012.13966.

[VKO+20] Antti P. Vepsäläinen, Amir H. Karamlou, John L. Orrell, Akshunna S. Dogra, Ben Loer, Francisca Vasconcelos, David K. Kim, Alexander J. Melville, Bethany M. Niedzielski, Jonilyn L. Yoder, Simon Gustavsson, Joseph A. Formaggio, Brent A. VanDevender, and William D. Oliver. Impact of ionizing radiation on superconducting qubit coherence. *Nature*, 584(7822):551–556, 2020. doi:10.1038/s41586-020-2619-8.

[VL91] Jan Van Leeuwen. *Handbook of theoretical computer science (vol. A) algorithms and complexity*. MIT Press, 1991. ISBN: 978-0-444-88071-0.

[VMB22] Vivien Vandaele, Simon Martiel, and Timothée Goubault de Brugière. Phase polynomials synthesis algorithms for NISQ architectures and beyond. *Quantum Science and Technology*, 7(4):045027, 2022. doi:10.1088/2058-9565/ac5a0e.

[VT99] Guifré Vidal and Rolf Tarrach. Robustness of entanglement. *Physical Review A*, 59:141–155, 1999. doi:10.1103/PhysRevA.59.141.

[WBB+14] Xin Wang, Lev S. Bishop, Edwin Barnes, J. P. Kestner, and S. Das Sarma. Robust quantum gates for singlet-triplet spin qubits using composite pulses. *Physical Review A*, 89(2):022310, 2014. doi:10.1103/PhysRevA.89.022310.

[WBZ19] Robert Wille, Lukas Burgholzer, and Alwin Zulehner. Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations. In *Proceedings of the 56th ACM/IEEE Annual Design Automation Conference (DAC'19)*, pages 1–6, 2019. doi:10.1145/3316781.3317859.

[WCL+23] Qingfeng Wang, Ze-Pei Cian, Ming Li, Igor L. Markov, and Yunseong Nam. Ever more optimized simulations of fermionic systems on a quantum computer. In *Proceedings of the 60th ACM/IEEE Design Automation Conference (DAC'23)*, 2023. doi:10.1109/DAC56929.2023.10247693.

[WGT+08] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler. RevLib: An online resource for reversible functions and reversible circuits. In *Proceedings of the 38th IEEE International Symposium on Multiple Valued Logic (ISMVL'08)*, pages 220–225, 2008. doi:10.1109/ISMVL.2008.43.

[WHWH20] Mark Webber, Steven Herbert, Sebastian Weidt, and Winfried K. Hensinger. Efficient Qubit Routing for a Globally Connected Trapped Ion Quantum Computer. *Advanced Quantum Technologies*, 3(8):2000027, 2020. doi:10.1002/QuTe.202000027.

[WKAGR08] Hefeng Wang, Sabre Kais, Alán Aspuru-Guzik, and Mark R. Hoffmann. Quantum algorithm for obtaining the energy spectrum of molecular systems. *Physical Chemistry Chemical Physics*, 10(35):5388–5393, 2008. doi:10.1039/B804804E.

[WKM21] Robert Wille, Oliver Keszocze, and Naser Mohammadzadeh. Exact physical design of quantum circuits for ion-trap-based quantum architectures. In *Proceedings of the 24th ACM/IEEE Design, Automation and Test Europe Conference (DATE'21)*, 2021. doi:10.23919/DATE51398.2021.9474188.

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st ACM SIGACT Annual Symposium on Theory of Computing*, pages 515–526, 2019. doi:10.1145/3313276.3316404.

[WLD14] Robert Wille, Aaron Lye, and Rolf Drechsler. Optimal swap gate insertion for nearest neighbor quantum circuits. In *Proceedings of the 19th ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC'14)*, pages 489–494, 2014. doi:10.1109/ASPDAC.2014.6742939.

[XLL+21] Huikai Xu, Weiyang Liu, Zhiyuan Li, Jiaxiu Han, Jingning Zhang, Kehuan Linghu, Yongchao Li, Mo Chen, Zhen Yang, Junhua Wang, et al. Realization of adiabatic and diabatic CZ gates in superconducting qubits coupled with a tunable coupler. *Chinese Physics B*, 30(4):044212, 2021. doi:10.1088/1674-1056/abf03a.

[YTO+18] Jun Yoneda, Kenta Takeda, Tomohiro Otsuka, Takashi Nakajima, Matthieu R Delbecq, Giles Allison, Takumu Honda, Tetsuo Kodera, Shunri Oda, Yusuke Hoshi, et al. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%. *Nature nanotechnology*, 13(2):102–106, 2018. doi:10.1038/s41565-017-0014-x.

[ZCL+20] Chengxian Zhang, Tao Chen, Sai Li, Xin Wang, and Zheng-Yuan Xue. High-fidelity geometric gate for silicon-based spin qubits. *Physical Review A*, 101(5):052302, 2020. doi:10.1103/PhysRevA.101.052302.

[ZDTT23] Zeqiao Zhou, Yuxuan Du, Xinmei Tian, and Dacheng Tao. QAOA-in-QAOA: Solving Large-Scale MaxCut Problems on Small Quantum Machines. *Physical Review Applied*, 19(2):024027, 2023. doi:10.1103/PhysRevApplied.19.024027.

