Automatic De-Quantization of Quantum Programs Using Constant Propagation

Lian Remme*†, Alexander Weinert*, Andre Waschk*, Lukas Burgholzer†‡, Robert Wille†‡
*German Aerospace Center (DLR), Institute of Software Technology, Cologne, Germany
†Chair for Design Automation, Technical University of Munich, Munich, Germany
‡Munich Quantum Software Company GmbH, Garching near Munich, Germany
lian.remme@dlr.de alexander.weinert@dlr.de andre.waschk@dlr.de
lukas.burgholzer@tum.de robert.wille@tum.de

***Abstract*—Quantum computing promises to solve problems beyond the reach of classical computers, but today's quantum hardware is error-prone and much slower than classical hardware. Every quantum operation is costly, making it crucial to minimize quantum resource usage in near-term algorithms. Quantum resources should only be used when they are truly essential for quantum advantage, and not wasted on operations that can be efficiently handled by classical computation.**

In this work, we focus on *de-quantizing* quantum operations to classical computation whenever possible. The approach we propose for this is *hybrid quantum-classical constant propagation*, an optimization which reduces quantum operations by trading them for fast, reliable classical instructions. This is done by tracking between quantum and classical states to identify and eliminate unnecessary quantum gates and controls.

We formalize a hybrid state model for quantum-classical constant propagation, implement our optimizations in the open-source MQT Core tool, and evaluate them on benchmark circuits.

The obtained results show that quantum-classical constant propagation can reduce costly multi-qubit operations, making quantum programs more practical and robust for near-term devices. This opens the door to new hybrid compiler strategies that leverage the best of both quantum and classical worlds.

***Index Terms*—quantum-classical computing, optimization, quantum-classical optimization, constant propagation, compilation, formal semantics**

I. Introduction

Quantum computing is a technology that uses the laws of quantum mechanics to process information. Unlike classical bits, quantum bits (qubits) can exist in superpositions of zero and one. Together with other properties like entanglement, this leads to quantum algorithms which perform certain computations faster than any known classical algorithm.

The quest for *quantum advantage*, i.e., for solving problems that are intractable for classical computers [1], has driven advances in quantum algorithms and hardware. Landmark algorithms such as Shor's prime factorization [2] and Grover's search [3] demonstrate the potential of quantum computation.

Yet, quantum calculations have a downside. In the current *Noisy Intermediate-Scale Quantum (NISQ)* era, qubits are fragile, error-prone, and quantum operations are orders of magnitude slower than their classical counterparts.

To make quantum computing practical, we must use quantum resources only when they are essential for quantum advantage. Whenever possible, we should *de-quantize* operations to classical processors, to leveraging their speed and reliability as well as avoiding unnecessary quantum operations.

This principle applies at the level of algorithms and *within* quantum calculations. We ask: Given a calculation, which parts *must* remain quantum to unlock quantum advantage? Which parts can efficiently be delegated to classical computation? By answering these questions, we can accelerate quantum programs and make their execution more robust.

We propose *quantum-classical constant propagation*: a static analysis that enables transformations which reduce quantum resource usage in quantum circuits. These semantic-preserving transformations replace quantum operations with classical computation and control. This is especially useful for multi-qubit gates, which are slow, error-prone, and challenging to implement on hardware with limited connectivity.

Our work expands the transformation routines available for quantum circuits. We build on quantum constant propagation by Chen and Stade, which does not consider quantum-classical interplay [4]. Similarly, our proposed Hadamard lifting extends measurement lifting by Rovara et al. [5].

We evaluate our transformations on benchmark circuits. Our experiments show that quantum calculations can benefit from quantum-classical constant propagation. We show that, in general, our quantum-classical approach reduces more quantum resources than the existing quantum constant propagation.

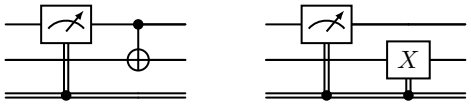
Some circuits benefit more from measurement lifting than from constant propagation. Hadamard lifting, while promising, does not provide additional improvements. We find that constant propagation and measurement lifting are complementary: each excels in scenarios where the other is less effective, making the combination of both particularly powerful.

Our main contributions are:

- Quantum-classical constant propagation: a technique for reducing quantum resources in hybrid quantum-classical computations.
- A formalization of the quantum-classical machine state, laying the foundation for constant propagation.
- An open-source implementation of constant propagation and Hadamard lifting in the Munich Quantum Toolkit (MQT) Core project [6][1].

[1]The functionality is given in a fork [7].

979-8-3195-1200-0/26/$31.00 ©2026 IEEE



(a) (b)

Fig. 1: An example for a dynamic circuit. The left and right circuits are equivalent. The quantum control is removed in exchange for a classical condition on a gate.

The remainder of the paper is organized as follows: Section II provides background and motivating examples. Section III surveys related work. Section IV describes Hadamard lifting. Section V formalizes our hybrid state model and complexity bounds and Section VI details the semantics-preserving transformations enabled by constant propagation. Finally, Section VII presents our evaluation, and Section VIII concludes with future directions.

II. Motivation and Background

In this section, we review the necessary background knowledge for our work as well as the challenges in the intersection of quantum and classical computing. We first give an overview over circuit-based quantum computing and the current state of the art of quantum hardware in Section II-A. Afterwards, we recall measurement lifting due to Rovara et al. [5] and motivate our Hadamard lifting as an extension in Section II-B. Finally, we recall quantum constant propagation due to Chen and Stade [4] and motivate our quantum-classical constant propagation as an extension in Section II-C.

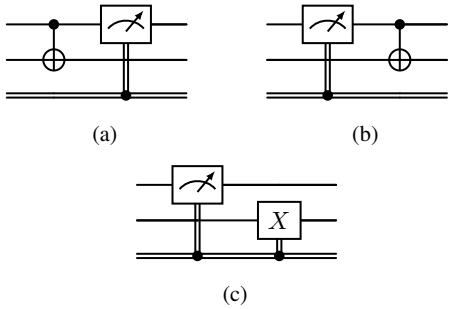
A. Dynamic Circuit-based Quantum Computing

Quantum circuits and their typical graphical notation are extensively used in this work. We refer the reader to [8, Chapter 4] for an introduction to these topics.

Currently available quantum computers are error-prone. Operations on IBM's superconducting Quantum Processing Units (QPUs) suffer error rates between 10^{-2} and 10^{-4}, with measurement being especially error-prone. Only single-qubit gates offer relative robustness [9–11]. IonQ's QPU Aria shows similar error rates (between 10^{-3} and 10^{-4} [12]). In contrast, the error rates of classical CPUs are negligible.

Quantum and classical computers also differ critically in their execution speed. IBM's superconducting QPUs execute a few hundred thousand layer operations per second [9–11]. Executing a single instruction on IonQ's Aria [12] can take hundreds of microseconds ($135\,\mu s$ for one-qubit operations, $600\,\mu s$ for two-qubit gate operations). This translates to just a few thousand quantum operations per second. Meanwhile, classical CPUs execute billions of instructions per second [13].

A typical approach to reduce the errors of faulty quantum resources is the use of *dynamic circuits*, i.e., hybrid circuits that use gates guarded by registers. Dynamic circuits can use measurement outcomes to control subsequent quantum operations or to trigger classical computation and branching.



(a) (b)

(c)

Fig. 2: Measurement lifting in action: By commuting the measurement from Figure 2a forward to create the circuit in Figure 2b, a quantum control can be replaced with a classical control. This creates the circuit in Figure 2c and eliminates a costly two-qubit gate. All three circuits are equivalent.

Thus, they unlock new possibilities for efficiency. For instance, when a quantum control immediately follows a measurement, it can be replaced with a classical condition, eliminating costly multi-qubit gates. We illustrate this simplification in Figure 1a and Figure 1b. This strategy has been applied manually for practical gains: Algorithms like iterative phase estimation [14] and Shor's algorithm for $2n + 1$ qubits [15] exploit dynamic circuits to minimize quantum resource usage.

Since currently available quantum computers are severely limited in terms of their available quantum resources, the optimization of quantum circuits is an active area of research [5, 16–29]. One goal of the optimizations is to reduce quantum resources, e. g. the number of qubits [5, 20], circuit depth [20–22, 24], or gate count [20, 23–26]. This is usually done by removing unnecessary quantum operations. However, another approach is to *replace* quantum operations with classical operations whenever possible.

In the following subsections, we present two particular optimizations, namely quantum constant propagation by Chen and Stade [4] and measurement lifting by Rovara et al. [5]. We moreover motivate our respective extensions, namely quantum-classical constant propagation and Hadamard lifting.

B. Measurement Lifting

Measurement lifting was introduced by Rovara et al. [5] as part of their qubit-reuse strategy. It is an optimization that pushes measurements as far forward in the circuit as possible. This exposes opportunities to simplify or eliminate quantum operations, making circuits more efficient and robust.

The core idea is that by commuting measurements past certain gates, we can often replace quantum controls with classical ones, or remove unnecessary gates altogether. We illustrate this in Figure 2, where we commute a measurement past a quantum control, i.e., we move from Figure 2a to Figure 2b. Afterwards, the quantum control can be replaced by a classical control, seen in Figure 2c. This reduces the number of expensive two-qubit gates.

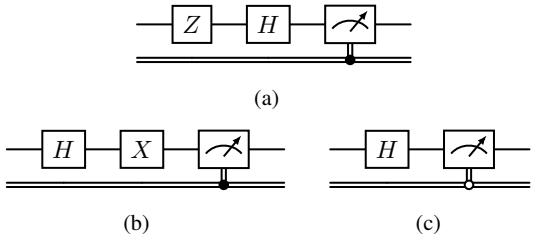


Fig. 3: Hadamard lifting: By commuting Pauli and Hadamard gates, measurements can be lifted further, enabling more quantum resource reduction. To create the circuit in Figure 3b from Figure 3a, the Hadamard gate is lifted in front of the Pauli gate. Afterwards, the measurement is lifted in front of the Pauli gate and the result negated, to create Figure 3c. All three circuits are equivalent.

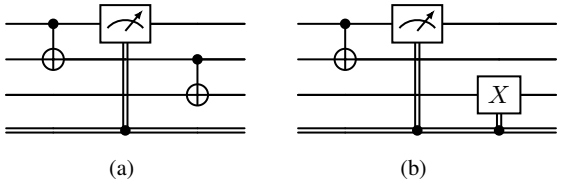


Fig. 4: A circuit for which constant propagation is useful: the state of the second qubit becomes determined by a prior measurement in Figure 4a, allowing a quantum control to be replaced by a classical condition in Figure 4b. Both circuits are equivalent.

Measurements can be commuted with phase gates since they do not affect measurement outcomes, and with Pauli X and Y gates if the measurement outcome is negated. Measurement lifting can reduce quantum control and create gates that do not influence subsequent measurements. These gates can then be trivially removed, decreasing the depth of the circuit.

Measurement lifting can be blocked by Hadamard gates before a measurement. As Hadamard gates put qubits into a superposition, they prevent measurements from being lifted. To overcome this, we introduce *Hadamard lifting*: a technique that leverages the commutation relations between Hadamard and Pauli gates to move Hadamards away from measurements.

For example, a Hadamard gate followed by a Pauli-Z gate is equivalent to a Pauli-X gate followed by a Hadamard gate. By applying these commutation rules we can transform circuits so that measurements can be lifted even further, unlocking new opportunities for classical control and further quantum resource reduction.

We illustrate this in Figure 3, where we first commute the Hadamard gate in front of the Pauli gate, obtaining the circuit in Figure 3b from that in Figure 3a. Afterward, we can lift the measurement in front of the Pauli gate to obtain the circuit in Figure 3c.

C. Quantum Constant Propagation

Hadamard and measurement lifting are pattern-based optimization routines. While this makes the algorithms straightforward to run, it reduces the cases which they can be applied to. For example, the circuit in Figure 4a would not be improved by measurement lifting, even though a transformation to the circuit in Figure 4b would be possible.

We could aim to find a comprehensive catalog of patterns for replacing quantum control by classical control. However, we aim for a more general solution: constant propagation. We simulate a restricted, efficiently representable abstraction of the quantum-classical machine state. The analysis detects relations between qubits and registers that pattern matching misses. Using these relations, we can then replace quantum control by classical control.

Constant propagation is also used for classical programs. It propagates the data-flow of a program to check whether a variable has a fixed value at a certain point [30, Chapter 9.4].

Chen and Stade show how to propagate quantum-state information through a circuit to find superfluous gates [4]. If a control qubit is always 0, the controlled operation can be removed. If a control qubit is always 1, the control can be removed. Additionally, if a gate is controlled by more than one qubit, it is checked whether all its controls can be simultaneously satisfied. If not, the gate is removed. Also, if more than one qubit controls an operation, it is checked whether one or more controlling qubits are implied by others. In that case, the implied qubit(s) can be removed.

We extend this idea to dynamic circuits: Our quantum machine abstraction tracks both quantum amplitudes and correlated registers. This enables transformations as depicted in Figure 4.

III. Related Work

Measurements can provide different functionality in quantum circuits. Uotila et al. [31] evaluated the role of measurements in quantum circuits. They distinguish three different usages: Read-out at the end of static circuits, read-out in dynamic circuits and succeeding modifications depending on the read-out results, and measurements to do quantum error correction. These different usages of measurements emphasize how little static circuits utilize measurements. Our work modifies circuits such that measurements that used to be read-out at the end of static circuits become read-out in dynamic circuits.

There are several works about quantum circuit optimization. A survey by Karuppasamy et al. [20] classifies optimization techniques. There are heuristic ones which find effective solutions or approximations. Heuristics are often applied iteratively and swap, merge or decompose gates. Machine learning methods utilize machine learning for optimization. Unitary synthesis works with unitary matrices which represent quantum circuits. Circuits are supposed to be efficiently constructed and factorized from the matrices. The last technique is the algorithmic approach. In this case, systematic strategies are applied. The survey considers hybrid quantum classical algorithms, but not dynamic circuits or their optimization.

Another survey by Yan et al. [32] points out the increasing relevance of artificial intelligence in circuit optimization. They also contribute a website with a list of circuit synthesis or compilation strategies that use AI.

Arora et al. [25] contributed an optimization routine to reduce quantum gates. The routine locally optimizes sub-circuits and merges the sub-circuits later on. Hao, Xu and Tannu [33] provided unitary synthesis to reduce T count, Clifford gate count, and approximation errors. A hybrid evolutionary algorithm by Sünkel et al. [22] is used for circuit depth minimization. Rattacaso et al.'s optimization routine [16] optimizes circuits during quantum compilation routines.

Besides these routines are also some which use machine learning approaches. Praba et al. [21] worked on both supervised and reinforcement learning strategies. Quetschlich, Burgholzer and Wille [17], as well as Mills et al. [34], implemented reinforcement learning agents which decide on the optimize application order. Alpha-Tensor quantum is a reinforcement learning agent to reduce T gate count by Zen, Nägele and Marquardt [18]. Another reinforcement learning tool for circuit optimization, Quarl, has been introduced by Li, Ding und Xie [23].

All of the mentioned works work on static quantum circuits. This means that none of them utilized the possibility to use classical resources instead of quantum ones. In contrast, the following papers address work on dynamic circuits in the context of circuit optimization:

Remme, Weinert and Waschk evaluated hybrid optimization routines and how to consider classical computing resources within dynamic circuits [35]. Ye et al. [36] provided the Quantum Skeletal Program Enumeration (QSPE), a tool to validate and test compilers that work on dynamic circuits. They also created a set of dynamic circuits as benchmarks. Qukkos by Dang, Son and Kim [19] uses measurement-based quantum computing [37] to lower circuit depth and improve fidelity. Quantum circuits are transformed into measurement operations with adaptive corrections. The exact transformations are not given in the paper.

None of these works use classical calculations to reduce quantum resources. Our work fills this gap.

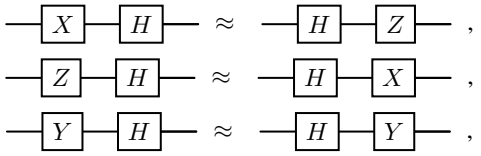
IV. Hadamard Lifting

In this section, we aim to improve the efficiency of measurement lifting [5]. We introduce Hadamard lifting: A gate reordering routine to improve the results of measurement lifting. This could provide additional effectiveness.