[ZHW19] Alwin Zulehner, Stefan Hillmich, and Robert Wille. How to Efficiently Handle Complex Values? Implementing Decision Diagrams for Quantum Computing. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'19)*, pages 1–7, 2019. doi:10.1109/ICCAD45719.2019.8942057.

[ZPW18] Alwin Zulehner, Alexandru Paler, and Robert Wille. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Transactions on CAD*, 38(7):1226–1236, 2018. doi:10.1109/TCAD.2018.2846658.

[ZSW20] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What Limits the Simulation of Quantum Computers? *Physical Review X*, 10(4):041038, 2020. doi:10.1103/PhysRevX.10.041038.

[ZWD+20] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. doi:10.1126/Science.abe8770.

Index

Publications of the Author

[BPK23a] Sebastian Brandhofer, Ilia Polian, and Kevin Krsulich. Optimal partitioning of quantum circuits using gate cuts and wire cuts. *IEEE Transactions on Quantum Engineering*, 2023. doi:10.1109/TQE.2023.3347106.

[BCC+23] Sebastian Brandhofer, Ziang Chen, Li-Wei Chen, Xianyue Zhao, Nan Du, and Ilia Polian. Exploring gate-diversity enabled by reconfigurable memristive technology. In *Proceedings of the 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS'23)*, Istanbul, Turkey, December 2023.

[BPK23b] Sebastian Brandhofer, Ilia Polian, and Kevin Krsulich. Optimal qubit reuse for near-term quantum computers. In *Proceedings of the 4th IEEE International Conference on Quantum Computing and Engineering (QCE'23)*, Bellevue, USA, September 2023. doi:10.1109/QCE57702.2023.00100.

[BKNB23] Sebastian Brandhofer, Jinwoong Kim, Siyuan Niu, and Nicholas T. Bronn. SAT-Based Quantum Circuit Adaptation. In *Proceedings of the ACM/IEEE Conference on Design, Automation Test in Europe (DATE'23)*, Antwerp, Belgium, April 2023. doi:10.23919/DATE56975.2023.10137140.

[BBD+23] Sebastian Brandhofer, Daniel Braun, Vanessa Dehn, Gerhard Hellstern, Matthias Hüls, Yanjun Ji, Ilia Polian, Amandeep Singh Bhatia, and Thomas Wellens. Benchmarking the performance of portfolio optimization with QAOA. *Quantum Information Processing*, 22(1):1–27, 2023. doi:10.1007/s11128-022-03766-5.

[GKP+23] Paul R. Genssler, Florian Klemme, Shivendra Singh Parihar, Sebastian Brandhofer, Girish Pahwa, Ilia Polian, Yogesh Singh Chauhan, and Hussam Amrouch. Cryogenic embedded system to support quantum computing: From 5-nm FinFET to full processor. *IEEE Transactions on Quantum Engineering*, 4:1–11, 2023. doi:10.1109/TQE.2023.3300833.

[BMDP23] Sebastian Brandhofer, Casey R Myers, Simon J Devitt, and Ilia Polian. Multiplexed pseudo-deterministic photon source with asymmetric switching elements, 2023. doi:10.48550/arXiv.2301.07258.

[JBP22] Yanjun Ji, Sebastian Brandhofer, and Ilia Polian. Calibration-aware transpilation for variational quantum optimization. In *Proceedings of the 3rd IEEE International Conference on Quantum Computing and Engineering (QCE'22)*, Broomfield, USA, September 2022. doi:10.1109/QCE53715.2022.00040.

[BDP21a] Sebastian Brandhofer, Simon Devitt, and Ilia Polian. Error analysis of the variational quantum eigensolver algorithm. In *Proceedings of the 16th IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH'21)*, Virtual, November 2021. doi:10.1109/NANOARCH53687.2021.9642249.

[BBP21] Sebastian Brandhofer, Hans Peter Büchler, and Ilia Polian. Optimal mapping for near-term quantum architectures based on Rydberg atoms. In *Proceedings of the 40th IEEE International Conference On Computer-Aided Design (ICCAD'21)*, Munich, Germany, November 2021. doi:10.1109/ICCAD51958.2021.9643490.

[BDP21b] Sebastian Brandhofer, Simon Devitt, and Ilia Polian. ArsoNISQ: Analyzing quantum algorithms on near-term architectures. In *Proceedings of the 26th IEEE European Test Symposium (ETS'21)*, Virtual, May 2021. doi:10.1109/ETS50041.2021.9465414.

[BDWP21] Sebastian Brandhofer, Simon Devitt, Thomas Wellens, and Ilia Polian. Noisy intermediate-scale quantum (NISQ) computers: How they work, how they fail, how to test them? In *Proceedings of the 39th IEEE VLSI Test Symposium (VTS'21)*, Virtual, April 2021. doi:10.1109/VTS50974.2021.9441047.

[BKW20] Sebastian Brandhofer, Michael A. Kochte, and Hans-Joachim Wunderlich. Synthesis of Fault-Tolerant Reconfigurable Scan Networks. In *Proceedings of the ACM/IEEE Conference on Design, Automation Test in Europe (DATE'20)*, Virtual, March 2020. doi:10.23919/DATE48585.2020.9116525.

Declaration

All the work contained within this thesis,
except where otherwise acknowledged, was
solely the effort of the author.
At no stage was any collaboration entered into
with any other party.

Razvan-Paul-Sebastian brandhofer