A. Lifting General Pauli-Gates Over Hadamard Gates

Measurement lifting [5] is an optimization routine which lifts measurements in front of certain gates. This is done to decrease the number of gates that are needed.

While measurements can be lifted over Pauli- and phase-gates, they cannot be lifted over Hadamard gates. To increase the efficiency of measurement lifting, we use the following commutation properties of Hadamard gates to move them away from measurements:



where we write $\approx$ to denote semantical equivalence of quantum circuits.

In case of commuting a Pauli-Y and a Hadamard gate, the circuit receives a global phase of -1, which we account for by adding a global phase gate to the circuit.

These commutations are also applicable if the Hadamard- and Pauli-X or Pauli-Z gate are controlled by exactly the same qubits. In this case, there may not be any other gate applied between the controlled gates. In case of a controlled Pauli-Y gate, a relative phase would be added by commuting the gates.

B. Lifting Controlled Pauli-Z-Gates Over Hadamard Gates

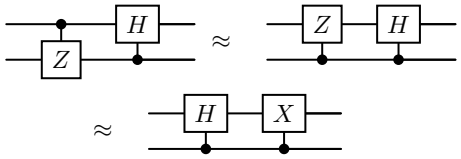
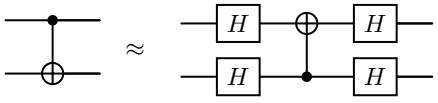


Fig. 5: We can commute a controlled Pauli-Z gate with a controlled Hadamard even if the targets are not applied to the same qubit. For this we use the fact that the Pauli-Z gate can be applied to any positively controlled control instead of the original target, meaning we can transform the first circuit into the second one. Afterward, the Hadamard- and Pauli-gate can be commuted, resulting in the third circuit.

If we have positively controlled Pauli-Z gates, it is not relevant which qubit is the target and which one is the control. Therefore, we can apply the routine depicted in Figure 5 to positively controlled Pauli-Z gates and Hadamard gates which are applied to the same qubits, even if they do not have the same target qubit.

C. Commuting a Single Hadamard Through CNOT

For CNOT gates, the following equivalence holds:



Therefore, if we have a Hadamard gate between a CNOT target and a measurement, we can do the transformation as depicted in Figure 6. This is useful for measurement lifting. After the transformation, the measurement can be lifted before the control, and the result of the measurement can be used as classical control, while the quantum control can be removed.

This routine introduces more gates to the circuit than it removes. However, as stated in Section II-A, multi-qubit gates take much more time and are much more error-prone than single-qubit gates. That justifies this transformation.

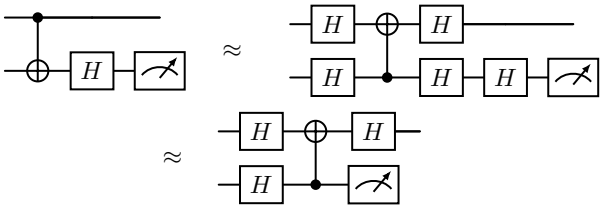


Fig. 6: If we have a CNOT target followed by a Hadamard gate followed by a measurement, one can change this to a measurement following the control. First, the target and control is flipped by applying four new Hadamard gates, which changes the circuit from the first to the second one. Then, the two Hadamard gates after each other cancel out, yielding the third circuit.

V. Constant Propagation

In this section, we will explain quantum-classical constant propagation and define the abstract state we use for this. Measurement lifting is applied to simplify the circuit close to measurements. Meanwhile, constant propagation is most effective at the start of a circuit.

We first recall some basic notation we require for this work.

We define $\mathbb{B} = \{0, 1\}$, we write $\mathbb{N}$ to denote the positive natural numbers, and we write $\mathbb{C}$ to denote the complex numbers. Moreover, for $n \in \mathbb{N}$ we define $[n] = \{1, \ldots, n\}$ and, given a sequence $\beta = b_1 \cdots b_n$ we define $\beta[i \to x] = b_1 \cdots b_{i-1} x b_{i+1} \cdots b_n$.

A. Concrete Execution

We begin by formalizing the semantics of executing a quantum circuit. In this section, we provide a denotational semantics of quantum circuits, which we view as a series of instructions. Each instruction is either a guarded or unguarded application of some gate to the qubits, a measurement of a qubit, or a reset of a qubit. We do not consider the parallelism inherent in the circuit notation and instead identify a circuit with an arbitrary, semantic-preserving interleaving of its instructions.

For the remainder of this work, fix some number $n \in \mathbb{N}$ of qubits. A *(concrete) quantum state* $|\psi\rangle$ is a mapping $|\psi\rangle \colon \mathbb{B}^n \to \mathbb{C}$ that satisfies $\left(\Sigma_{q \in \mathbb{B}^n} ||\psi\rangle(q)|^2 = 1\right)$. We write Ψ to denote the set of all concrete quantum states. Given a mapping $f \colon \mathbb{B}^n \to \mathbb{C}$ we define its normalization $|| f ||(q) = f(q)/(\Sigma_{q \in \mathbb{B}^n} |f(q)|^2)$.

For the remainder of this work, fix some number $m \in \mathbb{N}$ of registers. A *(concrete) hybrid state* $\varphi = (|\psi\rangle, \beta)$ comprises a quantum state $|\psi\rangle$ and a bit sequence $\beta \in \mathbb{B}^m$. We write Φ to denote the set of all hybrid states. A *(concrete) machine state* ρ is a probability distribution over Φ, i.e., we have $\rho \colon \Phi \to [0; 1]$ and $(\sum_{\varphi \in \Phi} \rho(\varphi)) = 1$.

We now define the effects of operations on machine states. We formalize these by defining a family of functions $[\![\cdot]\!]$ for the unguarded and guarded applications of unitary gates, as well as measurements and resets of qubits. The function $[\![\cdot]\!]$ takes a machine state as an input and yields a resulting machine state.

We first define the effect of applying a gate to a hybrid state. Let $\varphi = (|\psi\rangle, \beta)$ be a hybrid state and pick $|\psi'\rangle$ such that $U|\psi'\rangle = |\psi\rangle$. The probability of the machine being in state φ after applying U equals the probability of it being in state $|\psi'\rangle$ before applying U. As gate applications are reversible, $|\psi'\rangle$ is unique and we arrive at the following definition.

Definition V.1 ($[\![\texttt{app}\ U]\!]$)**.** Let U be a unitary gate. We define the effect of applying U to a machine state ρ as

$$[\![\texttt{app}\ U]\!](\rho) \colon (|\psi\rangle, \beta) \mapsto \rho(U^{-1}|\psi\rangle, \beta)$$

Gates can also be guarded by registers: A guarded gate U is applied only if certain registers are set to 1. W.l.o.g, we treat only this case in this work, and omit the dual case where U is applied if certain registers are set to 0.

Definition V.2 ($[\![\texttt{capp}\ U\ i_1 \ldots i_k]\!]$)**.** Let U be a unitary gate and let $i_1 < \cdots < i_k \in [n]$. We define the effect of applying U guarded by the registers $i_1, \ldots, i_k$ to the machine state ρ as

$$[\![\texttt{capp}\ U\ i_1 \ldots i_k]\!](\rho) \colon (|\psi\rangle, \beta) \mapsto \begin{cases} \rho(U^{-1}|\psi\rangle, \beta) & \text{if } \forall j \in [k] : b_j = 1 \\ \rho(|\psi\rangle, \beta) & \text{otherwise} \end{cases}$$

where $\beta = b_1 \cdots b_m$.

We now proceed to define the result of measuring qubits. Recall that measuring a qubit not only causes the state of the measured qubit to collapse, but also influences all qubits that are entangled with the measured one. Hence, we first define the effect of measuring on a quantum state.

Definition V.3 (Measurement)**.** Let $|\psi\rangle$ be a quantum state, let $b \in \mathbb{B}$ and let $i \in [n]$. We define the quantum state resulting from measuring the i-th qubit as value b as $|\psi\rangle_{i \to b} = || f ||$ where $f \colon \mathbb{B}^n \to \mathbb{C}$ with

$$f(q_1 \cdots q_n) = \begin{cases} |\psi\rangle(q_1 \cdots q_n) & \text{if } q_i = b \\ 0 & \text{otherwise} \end{cases}$$

We moreover define the probability of measuring the i-th qubit as value b as

$$\text{Prob}_{|\psi\rangle}(i \to b) = \Sigma_{q = q_1 \cdots q_n \in \mathbb{B}^n, q_i = b} |\, |\psi\rangle(q)\,|^2 \ .$$

Intuitively, we say that a hybrid state $\varphi = (|\psi\rangle, \beta)$ is a $i \to j$-predecessor of a hybrid state $\varphi' = (|\psi'\rangle, \beta')$ with $\beta' = b'_1 \cdots b'_m$ if β and β' differ at most position j and if $|\psi'\rangle$ results from $|\psi\rangle$ by measuring the i-th qubit as b'_j.

Definition V.4 ($i \to j$-predecessors)**.** Let $\varphi = (|\psi\rangle, \beta)$ with $\beta = b_1 \cdots b_m$ and $\varphi' = (|\psi'\rangle, \beta')$ with $\beta' = b'_1 \cdots b'_m$ be hybrid states. We say that φ is a $i \to j$*-predecessor* of φ' if $\beta' = \beta[j \to b'_j]$ for some $b'_j \in \mathbb{B}$ and if $|\psi'\rangle = |\psi\rangle_{i \to b'_j}$. We define $\text{mpred}_{i \to j}(\varphi')$ as the set of $i \to j$-predecessors of φ'.

We now define the effect of measuring the i-th qubit into the j-th register. Intuitively, the probability of obtaining a

hybrid state $(|\psi\rangle, \beta)$ with $\beta = b_1 \cdots b_m$ from the potential predecessor state $\varphi' = (|\psi'\rangle, \beta')$ by measuring the i-th qubit is the product of the probability $\rho(\varphi')$ of having obtained φ' beforehand and of the probability of measuring the i-th qubit in that state as b_j, which we write as $\text{Prob}_{|\psi'\rangle}(i \to b_j)$.

Definition V.5 ($[\![\texttt{meas}\ i \to j]\!]$)**.** We define

$$[\![\texttt{meas}\ i \to j]\!](\rho)\colon (\varphi) \mapsto \Sigma_{(|\psi'\rangle,\beta') \in \text{mpred}_{i\to j}(\varphi)} (\rho(|\psi'\rangle, \beta') \cdot \text{Prob}_{|\psi'\rangle}(i \to b_j)) \ ,$$

where $\beta = b_1 \cdots b_m$.

The definition of $[\![\texttt{meas}\ i \to j]\!](\rho)$ contains a sum over all potential $i \to j$ predecessors of the given hybrid state. As $\text{mpred}_{i\to j}(\varphi)$ is, in general, infinite, it is not trivial to compute this sum. We observe, however, that the summand reduces to 0 if $\rho(|\psi'\rangle, \beta') = 0$. Hence, as long as the machine state ρ assigns non-zero probabilities to only finitely many hybrid states, the machine state ρ' is straightforwardly computable.

We now proceed to define the effect of resetting the i-th qubit to 0. This can be implemented as a derived operation that is implemented by measuring the qubit and, if it is measured as 1, inverting it using an X-gate. This implementation, however, requires the measured value to be stored in some register although it is only used for the reset operation. Hence, we define the reset-operation natively.

We first define the effect of setting an individual qubit to a specific value. While this operation is not possible in real-life quantum computers, it allows us a more straightforward formalization of the reset-operation later on.

Definition V.6 (Setting qubits)**.** Let $|\psi\rangle$ be a quantum state, let $i \in [n]$, and let $b \in \mathbb{B}$. We write $|\psi\rangle[i \to b]$ to denote the quantum state $||\, f \,||$ with

$$f\colon (q) \mapsto \begin{cases} |\psi\rangle(q) + |\psi\rangle(q[i \to 1-b]) & \text{if } q_i = b \\ 0 & \text{otherwise,} \end{cases}$$

where $q = q_1 \cdots q_n$.

Using this operation allows us to define the $i{\downarrow_b}$-predecessors of a quantum state. Intuitively, a quantum state $|\psi\rangle$ is a $i{\downarrow_0}$-predecessor of a quantum state $|\psi'\rangle$ if $|\psi'\rangle$ results from $|\psi\rangle$ by measuring the i-th qubit as 0. Analogously, a quantum state $|\psi\rangle$ is a $i{\downarrow_1}$-predecessor of a quantum state $|\psi'\rangle$ if $|\psi'\rangle$ results from $|\psi\rangle$ by measuring the i-th qubit as 1 and that qubit is subsequently set to 0.

Definition V.7 ($i \downarrow_b$-predecessors)**.** Let $\varphi = (|\psi\rangle, \beta)$ with $\beta = b_1 \cdots b_m$ and $\varphi' = (|\psi'\rangle, \beta')$ with $\beta' = b'_1 \cdots b'_m$ be hybrid states. We say that φ is a *$i{\downarrow_0}$-predecessor* of φ' if $\beta' = \beta$ and if $|\psi'\rangle = |\psi\rangle_{i\to 0}$. We say that φ is a *$i{\downarrow_1}$-predecessor* of φ' if $\beta' = \beta$ and if $|\psi'\rangle = |\psi\rangle_{i\to 1}[i \to 0]$. For $b \in \mathbb{B}$ we define $\text{rpred}_{i{\downarrow_b}}(\varphi')$ as the set of $i \downarrow_b$-predecessors of φ'.

This definition now allows us to define the effect of the reset operation on a machine state analogously to the application of a measurement operation.

Definition V.8 ($[\![\texttt{reset}\ i]\!]$)**.** We define

$$[\![\texttt{reset}\ i]\!](\rho)\colon \varphi \mapsto \Sigma_{b\in\mathbb{B},(|\psi'\rangle,\beta') \in \text{rpred}_{i{\downarrow_b}}(\varphi)} (\rho(|\psi'\rangle, \beta') \cdot \text{Prob}_{|\psi'\rangle}(i \to b)) \ .$$

The above definitions provide us with a semantics of quantum circuits. In the next section we discuss how to obtain information about the machine state without the overhead of concrete execution.

B. Abstract Execution

In the previous section we have formalized the state of a quantum computer and its evolution during the execution of a quantum circuit. Each quantum machine state may assign non-zero probabilities to arbitrarily many hybrid states, each of which may comprise a sequence of m registers and one quantum state containing up to 2^n many non-zero amplitudes. Hence, the size of any individual quantum machine state is theoretically not bounded from above. To obtain constant propagation information efficiently, we impose a fixed upper limit on the size of quantum machine states in this section.

Definition V.9 (Representable states)**.** We call the set $\text{Ampl}(|\psi\rangle) = \{|\psi\rangle(q) \mid q \in \mathbb{B}^n, |\psi\rangle(q) \neq 0\}$ the *(non-zero) amplitudes* of $|\psi\rangle$. For $N \in \mathbb{N}$ we say that $|\psi\rangle$ is *N-representable* if $|\psi\rangle$ has at most N non-zero amplitudes. We say that the hybrid state $(|\psi\rangle, \beta)$ is *N-representable* if $|\psi\rangle$ is N-representable. We say that the machine state ρ is *N, M-representable* if ρ assigns a non-zero probability to at most M hybrid states.

Our abstract domain consists mainly of N, M-representable machine states together with the state $\top$ that represents all non-N, M-representable states.

Definition V.10 (Abstract states)**.** We define the set of *abstract hybrid states* as $\{\top\} \cup \{\varphi \in \Phi \mid \Phi \text{ is } N\text{-representable}\}$. We define the set of *abstract machine states* $\{\top\} \cup \{\rho \in \mathcal{P} \mid \rho \text{ is } N, M\text{-representable}\}$.

Having defined the concrete application of operations via $[\![\cdot]\!]$ we now proceed to define the abstract application.

Definition V.11 (Abstract execution)**.** Let $x \in \{\texttt{app}\ U, \texttt{capp}\ U\ \ i_1 \ldots i_k, \texttt{meas}\ i \to j, \texttt{reset}\ i\}$. We define the abstract execution function $[\![x]\!]^\#$ as follows:

$$[\![x]\!]^\#(\rho) = \begin{cases} \top & \text{if } \rho = \top \\ \top & \text{if } [\![x]\!](\rho) \text{ is not } N, M\text{-representable} \\ [\![x]\!](\rho) & \text{otherwise} \end{cases}$$

C. Efficient Machine State Storage via Hybrid Union Tables

In the last section we restricted the size of a machine state to not exceed a pre-defined threshold to counter the inherent exponential growth of quantum states [38]. In addition to this artificial upper bound, we aim to use the available space for keeping machine states as efficiently as possible. Chen and Stade [4] introduced union tables to efficiently store quantum states, which we now extend to efficiently store machine states.

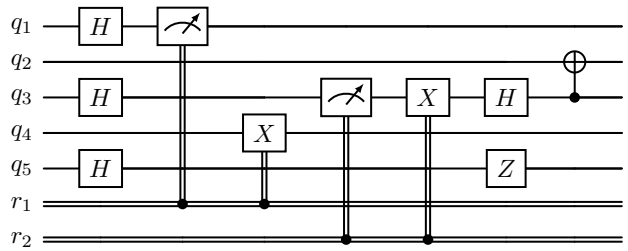


Fig. 7: An example of a hybrid quantum circuit. We show the resulting union table in Figure 8.

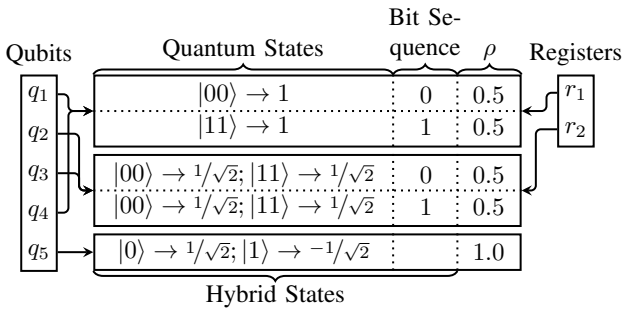


Fig. 8: The union table that results from the circuit in Figure 7. Adapted from [4, Figure 3].

The key insight is that one can store the amplitudes of non-entangled sets of qubits separately. Consider, e.g., a simple quantum circuit with three qubits, each of which has a Hadamard-gate applied to it. This yields the quantum state $|\psi\rangle$ with $|\psi\rangle(q) = 1/\sqrt{8}$ for each $q \in \mathbb{B}^3$. Storing $|\psi\rangle$ explicitly would require storing eight amplitudes, one for each combination of the results of measuring the three qubits individually. As the three qubits are not entangled, however, it suffices to consider each as an individual quantum system. This allows us to store the amplitudes of each qubit individually, resulting in six stored amplitudes, two for each qubit.

We extend this observation to the hybrid setting by considering a register i "entangled" with a qubit j if we (a) measure the value of qubit j into register i or if we (b) use register i to guard the application of a gate to qubit j. This allows us to store a machine state in a hybrid union table.

For the sake of brevity we omit a formal definition and instead refer to the example circuit shown in Figure 7 and its resulting union table shown in Figure 8. That union table contains three entries which characterize qubits q_1 and q_4 together with register r_1, qubits q_2 and q_3 together with register r_2, and qubit q_5 without any registers, respectively. If later on qubit q_1 and q_3 were to be entangled, the first and second entry of the union table would be merged as described by Chen and Stade [4].

VI. Transformation Routines via Constant Propagation

Using the constant propagation of the last section, we can apply transformation routines to the quantum circuit, which do not change the semantics. In this section, we will introduce these transformations. For the sake of brevity, we introduce notation for the case that a quantum state $|\psi\rangle$ assigns the value b to the i-th qubit: We define $|\psi\rangle_i = b \Leftrightarrow \forall b_1 \cdots b_n \in \mathbb{B}^n. b_i = 1 - b \Rightarrow |\psi\rangle(b_1 \cdots b_n) = 0$.

General Control Reduction: Chen and Stade [4] argued that quantum constant propagation can be used to reduce the number of controlled gates as follows. If we know that a controlling qubit is always 0 (1), we can remove the controlled gate (the control). By including registers into our analysis, we extend this principle to classical controls of an instruction. If we know that a controlling register will always be 0 (1), we can remove the instruction (the control).

Unsatisfiable Controls: As Chen and Stade pointed out, quantum constant propagation can also remove gates with multiple controls, if the combination of controls is not satisfiable [4]. We extend this principle to classical controls.

Quantum Control Reduction: A key goal of hybrid quantum constant propagation is to replace quantum resources by classical ones. One possible avenue for this is to replace qubit-controlled gates by register-controlled ones. This is done by checking whether there exist $i \in [n]$, $j \in [m]$ and $b, b' \in \mathbb{B}$ such that $|\psi\rangle_i = b \Leftrightarrow r_j = b'$. In either case, a gate controlled by qubit q_i can be replaced by one controlled (either positively or negatively) by register r_j.

This generalizes the measurement lifting depicted in Figure 1. It allows for routines like the one of Figure 4, without depending on pattern-matching certain circuit structures.

Implied Qubits/Registers: If a gate is controlled by multiple registers or qubits, we can remove some of the controls by checking for implications.

If there exist $i, j \in [n]$ and $b, b' \in \mathbb{B}$ such that q_i and q_j control the same gate and such that $|\psi\rangle_i = b \Rightarrow |\psi\rangle_j = b'$, we can remove either q_i or q_j from the controls. Analogously, if for $i \in [n]$ and $j \in [m]$ we have that q_i and r_j control the same gate and $r_j = b \Rightarrow |\psi\rangle_i = b'$, we can remove q_i from the controls.

Phase Gate Reduction: Phase gates are matrices that only have non-zero entries on their diagonal. If the phase gate applies the same phase to all non-zero amplitudes of the quantum state, the phase gate only applies a global phase and can be deleted. Instead, the global phase is added to the circuit, as long as it is not 1.

VII. Experimental Evaluation

The proposed implementation passes have been implemented on top of the MQT Compiler Collection [39] as part of the MQT Core project [6, 7], which is part of the MQT [40]. The implementation uses the Multi-Level Intermediate Representation (MLIR) framework [41, 42]. We evaluated measurement lifting [5], measurement lifting with Hadamard lifting (see Section IV), constant propagation (see Section V), and the

combination of constant propagation and measurement lifting. We also applied quantum constant propagation as proposed by Chen and Stade [4]. In this case, we did not implement the tool in MQT Core, but used the tool provided by them.

All passes were applied until a fixed point. When multiple passes were used they were applied alternately until convergence. Constant propagation used 16,4-representable machine states, i.e., we track at most 4 hybrid states, each of which comprises at most 16 non-zero amplitudes. Analogously, we allowed at maximum 16 nonzero amplitudes for quantum constant propagation.

We ran the optimization routines on the benchmark circuits of MQT Bench [43]. To use the circuits, we removed all `barrier` instructions. This is because they are mostly used for visual separation of different circuit parts in MQT Bench. Additionally, they would—as is their functionality—prevent us from applying the optimization techniques effectively and showing whether the techniques work in principle or not.

We ended up with 3 797 circuits to benchmark. The circuits implemented the following algorithms:

AE Amplitude estimation (2 to 19 qubits) [44],
BV Bernstein-Vazirani (2 to 200 qubits) [45],
CDKM CDKM ripple-carry adder circuit (2 to 200 qubits, even numbers only) [46, 47],
DJ Deutsch Josza (2 to 200 qubits) [48],
FA quantum full adders (4 to 200 qubits, even numbers only),
GHZ the creation of a GHZ state (2 to 200 qubits) [49],
GS the representation of a graph state (3 to 200 qubits),
Grover Grover's algorithm (2 to 20 qubits) [3],
HA quantum half adders (3 to 199 qubits, odd numbers only),
HHL Harrow–Hassidim–Lloyd (3 to 200 qubits) [50],
HRS HRS Cumulative Multiplier $(5, 9, 13, \ldots$ 197 qubits) [51],
MA modular adder circuits (2 to 200 qubits, even numbers only),
Multiplier quantum multipliers $(4, 8, 12, \ldots 108$ qubits),
QAOA quantum approximate optimization algorithm (2 to 200 qubits) [52],
QFT quantum fourier transform (2 to 200 qubits) [53],
QFTa Draper QFT Adder (2 to 200 qubits, even numbers only) [53],
QFTe QFT with a previous entangling routine for qubits (2 to 20 and 100 qubits),
QFTr RG QFT multiplier $(4, 8, 12, \ldots 92$ qubits) [54],
QNN quantum neural networks (2 to 200 qubits) [55],
QPEe/QPEi quantum phase estimation with an exact and inexact representation of the phase (2 to 200 qubits each) [56],
QUARK a cardinality and a copula circuit from the modeling application in the QUARK framework (2 to 200 qubits, copula circuits have an even number of qubits only) [57, 58],
QWalk quantum walk (3 qubits) [59],
Shor Shor's algorithm (18 and 42 qubits) [2],
VBE VBE ripple carry adder $(4, 7, 10, \ldots 199$ qubits) [46],
VQE variational quantum eigensolver with SU(2) ansatz [60], real amplitudes ansatz [61], and two local ansatz [62] (2 to 200 qubits each) [63]
WState the creation of a W-State (2 to 200 qubits) [64].

We show detailed results and the relative reduction of quantum resources achieved in Figure 9. The effectiveness of the optimization techniques depends heavily on the targeted circuits. In some algorithms, quantum resources decrease drastically, while in others, no change occurred.

The results show that there are algorithms which benefit from quantum-classical constant propagation, namely HA, HRS, quantum multiplier, QFT, QFTr, both QPEs, and circuits for the creation of the WState. In all cases, the hybrid constant propagation leads to a larger amount of reduced quantum resources than the other optimization techniques.

For other algorithms, we could not find an advantage of using a hybrid approach. Here, no difference could be detected between a hybrid and a full-quantum constant propagation. This applies to CDKM, FA, MA, QFTa, and VBE.

Measurement lifting does not reduce the total amount of gates, but can reduce the amount of controlled quantum gates. In some cases, this leads to a larger reduction of controlled gates than constant propagation can achieve. We observe this for GHZ, GS, Grover, QFTa, and both QPEs.

Applying Hadamard lifting to circuits on top of measurement lifting did not or only minimally improve the results. This could mean that the patterns targeted by Hadamard lifting do not appear in today's quantum algorithms.

The combination of measurement lifting and constant propagation can remove more quantum resources than using one algorithm on its own. This is the case for HHL, QFTe, and both QPEs. We assume the reason for this to be that constant propagation can find removable quantum resources which pattern matching approaches do not detect. It reduces quantum resources at the start of the circuits, and becomes worse as the circuit progresses. On the other hand, measurement lifting is especially powerful at the end, where all qubits are measured. Therefore, there can be circuits where both algorithms find distinct quantum resources to remove, which makes their combination especially powerful.

A special case in this regard is the HHL algorithm. Here, all individual optimization techniques yield almost no quantum resource reduction. In contrast, the combination of constant propagation and measurement lifting reduces quantum resources on average by over 70%. This is because measurement lifting reorders gates and measurements, which does not reduce the quantum resources. However, after reordering, constant propagation can be applied effectively.

For some algorithms, the application of constant propagation or measurement lifting results in a complete removal of controlled quantum gates. This has different reasons. One reason it that controls are zero if the qubits are $|0\rangle$ at the start of the circuit, as we assume for constant propagation. The circuits that represent additions or multiplications (CDKM, FA, HA, HRS, VBE) get heavily reduced in the number of controls in this case. In these cases, we need to keep in mind

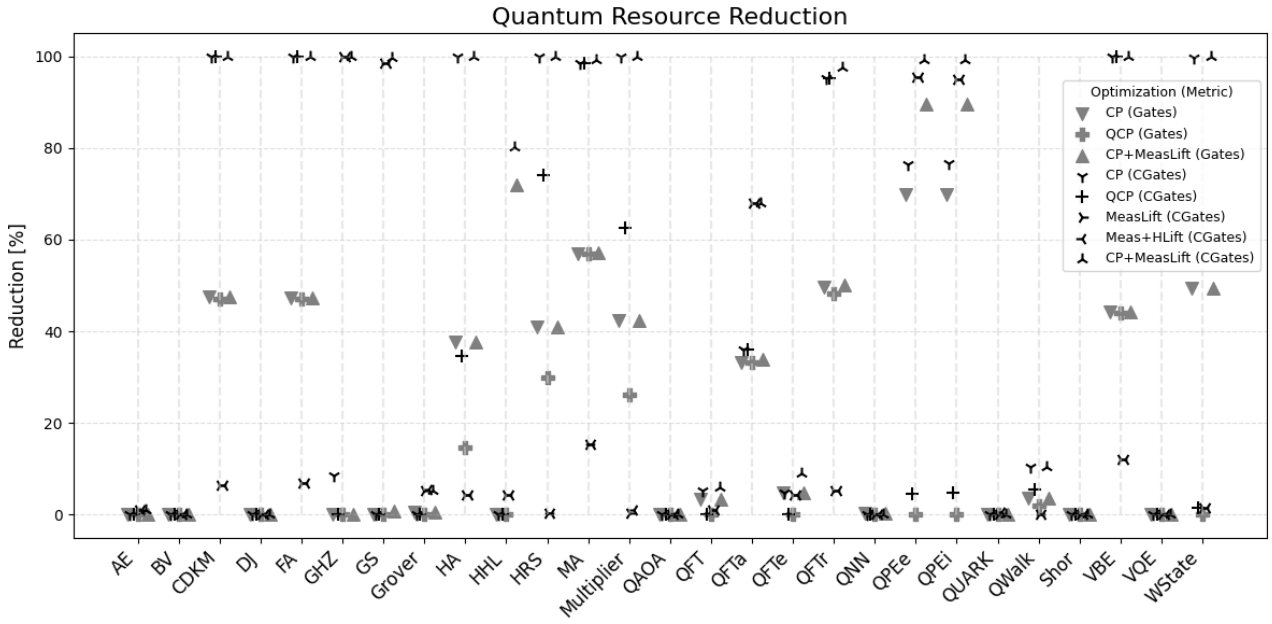


Fig. 9: The amount of quantum resources reduced on average in circuits implementing different algorithms by quantum-classical constant propagation (CP), quantum constant propagation (QCP), measurement lifting (MeasLift), measurement lifting with Hadamard lifting (Meas+HLift), and constant propagation with measurement lifting (QCP+MeasLift). "Gates" is the reduction of whole quantum gates, while "CGates" is the reduction of controlled gates. The latter is both the complete removal of the gates as well as turning the quantum controlled gates into non-quantum controlled gates.
%

that the results are different if the circuits are sub-circuits of larger ones and do not start with all qubits in $|0\rangle$ state.

In GHZ, QPE and WState, the removal of the controlled gates roots in a mix of measurements that directly follow controls (removed by measurement lifting), controls that are determined by registers and can become classically controlled, and controlled phase gates that add only a global phase.

In general, our findings show that hybrid constant propagation is a promising addition to existing algorithms. It can lead to the reduction of quantum resources when other optimization techniques fail, and enhances a compiler's ability to de-quantize a quantum circuit to make it more robust.

However, it needs to be pointed out that the compilation time of constant propagation is much larger than the one of measurement lifting. This is depicted in Figure 10.

For all algorithms except Grover, constant propagation increases the compilation time greatly. One way to deal with this could be to apply constant propagation to small and therefore quickly compiled versions of an algorithm. If the results of this routine are promising (i. e. quantum resources are reduced), there are two options: Constant propagation can be applied to a larger circuit and the longer compilation time is accepted, or the cases in which constant propagation can remove a gate are evaluated and applied to a larger circuit via pattern matching. In the second case, one has to make sure not to remove non-superfluous quantum resources unwillingly.

One observation is that, for some circuits, the combination of measurement lifting and constant propagation is faster than the execution of constant propagation alone. This could be explained by measurement lifting applying changes early that would be applied in later constant propagation iterations, causing a longer calculation time until a fixed point is reached.

Figure 10 shows that quantum constant propagation usually runs faster than other evaluated optimization techniques. This result has to be viewed with care, as we used a different tool for this routine. That tool did not use MLIR, hence it can be less easily integrated in a compilation stack.

VIII. Conclusion

We presented quantum-classical constant propagation as a strategy for reducing the quantum resources within a quantum circuit. We were driven by the practical constraints of NISQ devices: qubit operations, especially multi-qubit operations, are slow and error-prone. Therefore, removing unnecessary quantum resources yields direct gains in fidelity and runtime.

We combined our techniques with measurement lifting [5] and implemented them in MQT Core [6, 7].

We showed that quantum-classical constant propagation can effectively reduce quantum resources in cases where measurement lifting or quantum constant propagation cannot. We also observed that constant propagation takes a long time and does not guarantee reduced quantum resources.

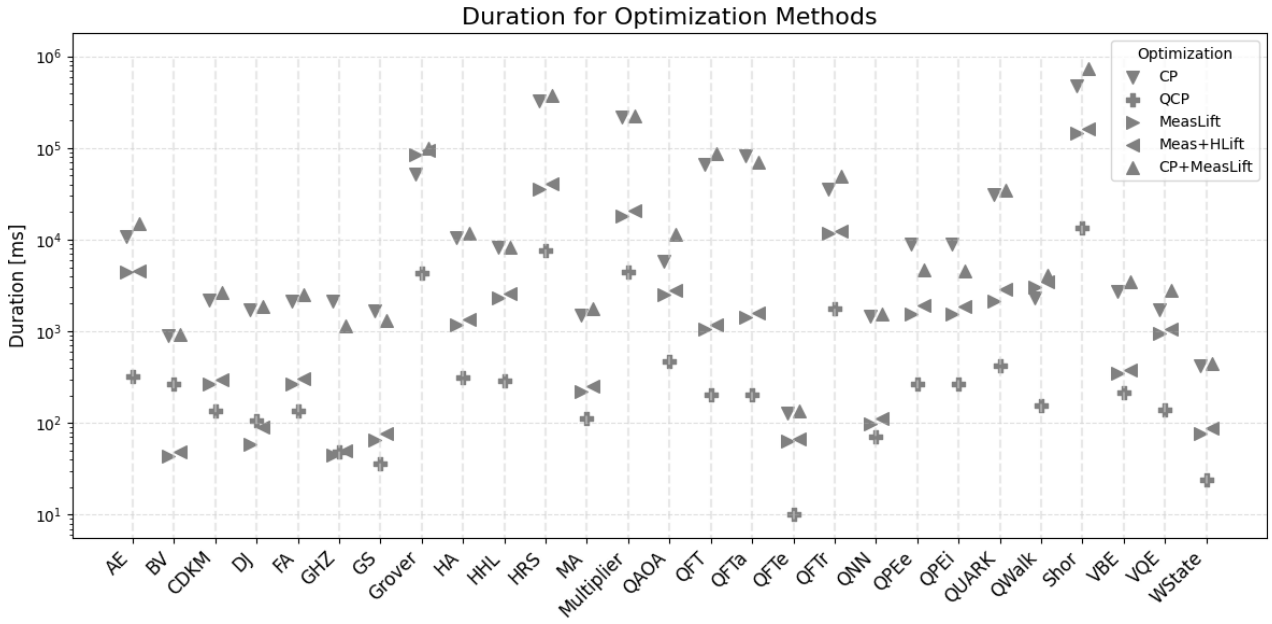


Fig. 10: The duration in ms needed to apply the optimization techniques quantum-classical constant propagation (CP), quantum constant propagation (QCP), measurement lifting (MeasLift), measurement lifting with Hadamard lifting (Meas+HLift), and constant propagation with measurement lifting (QCP+MeasLift). It is noteworthy that the usage of constant propagation makes the compilation much longer. Measurement lifting is quicker than measurement lifting with Hadamard lifting. In some cases, constant propagation and measurement lifting is quicker than constant propagation alone. The comparison with quantum constant propagation should be taken with care, as this routine has been executed on another framework without casting to MLIR.

We therefore suggest that a reduction potential of constant propagation can be examined by applying the routine on small circuits. We propose to apply constant propagation on a small test circuit of the algorithm under analysis. If the quantum resources of this circuit can be reduced, it could be worthwhile to also apply constant propagation to larger circuits of the same algorithm. Alternatively, one could analyze the changes introduced by constant propagation and find patterns to apply instead of running constant propagation on the complete circuit. In this case, one needs to be careful not to unwittingly change the semantics.

We could also show that in some scenarios constant propagation and measurement lifting together are more effective than each routine on its own. We could, however, not show that Hadamard lifting improves measurement lifting.

We conclude that quantum-classical constant propagation, measurement lifting, and their combination, can be useful tools to improve quantum circuits, especially in the NISQ era. It can ease qubit mapping pressure, and lower calculation times as well as error rates on current hardware. By selectively outsourcing work to classical control when it is safe to do so, we keep quantum resources focused where they deliver value. This focused use of quantum hardware is a practical step toward reliable, scalable quantum advantage.

Acknowledgment

We would like to thank Damian Rovara for the fruitful discussions that enriched this work, as well as for generously providing code templates.

Generative AI has been used in this work to create first code drafts (Microsoft Copilot) and review human-written text (GPT-4.1).

This work has been funded by DLR QCI via the CLIQUE project, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No. 101001318, and the Munich Quantum Valley (MQV K5 + K7), which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

References

[1] J. Preskill, "Quantum computing and the entanglement frontier," *Bulletin of the American Physical Society*, 2013.

[2] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Annual Symposium on Foundations of Computer Science*, IEEE, 1994.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symposium on Theory of Computing*, ACM, 1996.

[4] Y. Chen and Y. Stade, "Quantum constant propagation," in *International Static Analysis Symposium*, Springer, 2023.

[5] D. Rovara, L. Burgholzer, and R. Wille, "Qubit reuse beyond reorder and reset: Optimizing quantum circuits by fully utilizing the potential of dynamic circuits," *arXiv preprint arXiv:2511.22712*, 2025.

[6] L. Burgholzer, Y. Stade, T. Peham, and R. Wille, "MQT Core: The backbone of the Munich Quantum Toolkit (MQT)," *Journal of Open Source Software*, 2025.

[7] "GitHub - LiRem101/mqt-core at mlir/quantum-resource-reduction · GitHub," Accessed: May 11, 2026. [Online]. Available: https://github.com/LiRem101/mqt-core/tree/mlir/quantum-resource-reduction

[8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information - 10th Anniversary Edition*, 10th ed. Cambridge University Press, 2010.

[9] IBMQuantum. "Ibm_aachen," Accessed: Feb. 2, 2026. [Online]. Available: https://quantum.cloud.ibm.com/computers?system=ibm%5C_aachen

[10] IBMQuantum. "Ibm_torino," Accessed: Feb. 2, 2026. [Online]. Available: https://quantum.cloud.ibm.com/computers?system=ibm%5C_boston

[11] IBMQuantum. "Ibm_kingston," Accessed: Feb. 2, 2026. [Online]. Available: https://quantum.cloud.ibm.com/computers?system=ibm%5C_kingston

[12] I. IonQ. "IonQ Aria: Practical Performance," Accessed: Feb. 2, 2026. [Online]. Available: https://ionq.com/resources/ionq-aria-practical-performance

[13] I. Corporation. "Intel® Core™, Intel® Core Ultra, and Intel Processor Comparison Chart for Laptops," Accessed: Feb. 2, 2025. [Online]. Available: https://www.intel.com/content/www/us/en/content-details/843334/intel-core-intel-core-ultra-and-intel-processor-comparison-chart-for-laptops.html

[14] M. Dobšíček, G. Johansson, V. Shumeiko, and G. Wendin, "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark," *Physical Review A*, 3 2007.

[15] S. Beauregard, "Circuit for Shor's algorithm using 2n+3 qubits," *Quantum Information & Computation (QIC)*, 2003.

[16] D. Rattacaso, D. Jaschke, M. Ballarin, I. Siloi, and S. Montangero, "Quantum circuit compilation with quantum computers," *Physical Review Research*, 3 2025.

[17] N. Quetschlich, L. Burgholzer, and R. Wille, "Compiler optimization for quantum computing using reinforcement learning," in *Design Automation Conference (DAC)*, IEEE, 2023.

[18] R. Zen, M. Nägele, and F. Marquardt, "Reusability report: Optimizing T-count in general quantum circuits with AlphaTensor-Quantum," *Nature Machine Intelligence*, 2025.

[19] S. Dang, Y. Son, and B. Kim, "Qukkos: Measurement-Based Quantum Compilation System," in *International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, IEEE, 2025.

[20] K. Karuppasamy, V. Puram, S. Johnson, and J. P. Thomas, "A Comprehensive Review of Quantum Circuit Optimization: Current Trends and Future Directions," *Quantum Reports*, 2025.

[21] M. Praba, A. A. A. Samhan, V. S. Pandi, A. Sowmiya, N. Ravi, and G. Karthikeyan, "Integration of Machine Learning with Quantum Computing: Compilation of Quantum Circuit for Efficacy on Noisy Intermediate-Scale Quantum Devices," in *Asian Conference on Innovation in Technology*, IEEE, 2025.

[22] L. Sünkel, P. Altmann, M. Kölle, G. Stenzel, T. Gabor, and C. Linnhoff-Popien, "Quantum Circuit construction and Optimization through Hybrid Evolutionary Algorithms," in *Genetic and Evolutionary Computation Conference*, 2025.

[23] Z. Li, J. Peng, Y. Mei, S. Lin, Y. Wu, O. Padon, and Z. Jia, "Quarl: A Learning-Based Quantum Circuit Optimizer," *Proceedings of the ACM on Programming Languages*, 2024.

[24] T. Fösel, M. Yuezhen Niu, F. Marquardt, and L. Li, "Quantum circuit optimization with deep reinforcement learning," *arXiv e-prints*, 2021, Art. no. arXiv:2103.07585. arXiv: 2103.07585 [quant-ph].

[25] J. Arora, M. Xu, S. Westrick, P. Liu, D. Li, Y. Ding, and U. A. Acar, "Local Optimization of Quantum Circuits," in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2025.

[26] T. V. Forster, N. Quetschlich, and R. Wille, "Quantum Circuit Optimization for the Fault-Tolerance Era: Do We Have to Start from Scratch?" In *IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, 2025.

[27] P. Hopf, L. Burgholzer, and R. Wille, "Quantum Circuit Compilation for Superconducting Bus-Resonator Architectures," in *Design, Automation and Test in Europe (DATE)*, 2026.

[28] N. Quetschlich, L. Burgholzer, and R. Wille, "Predicting Good Quantum Circuit Compilation Options," in *IEEE International Conference on Quantum Software (QSW)*, IEEE, 2023.

[29] N. Quetschlich, F. J. Kiwit, M. A. Wolf, C. A. Riofrio, L. Burgholzer, A. Luckow, and R. Wille, "Towards Application-Aware Quantum Circuit Compilation," in *IEEE International Conference on Quantum Software (QSW)*, IEEE, 2024.

[30] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, 2nd ed. Pearson Education, 2007.

[31] V. Uotila, I. Salmenperä, L. Becker, A. Meijer-Van De Griend, A. R. Shinde, and J. K. Nurminen, "Perspectives on Utilization of Measurements in Quantum Algorithms," in *IEEE International Conference on Quantum Software (QSW)*, IEEE, 2025.

[32] G. Yan, W. Wu, Y. Chen, K. Pan, X. Lu, Z. Zhou, Y. Wang, R. Wang, and J. Yan, "Quantum Circuit Synthesis and Compilation Optimization: Overview and Prospects," *arXiv preprint arXiv:2407.00736*, 2024.

[33] T. Hao, A. Xu, and S. Tannu, "Reducing T Gates with Unitary Synthesis," *arXiv preprint arXiv:2503.15843*, 2026.

[34] D. Mills, I. Williams, J. Swain, G. Matos, E. Rinaldi, and A. Koziell-Pipe, "Reinforcement Learning for Adaptive Composition of Quantum Circuit Optimisation Passes," *arXiv preprint arXiv:2601.21629*, 2026.

[35] L. Remme, A. Weinert, and A. Waschk, "Optimization of Hybrid Quantum-Classical Algorithms," in *IEEE International Conference on Quantum Software (QSW)*, IEEE, 2025.

[36] J. Ye, F. Zhang, S. Xia, X. Guo, X. Wu, J. Zhao, and Y. Xue, "QSPE: Enumerating Skeletal Quantum Programs for Quantum Library Testing," *arXiv preprint arXiv:2602.00024*, 2026.

[37] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, "Measurement-based quantum computation," *Nature Physics*, 2009.

[38] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, 1982.

[39] L. Burgholzer, D. Haag, Y. Stade, D. Rovara, P. Hopf, and R. Wille, "The MQT Compiler Collection: A blueprint for a future-proof quantum-classical compilation framework," 2026. arXiv: 2604.08674.

[40] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, et al., "The MQT handbook: A summary of design automation tools and software for quantum computing," in *IEEE International Conference on Quantum Software (QSW)*, 2024. arXiv: 2405.17543, A live version of this document is available at https://mqt.readthedocs.io.

[41] C. Lattner, M. Amini, U. Bondhugula, A. Cohen, A. Davis, J. Pienaar, et al., "Mlir: Scaling compiler infrastructure for domain specific computation," in *International Symposium on Code Generation and Optimization*, IEEE, 2021.

[42] P. Hopf, E. Ochoa, Y. Stade, D. Rovara, N. Quetschlich, I. A. Florea, J. Izaac, R. Wille, and L. Burgholzer, "Integrating Quantum Software Tools with (in) MLIR," in *Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*, 2026.

[43] N. Quetschlich, L. Burgholzer, and R. Wille, "MQT Bench: Benchmarking software and design automation tools for quantum computing," *Quantum*, 2023, MQT Bench is available at https://www.cda.cit.tum.de/mqtbench/.

[44] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum Amplitude Amplification and Estimation," *arXiv preprint quant-ph/0005055*, 2000.

[45] E. Bernstein and U. Vazirani, "Quantum complexity theory," in *Symposium on Theory of Computing*, 1993.

[46] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Physical Review A*, 1 1996.

[47] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," *arXiv preprint quant-ph/0410184*, 2004.

[48] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 1992.

[49] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," in *Fundamental Theories of Physics*, Springer Verlag, 1989.

[50] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum Algorithm for Linear Systems of Equations," *Physical Review Letters*, 15 2009.

[51] T. Häner, M. Roetteler, and K. M. Svore, "Optimizing quantum circuits for arithmetic," *arXiv preprint arXiv:1805.12445*, 2018.

[52] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
[53] T. G. Draper, "Addition on a quantum computer," *arXiv preprint quant-ph/0008033*, 2000.
[54] L. Ruiz-Perez and J. C. Garcia-Escartin, "Quantum arithmetic with the quantum Fourier transform," *Quantum Information Processing*, 2017.
[55] IBMQuantum. "Estimatorqnn - Qiskit Machine Learning 0.9.0," Accessed: Feb. 27, 2026. [Online]. Available: https://qiskit-community.github.io/qiskit-machine-learning/stubs/qiskit%5Fmachine%5Flearning.neural%5Fnetworks.EstimatorQNN.html
[56] A. Y. Kitaev, "Quantum measurements and the Abelian stabilizer problem," *arXiv preprint quant-ph/9511026*, 1995.
[57] J. R. Finžgar, P. Ross, L. Hölscher, J. Klepsch, and A. Luckow, "Quark: A Framework for Quantum Computing Application Benchmarking," in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, 2022.
[58] QUARK-framework. "Github - QUARK-framework/QUARK: Framework for Quantum Computing Application Benchmarking (until version 2.2)," Accessed: Feb. 27, 2026. [Online]. Available: https://github.com/QUARK-framework/QUARK
[59] J. Kempe, "Quantum random walks: An introductory overview," *Contemporary Physics*, 2003.
[60] IBMQuantum. "EfficientSU2 — IBM Quantum Documentation," Accessed: Feb. 27, 2026. [Online]. Available: https://quantum.cloud.ibm.com/docs/de/api/qiskit/qiskit.circuit.library.EfficientSU2
[61] IBMQuantum. "RealAmplitudes — IBM Quantum Documentation," Accessed: Feb. 27, 2026. [Online]. Available: https://quantum.cloud.ibm.com/docs/de/api/qiskit/qiskit.circuit.library.RealAmplitudes2
[62] IBMQuantum. "TwoLocal — IBM Quantum Documentation," Accessed: Feb. 27, 2026. [Online]. Available: https://quantum.cloud.ibm.com/docs/de/api/qiskit/qiskit.circuit.library.TwoLocal
[63] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, 2014.
[64] W. Dür, G. Vidal, and J. I. Cirac, "Three qubits can be entangled in two inequivalent ways," *Physical Review A*, 2000.